Assoziative Suche für das Semantic Web

Ein netzbasierter Suchansatz unter Berücksichtigung semantischer und inhaltsbasierter Ähnlichkeiten

Dissertation
zur Verleihung des akademischen Grades
Doktor der Technischen Wissenschaften
an der Technischen Universität Graz
vorgelegt von

Dipl.-Ing. Peter Scheir
Institut für Wissensmanagement
peter.scheir@tugraz.at

Betreuer und erster Begutachter: Univ.-Prof. Dr. Klaus Tochtermann
Zweiter Begutachter: O.Univ.-Prof. Dr.Dr.h.c.mult. Hermann Maurer

Graz, Mai 2008

Kurzfassung

Während im gegenwärtigen Web das Sammeln von Information großteils von Menschen durchgeführt wird, soll es das Semantic Web ermöglichen diese Aufgabe durch Computerprogramme erledigen zu lassen. Zu diesem Zweck wird Information im Web mit maschinen-interpretierbaren Daten angereichert, welche Aufschluss über deren Bedeutung (Semantik) geben. Ansätze für die Suche im Semantic Web bauen auf diesen semantischen Metadaten auf und binden sie als zusätzliche Information in den Suchprozess ein um eine Steigerung der Retrieval-Leistung zu erzielen. Während die Vergabe von semantischen Metadaten zu Ressourcen im Web als eine zentrale Komponente für die Realisierung des Semantic Web gilt, ist im gegenwärtigen Web nur ein kleiner Anteil an Ressourcen mit semantischen Metadaten annotiert und somit für Maschinen interpretierbar.

In der vorliegenden Arbeit wird ein Beitrag zu dem sich derzeit noch in einer frühen Phase befindlichem Forschungsfeld der Suche im Semantic Web geleistet, indem ein Suchansatz für das Semantic Web entwickelt und evaluiert wird. Als Ergänzung zur exakten Suche basierend auf semantischen Metadaten sind in das entwickelte Modell Methoden aus dem assoziativen Retrieval integriert. Assoziatives Retrieval ist ein Such-Paradigma aus dem Information Retrieval, welches versucht ausgehend von einer Menge an relevanter Information potentiell relevante Zusatzinformation zu identifizieren, um die Suchergebnisse durch zusätzliche relevante Information zu verbessern. Im konkreten Fall werden Assoziationen basierend auf semantischer Ähnlichkeit zwischen Konzepten aus einer Ontologie und inhaltsbasierter Ähnlichkeit zwischen Ressourcen erzeugt. Neben einer generellen Steigerung der Retrieval-Leistung werden mit diesem Ansatz Situationen, in denen wenige Ressourcen mit semantischen Metadaten annotiert sind adressiert, um die Ergebnisse eines Retrieval-Systems auf Basis des entwickelten Modells zu verbessern. Das entwickelte Modell wird anhand eines Systems für die Suche am Semantic Desktop umgesetzt und basierend auf dieser Implementierung evaluiert.

Abstract

While in the current Web information is largely gathered by humans, the Semantic Web aims at having this task performed by computer programs. To achieve this goal information on the Web is enriched with machine-interpretable data that allows for identifying the meaning (semantics) of information on the Web. Approaches to search on the Semantic Web build upon this semantic metadata and integrate it as additional information into the process of search to increase retrieval effectiveness. While assigning semantic metadata to resources on the Web is seen as a central aspect for realizing the vision of the Semantic Web, only a small fraction of resources on the current Web is annotated with such metadata.

The work at hand contributes to the research field of search on the Semantic Web - currently being an early stage - by developing and evaluating an approach to information retrieval for the Semantic Web. In addition to exact search based on semantic metadata, methods stemming from associative information retrieval are integrated into the developed retrieval model. Associative information retrieval is a search paradigm which aims at identifying additional relevant information starting from information already known to be relevant and at increasing the performance of a retrieval system by this means. In the present work associations are based on semantic similarity between concepts from an ontology and on content-based similarity between resources. Besides an overall increase of retrieval effectiveness, the developed approach addresses situations in which few resources are annotated with semantic metadata, aiming at improving the results of a retrieval system that is based on the developed model. The model is implemented into a system for search on the Semantic Desktop and evaluated based on this implementation.

Ich versichere hiermit, diese Arbeit selbstständig verfasst
und keine anderen als die referenzierten Quellen verwendet zu haben.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Vorwort und Danksagung

Die vorliegende Arbeit ist in den letzten drei Jahren begleitend zu meiner Tätigkeit am Know-Center Graz und am Institut für Wissensmanagement der Technischen Universität Graz entstanden. Mit dem Know-Center und dem assoziierten Institut für Wissensmanagement existiert eine Umgebung, welche den wissenschaftlichen Austausch unterstützt und einen fruchtbaren Boden für die Erstellung dieser Dissertation darstellte. So konnte basierend auf der Vorarbeit von Mathias Lux (Lux, 2006) an das Konzept der semantischen Metadaten angeknüpft werden (vgl. Abschnitt 6.3.1)[1]. Auch die Vorarbeiten von Michael Granitzer (Granitzer, 2006) erwiesen sich als nützlich für die Entwicklung des hier vorgestellten Systems. Das KnowMiner-Framework diente als Grundlage für die Berechnung des verwendeten textuellen Ähnlichkeitsmaßes (vgl. Abschnitt 7.5.3). An dieser Stelle danke ich Prof. Dr. Hermann Maurer, Prof. Dr. Klaus Tochtermann und Dr. Stefanie Lindstaedt dafür, dass sie ein Ambiente wie dieses ins Leben gerufen und in den letzten Jahren maßgeblich geprägt haben.

Ebenfalls waren diese drei Personen ausschlaggebend in den Entstehungsprozess der vorliegenden Arbeit involviert. Bei meiner Bereichsleiterin Dr. Stefanie Lindstaedt möchte ich mich besonders dafür bedanken, dass sie

[1] Lux (2006) fasst den Begriff der semantischen Metadaten weiter als es Handschuh (2007) tut, diese können mit den in Abschnitt 6.3.1 beschriebenen semantischen Annotationen gleichgesetzt werden.

mir im Rahmen des APOSDLE-Projekts ermöglichte selbstständig an Problemstellung zu arbeiten, die sowohl von wissenschaftlichem als auch von praktischem Interesse sind und dass sie die Zeit fand mich bei meinen ersten wissenschaftlichen Publikationen zu unterstützen. Ebenfalls gilt mein Dank Prof. Dr. Klaus Tochtermann, der mir als Betreuer meiner Arbeit zu wichtigen Zeitpunkten mit entscheiden Tipps zur Seite stand. Schließlich danke ich Prof. Dr. Hermann Maurer der sich dankenswerter Weise als Begutachter der Arbeit zur Verfügung stellte.

Mein Dank gilt auch meinen Kollegen vom Know-Center und vom Institut für Wissensmanagement, welche meinen wissenschaftlichen Werdegang geprägt haben und mit denen ich unterschiedliche Aspekte dieser Arbeit diskutieren und bearbeiten durfte: Markus Strohmaier, der mir besonders in der Anfangsphase meines Dissertationsvorhabens zur Seite stand und meinen Sinn für die Wissenschaft schärfte. Meinem ehemaliger Kollegen, Mathias Lux mit dem sich in zahlreichen Diskussionen die Themen Ontologien, semantische Metadaten, und deren Verwendung für das Information Retrieval erschlossen haben. Meiner Sitznachbarin Viktoria Pammer, mit der ich eine Mitstreiterin auf dem Gebiet der Semantic-Web-Technologien fand und mit der ich gemeinsam an der semantischen Annotation von Dokumenten arbeiten durfte. Alexander Stocker für die zahlreichen mittäglichen Diskussionen zum Thema wissenschaftliches Arbeiten. Michael Granitzer für die Entwicklung des KnowMiner-Frameworks und für seine Kommentare zur Evaluierung des entwickelten Retrieval-Systems. Barbara Kump für ihre Zeit zur Diskussion der Validität des im Rahmen dieser Arbeit durchgeführten Experiments. Andreas Rath für seine Kommentare zur Komplexität des verwendeten Suchansatzes. Armin Ulbrich, Günter Beham und Tobias Ley für ihre kollegiale Zusammenarbeit im APOSDLE-Projekt, in dessen Rahmen die Implementierung des hier präsentierten Systems statt fand. Und schließlich Werner Klieber und Roman Kern für ihre Arbeit am KnowMiner-Framework.

Ich danke auch jenen Menschen, die mein privates Umfeld dahingehend geprägt haben, dass es sich positiv auf die Erstellung dieser Arbeit auswirkte. So danke ich meinen Eltern, die mich mein Leben lang unterstützt haben und dies immer noch bei jeder Gelegenheit tun. Ebenfalls gilt der Dank meiner Freundin Claudia Wagner für ihr Verständnis für die zahllosen

Wochenenden, die ich anstatt mit ihr mit der Dissertation verbrachte und dafür, dass sie sich die Zeit nahm diese Arbeit Korrektur zu lesen.

Auszüge aus der vorliegenden Arbeit sind bereits an anderen Stellen publiziert worden: Das erarbeitete Modell wurde in (Scheir u. a., 2005b), (Scheir, 2006), (Scheir u. Lindstaedt, 2006) und (Scheir u. a., 2007a) beschrieben, eine Beschreibung des entwickelten Systems erfolgte in (Scheir u. a., 2007a) und die Evaluierung des Prototypen wurde in (Scheir u. a., 2007a) und (Scheir u. a., 2007b) betrachtet. Neben allgemeinen Publikationen zum APOSDLE-System wie (Ulbrich u. a., 2006) und (Ley u. a., 2007) wurden auch die Einbettung dieser Arbeit in APOSDLE beschrieben (Ghidini u. a., 2007) und (Lindstaedt u. a., 2007). Ebenfalls wurde das entwickelte Werkzeug zur semantischen Annotation vorgestellt (Scheir u. a., 2006a) und (Pammer u. a., 2007). Zusätzliche erfolgte eine Betrachtung aktueller Systeme zur Suche im Semantic Web und am Semantic Desktop (Scheir u. a., 2007c).

1 Einleitung

Das Semantic Web (Berners-Lee u. a., 2001) ist als eine Erweiterung zum derzeit existierenden World Wide Web (WWW) (Berners-Lee u. a., 1999) gedacht. Während im gegenwärtigen Web das Sammeln von Information großteils von Menschen durchgeführt wird (während diese mit einem Browser durch das Web navigieren), soll es das Semantic Web ermöglichen diese Aufgabe durch Computerprogramme erledigen zu lassen. Um die Information im Web durch Maschinen verarbeitbar zu machen, zielt das Semantic Web darauf ab, diese mit maschinen-interpretierbaren Daten anzureichern, welche Aufschluss über deren Bedeutung (Semantik) geben sollen. Dies wird durch eine Schicht an *Metadaten* (Daten über Daten[1]) realisiert, welche in das aktuelle Web eingezogen wird.

Für die Formalisierung dieser maschinen-interpretierbaren Daten finden Ontologien (Gruber, 1993) Verwendung. Bei einer *Ontologie* handelt es sich um eine Wissensrepräsentation, mit welcher ein Teil der Realität modelliert und explizit spezifiziert wird. Die Spezifizierung erfolgt beispielsweise unter Verwendung einer formalen Sprache. Der Anwendungsfall von Ontologien im Kontext des Semantic Web ist die Beschreibung von Wissen und Zusammenhängen zwischen Wissen in einer Form, die es Maschinen ermöglicht daraus Schlüsse zu ziehen. Der Nutzen von Ontologien geht allerdings über deren Anwendung im Semantic Web hinaus. So werden beispielsweise Dokumentation und Teilung von Wissen in Form einer Ontologie als Anwen-

[1] Die Wörter Daten und Information werden hier, in diesem Zusammenhang synonym verwendet

dungsfälle von Ontologien für Menschen genannt.

Metadaten, für deren Definition auf das in einer Ontologie spezifizierte Wissen zurückgegriffen wird, werden als *semantische Metadaten* bezeichnet. Ein Charakteristikum semantischer Metadaten im Vergleich zu klassischen Metadaten ist, dass die Elemente der Wissensrepräsentationen, welche als Metadaten zu Ressourcen Verwendungen finden, mittels semantischen Relationen miteinander verbunden sein können. Die semantischen Relationen sind ebenfalls in der Ontologie definiert. Der Prozess der Vergabe semantischer Metadaten wird als *semantische Annotation* bezeichnet (Handschuh, 2005). Semantische Metadaten zu *Ressourcen*[2] werden auch als *semantische Annotationen* bezeichnet.

Die Anwendung von Technologien für das Semantic Web ist allerdings nicht nur auf das Web beschränkt. So hat es sich die Semantic-Desktop-Bewegung zum Ziel gemacht Technologien, die für das Semantic Web entwickelt wurden, für Desktop-Computing nutzbar zu machen. Ansätze Semantic-Web-Technologien für den Desktop bzw. für die Unternehmung zu nutzen sind allerdings nicht nur unter dem Namen *Semantic Desktop* bekannt. So präsentieren Maedche u. a. (2003) eine Architektur für ein *ontology-based knowledge management system* für den Einsatz im Unternehmen. Dieses System kann als Anwendung für den Semantic Desktop betrachtet werden, da Technologien, welche aus dem Semantic-Web-Umfeld entstanden sind, zur Realisierung des Systems herangezogen werden. Auch Uren u. a. (2006) betrachten die Verwendungen von Semantic-Web-Technologien für den Einsatz im Wissensmanagement. Uren u. a. (2006) sehen den Akt der semantischen Annotation von Dokumenten (also der Anreicherung von Dokumenten mit Metadaten aus einer Ontologie) als einen zentralen Nutzen für eine (Dokument-zentrierte) Wissensmanagement-Strategie im Unternehmen.

[2] In dieser Arbeit wird, wie im Semantic-Web-Umfeld üblich, unter einer Ressource alles das verstanden, für das eine URI vergeben werden kann

1.1 Problemstellungen

Im Folgenden werden zwei aktuelle Problemstellungen im Semantic Web beschrieben, welche die Grundlage dieser Arbeit bilden.

1.1.1 Problemstellung 1: Suche im Semantic Web

Suchansätze für Semantic Web und Semantic Desktop[3] basieren darauf, auf der Basis von semantischen Metadaten Ressourcen im Web oder am Desktop zu finden bzw. semantische Annotationen als zusätzliche Information in den Suchprozess einfließen zu lassen. Derzeit existiert ein breites Spektrum an Suchansätzen dieses neuen Suchparadigmas, welche unterschiedliche Charakteristika aufweisen. Auch besteht kein einheitlicher Zugang zum Thema Suche im Semantic Web (Castells u. a., 2007; Scheir u. a., 2007c).

Ein Möglichkeit der Differenzierung bildet die Unterscheidung zwischen Information Retrieval und Data Retrieval: Der Fokus im Information Retrieval liegt auf der Suche nach Information, welche die Informationsbedürfnisse des Nutzers eines Information-Retrieval-Systems befriedigt. Beim Data Retrieval wird nicht eine gereihte Liste an Informationsobjekten retourniert, sondern eine Menge an Ergebnissen, welche alle exakt auf eine Anfrage passen. Der Großteil der aktuell existierenden Retrieval-Ansätze für das Semantic Web, wie beispielsweise die Semantic-Web-Abfragesprache SPARQL sind Daten-Retrieval-lastig. Zentrale Charakteristika des Information Retrieval wie Vagheit, Unschärfe und Unsicherheit fallen bei diesen Suchansätzen weg, Anfragen an das Semantic Web gleichen jenen an eine Datenbank. Dies kann damit zusammenhängen, dass davon ausgegangen wird, dass im Semantic Web jede Art von Information mit genügend Metadaten versehen ist um detaillierte Aussagen über diese Information ableiten zu können.

Das Semantic Web befindet sich gegenwärtig in einer frühen Phase und derzeit ist noch unklar wie formal die semantische Schicht über dem bisherigen Web ausfallen soll. Hieraus resultiert auch das breite Spektrum der Suchansätze für das Semantic Web. Ein Semantic Web mit hohem Formalisierungsgrad ermöglicht es das Web wie eine Datenbank zu behandeln.

[3] Im Folgenden wird der Semantic Desktop, falls nicht extra erwähnt, als Teil der Semantic-Web-Bewegung betrachtet

Je mehr exakte Schlüsse aus den maschineninterpretierbaren Daten im Semantic Web durch ein Programm abgeleitet werden können, desto weniger besteht der Bedarf für Information-Retrieval-Ansätze, welche Vagheit und Unsicherheit beinhalten. Ein Semantic Web mit niedrigerem Formalisierungsgrad hingegen erfordert Information-Retrieval-Ansätze, welche in der Lage sind jene zusätzlichen semantischen Metadaten, die im Semantic Web angeboten werden, für die Suche und Reihung von Ergebnissen zu nutzen.

1.1.2 Problemstellung 2: Sparse Annotationen

Ansätzen für die Suche im Semantic Web ist gemein, dass sie semantische Metadaten für den Prozess der Suche nutzen. Die Anreicherung von Ressourcen mit semantischen Metadaten bietet zusätzliche Information, die in den Suchprozess eingebunden werden kann (Heflin u. Hendler, 2000; Spärck Jones, 2004)). Dies kann wiederum zu einer Steigerung der Retrieval-Leistung von Retrieval-Systemen für das Semantic Web führen. Während Handschuh (2005) die semantischen Annotation von Ressourcen im Web als zentrale Komponente für die Realisierung des Semantic Web sieht, stellen sowohl Kritiker (McCool, 2005) als auch Befürworter (Sabou u. a., 2006) des Semantic-Web-Gedankens fest, dass im gegenwärtigen Web nur ein kleiner Anteil an Ressourcen semantisch annotiert und somit für Maschinen interpretierbar ist.

In diesem Zusammenhang spricht man von einem niedrigen Durchdringungsgrad (coverage) des Webs mit semantischen Metadaten oder graphentheoretisch betrachtet von spärlichen (sparse) Annotationen der Ressourcen in jenem (Semantic-Web-)Graphen, der aus Ressourcen und den Konzepten aufgespannt wird, mit denen die Ressourcen semantisch annotiert sind.

Spärliche Annotation von Ressourcen stellt ein zentrales Hindernis für die Realisierung von Suchansätzen für das Semantic Web dar, welche auf Basis semantischer Metadaten operieren. Eine Suche basierend auf einer Menge von Konzepten kann zu keinem Ergebnis kommen, wenn die zu findenden Ressourcen nicht mit semantischen Metadaten annotiert sind. Szenarien in denen nur semantische Metadaten und kaum textuelle Inhalte von Ressourcen zur Verfügung stehen, wie zum Beispiel das Auffinden von Bildern oder (Semantic) Web Services, werden hierdurch verkompliziert oder unmöglich.

Aber auch die postulierten Vorteile von semantischer Zusatzinformation für die text-basierte Suche, wie die Erhöhung der Anzahl der gefundenen Ergebnisse, die Verbesserung der Reihung der Suchergebnisse (Heflin u. Hendler, 2000) (Iofciu u. a., 2005) oder das miteinander in Beziehung setzen von sonst isolierte Informationen mit Hilfe von Ontologien (Davies u. a., 2003), kommen nicht zum Tragen.

1.2 Lösungsansätze

Im Folgenden werden Lösungsansätze für die zuvor angeführten Problemstellung vorgeschlagen.

1.2.1 Ad. Problemstellung 1: Entwicklung eines Information-Retrieval-Modells für das Semantic Web

In der vorliegenden Arbeit wird ein Beitrag zu dem sich derzeit noch in einer frühen Phase befindlichem Forschungsfeld der Suche im Semantic Web geleistet, indem ein Suchansatz für das Semantic Web entwickelt und evaluiert wird. Die Modellentwicklung erfolgt aufbauend auf den Erkenntnissen einer im Vorfeld durchgeführten explorativen Studie zum Thema Suche im Semantic Web, in der ein Überblick über aktuelle Ansätze geben wird und Charakteristika dieser beleuchtet werden.

Die Möglichkeit zur Reihung der Suchergebnisse wird in dieser Arbeit als ein zentraler Aspekt für die Suche im Semantic Web betrachtet, da wie Stojanovic u. a. (2003) feststellen es unwahrscheinlich erscheint, dass Applikationen, bzw. jene Menschen die mit ihnen arbeiten, mit einer großen Menge an Suchergebnissen umgehen können werden, auch wenn diese alle relevant sind. Aus diesem Grund wird ein Information-Retrieval-Modell entwickelt, d.h. ein Modell, welches die Ergebnismenge nach deren Relevanz für eine Anfrage reiht. Dies steht im Gegensatz zu Data-Retrieval-Ansätzen für das Semantic Web wie der Abfragesprache SPARQL.

Die Entscheidung zur Interpretation von Suche im Semantic Web im Sinn von Information Retrieval und nicht im Sinn von Data Retrieval korrespondiert mit anderer aktueller Forschung im Semantic Web. So wurde mit April

2008 das *LarKC*-Projekt[4] gestartet, welches zum Ziel hat die Themen Suche von Information im Semantic Web und Schlussfolgern im Semantic Web zu fusionieren. Auch in diesem Projekt wird eine Richtung weg von exakten Schlüssen und Data Retrieval, hin zum Information Retrieval verfolgt.

Der in dieser Arbeit entwickelte Retrieval-Ansatz beruht auf einem Netzmodell. Konzepte aus einer Ontologie und Ressourcen, welche durch diese Konzepte semantisch annotiert sind werden als Knoten in einem Netz repräsentiert. Eine Kante im Netz von einem Konzeptknoten zu einem Ressourceknoten wird dann erzeugt, wenn eine Ressource mit einem Konzept semantisch annotiert ist. Ziel einer Suche im Netz ist es zu einer Anfrage, welche aus einer Menge an Konzepten besteht, eine Menge an Ressourcen zu finden und in weiterer Folge die Ergebnismenge nach Relevanz für die Anfrage zu reihen. Zu diesem Zweck wird die Netzstruktur, die dem hier vorgestellten Modell zu Grunde liegt, mittels eines Spreading-Activation-Ansatzes durchsucht. Ausgehend von einer Menge an initial aktivierten Konzeptknoten, welche zur Anfrage korrespondieren, breitet sich Aktivierung über die Kanten im Netz zu jenen Ressourcenknoten aus, welche mit diesen Konzeptknoten verbunden sind.

1.2.2 Ad. Problemstellung 2: Integration assoziativer Suche in das Modell

In das entwickelt Modell für die Suche im Semantic Web sind Methoden des Assoziativen Retrieval integriert, um einen Beitrag für die Suche in Situationen zu leisten, in denen Ressourcen spärlich mit semantischen Metadaten annotiert sind. Assoziatives Retrieval ist ein Such-Paradigma aus dem Information Retrieval, welches versucht ausgehend von einer Menge an relevanter Information potentiell relevante Zusatzinformation zu identifizieren um die Suchergebnisse durch zusätzliche relevante Information zu verbessern.

Als Ergänzung zur exakten Suche, basierend auf semantischen Metadaten, ist in das vorgestellte Modell assoziative Suchfunktionalität, basierend auf semantischer Ähnlichkeit zwischen Konzepten aus einer Ontologie

[4] `http://www.larkc.eu/` (18.05.2008) LarKC: *The Large Knowledge Collider: a platform for large scale integrated reasoning and Web-search*, gefördert im 7. Rahmenprogramm der Europäischen Kommission, Budget von €10.108.342,40

und inhaltsbasierter Ähnlichkeit zwischen Ressourcen, integriert. Neben einer generellen Steigerung der Retrieval-Leistung werden mit diesem Ansatz Situationen, in denen wenige Ressourcen mit semantischen Metadaten annotiert sind adressiert, um die Ergebnisse eines Retrieval-Systems auf Basis des entwickelten Modells zu verbessern.

Für die assoziative Suche werden Assoziationen zwischen Konzepten und Konzepten oder Assoziationen zwischen Ressourcen und Ressourcen in den Suchprozess miteingebunden. Auch beide Fälle sind möglich. Assoziationen zwischen Konzepten bzw. zwischen Ressourcen basieren auf deren Ähnlichkeit und werden ebenfalls als Kanten in jenem Netz modelliert, welches für die Suche von Ressourcen ausgehend von Konzepten Verwendung findet (vgl. Abschnitt 1.2.1). Das Retrieval Modell wird also um assoziative Suchfunktionalität erweitert, wobei das zugrunde liegende Modellierungsparadigma bestehen bleibt. Sowohl die exakte Suche als auch die assoziative Suche sind in einem Modell vereint. Auch wird das gleiche Suchparadigma, Spreading Activation, für beide Suchansätze verwendet.

1.2.3 Realisierung und Anwendungsdomäne

Das Semantic Web befindet sich derzeit in einer frühen Phase und ist noch nicht großflächig fertig gestellt. Allerdings existieren lokale Umgebungen im Desktop-Umfeld, welche auf Technologien für das Semantic Web aufbauen und in denen somit Bedingungen herrschen, wie sie für das Semantic Web angedacht sind. Das in dieser Arbeit entwickelte Modell ist in einer derartigen Desktop-Umgebung, in der Retrieval-Komponente des APOSDLE-Systems, implementiert. Das APOSDLE-System wird im Rahmen des APOSDLE-Projekts[5] entwickelt, dessen Ziel es ist Wissensarbeitern während ihrer Arbeit Hilfestellungen zu geben und ihnen in weiterer Folge den Aufbau von Kompetenzen für die Bewältigung von ähnlichen Situationen zu ermöglichen (Lindstaedt u. Mayer, 2006). Die erste, hier verwendete Version des APOSDLE-Systems wurde für die Domäne des *Requirements Engineering* entwickelt. Dies resultierte in einer Domänenontologie für die-

[5] `http://www.aposdle.org/` (18.05.2008) APOSDLE: *Advanced Process-Oriented Self-Directed Learning Environment*, gefördert im 6. Rahmenprogramm der Europäischen Kommission, Budget von €12.930.000

ses Feld und einer Menge an Dokumenten, welche verschiedene Themen aus dem Feld des Requirements Engineering aufbereiten. Dokumente, welche unterstützendes Material zum Thema Requirements Engineering enthalten, wie beispielsweise Definitionen, Beispiele oder Kurse, wurden zum Teil mit Konzepten aus der Domänenontologie annotiert.

Das APOSDLE-System, als System für den Semantic Desktop[6], stellt eine Umgebung zur Verfügung, welche alle Voraussetzungen für eine Evaluierung des entwickelten Suchansatzes erfüllt, da für die Suche gleiche Bedingungen wie im Semantic Web herrschen: So existieren eine Menge an Ressourcen, eine Ontologie und ein Menge an semantischen Annotationen, welche den Ressourcen Konzepte aus der Ontologie zuordnet. Die Adressierung der Ressourcen als auch die Formalisierung von Ontologie und semantischen Annotationen erfolgt mittels Standards für das Semantic Web. URIs werden für die Adressierung von Ressourcen verwendet und die Semantic-Web-Ontologiesprache OWL dient zur Formalisierung der Ontologie. Somit ist das hier vorgestellte *Modell* bereit für den Einsatz im Semantic Web. Das entwickelte *System* wird am Semantic Desktop umgesetzt und evaluiert, für seinen Einsatz im Semantic Web wären Modifikationen am System nötig.

1.2.4 Zusammenfassung

In der vorliegenden Arbeit wird ein Information-Retrieval-Modell für das Semantic Web präsentiert. Das vorgestellte Modell vereint die Funktionalität zur Suche nach Ressourcen ausgehend von einer Menge an Konzepten und die Funktionalität einer assoziativen Suche nach zusätzlichen relevanten Ressourcen (Assoziatives Retrieval). Als Modellierungsparadigma für das Retrieval-Modell wird ein Netzmodell verwendet, welches mittels eines Spreading-Activation-Ansatzes durchsucht wird. Das vorgestellte Modell ist in der Retrieval-Komponente des APOSDLE-Systems zur Suche am Semantic Desktop implementiert.

[6] In der vorliegenden Arbeit wird ein System, welches am Desktop implementiert ist und auf Technologien für das Semantic Web aufbaut, als System für den Semantic Desktop bezeichnet.

1.3 Forschungshypothese und Untersuchungsfragen

In der vorliegenden Arbeit wird eine Modell für Assoziatives Retrieval im Semantic Web entwickelt und in Form eines Retrieval-Systems für den Semantic Desktop umgesetzt, anhand dessen es evaluiert wird. Der vorliegenden Arbeit liegt folgende Forschungshypothese zu Grunde:

- Assoziatives Retrieval erhöht die Effektivität eines Information-Retrieval-Modells für das Semantic Web

Im Vorfeld der in der Arbeit durchgeführten Modellentwicklung werden aktuelle Ansätze zur Suche im Semantic Web beleuchtet und assoziative Retrieval-Methoden für ihre Anwendung im Kontext der Arbeit untersucht. Folgende Untersuchungsfragen werden bearbeitet:

- Welche Suchansätze für Semantic Web bzw. Semantic Desktop gibt es bereits und welche Themen eignen sich für eigene Forschung?
- Welche assoziativen Suchansätze gibt es bereits und welche Methoden werden für assoziative Suche verwendet?

Im Rahmen der in dieser Arbeit durchgeführten Modellentwicklung, Implementierung und Evaluierung werden folgende Untersuchungsfragen bearbeitet:

- Kann Assoziatives Retrieval für Suche im Semantic Web eingesetzt werden? Bzw. können assoziative Suchansätze in aktuelle Ansätze zur Suche im Semantic Web integriert werden?
- Kann Assoziatives Retrieval die Effektivität von Information-Retrieval-Systemen für Semantic Web oder Semantic Desktop steigern?

Assoziative Suchansätze basieren häufig auf Netzmodellen, Zusammenhänge zwischen Informationsobjekten werden als Graph modelliert. Es ergeben sich dadurch weitere Untersuchungsfragen:

- Wie verhalten sich Netz-basierte Retrieval-Modelle im Vergleich zu klassischen Modellen, wie dem Vektorraummodell? Können Netz-basierte Ansätze äquivalente Ergebnisse liefern?

- Können Netz-basierte Suchansätze für Information Retrieval im Semantic Web eingesetzt werden?

1.4 Aufbau der Arbeit

Kapitel 2 führt die grundlegenden Begriffe dieser Arbeit im Kontext von Semantic Web und Information Retrieval ein. Kapitel 3 geht auf Ansätze für die Suche im Semantic Web ein und gibt einen Überblick über aktuelle Modelle und Systeme in diesem Umfeld. Kapitel 4 stellt den Begriff des Assoziativen Retrieval vor, zeigt die Anwendung von Assoziativen Netzen und führt in Spreading Activation zur Verarbeitung von Assoziativen Netzen ein. Zusätzlich gibt es einen Überblick über Systeme, welche Assoziatives Retrieval implementieren, auf Assoziative Netzen basieren oder Spreading Activation nutzen. Kapitel 5 vergleicht netzbasierte Retrieval-Modelle mit dem Vektorraummodell.

In Kapitel 6 wird das im Rahmen dieser Arbeit entwickelte Retrieval-Modell zur Suche im Semantic Web vorgestellt. In Kapitel 7 wird das auf Basis dieses Modell entwickelte System präsentiert und auf Details der Implementierung eingegangen. Kapitel 8 führt allgemein in das Thema der Evaluierung von Information-Retrieval-Systemen ein und präsentiert darauf folgend den für die Evaluierung des entwickelten Systems gewählten Ansatz und die auf diesem Weg ermittelten Ergebnisse. Kapitel 9 fasst die Ergebnisse dieser Arbeit zusammen und gibt einen Ausblick.

Generell sind die Kapitel dieser Arbeit so gehalten, dass sie für sich selbst stehend gelesen werden können. Abbildung 1.1[7] gibt einen Überblick über die Abschnitte der Arbeit und einer möglichen Lesereihenfolge.

1.5 Wissenschaftliche Eigenleistungen

In diesem Abschnitt werden die wissenschaftlichen Eigenleistungen dieser Arbeit angeführt. Als wissenschaftliche Eigenleistungen werden jene Inhalte

[7] Um Modelle zu beschrieben wird in dieser Arbeit, wenn möglich, auf die *Unified Modeling Language* (*UML*) in der Version 2 (Object Management Group, 2007) zurückgegriffen, da diese ein standardisiertes Werkzeug für diese Aufgabe zur Verfügung stellt.

der Arbeit erachtet, die einen Teil zum Wissensbestand einer Disziplin beitragen und die in dieser Form - nach aktuellem Wissenstand des Autors - noch nicht verfügbar sind.

In Kapitel 3 wird Information Retrieval im Semantic Web untersucht und ein Überblick über aktuelle Modelle und Systeme für dieses Suchparadigma gegeben. Eine Sammlung und Klassifikation aktuell verfügbarer Systeme war in dieser Form nicht verfügbar und stellt somit eine wissenschaftliche Eigenleistung dar. Auch wird eine Definition für Information Retrieval im Semantic Web erarbeitet.

Kapitel 4 stellt die Konzepte des Assoziativen Retrievals, der Assoziativen Netze und von Spreading Activation vor. Ein Überblick über diese Konzepte ist in (Crestani, 1997a) verfügbar, welcher in diesem Kapitel vertieft wird. Zusätzlich werden Systeme basierend auf den zuvor genannten Konzepten untersucht. Hierbei wird zusätzlich zu den bereits in (Crestani, 1997a) verfügbaren System auf die Neuentwicklungen nach 1997 eingegangen.

Ebenfalls Kapitel 5 stellt eine wissenschaftliche Eigenleistung dieser Arbeit dar. Hier wird das Vektorraummodell mit netzbasierten Retrieval-Modellen verglichen. Hierzu existiert wenig Material in der Literatur und es ist keine Sammlung, welche unterschiedliche Aspekte des Vektorraummodells mit Netzmodellen vergleicht verfügbar.

Die Kapitel 6, 7 und 8 stellen die zentrale wissenschaftliche Eigenleistung dieser Arbeit dar. Hier wird das entwickelte Retrieval-Modell zur Suche im Semantic Web vorgestellt, das System, welches auf Basis dieses Modells implementiert wird, präsentiert und im Anschluss evaluiert. Ebenfalls wird auf die gewählte Evaluierungsmethodik eingegangen und die Ergebnisse der Evaluierung werden angeführt.

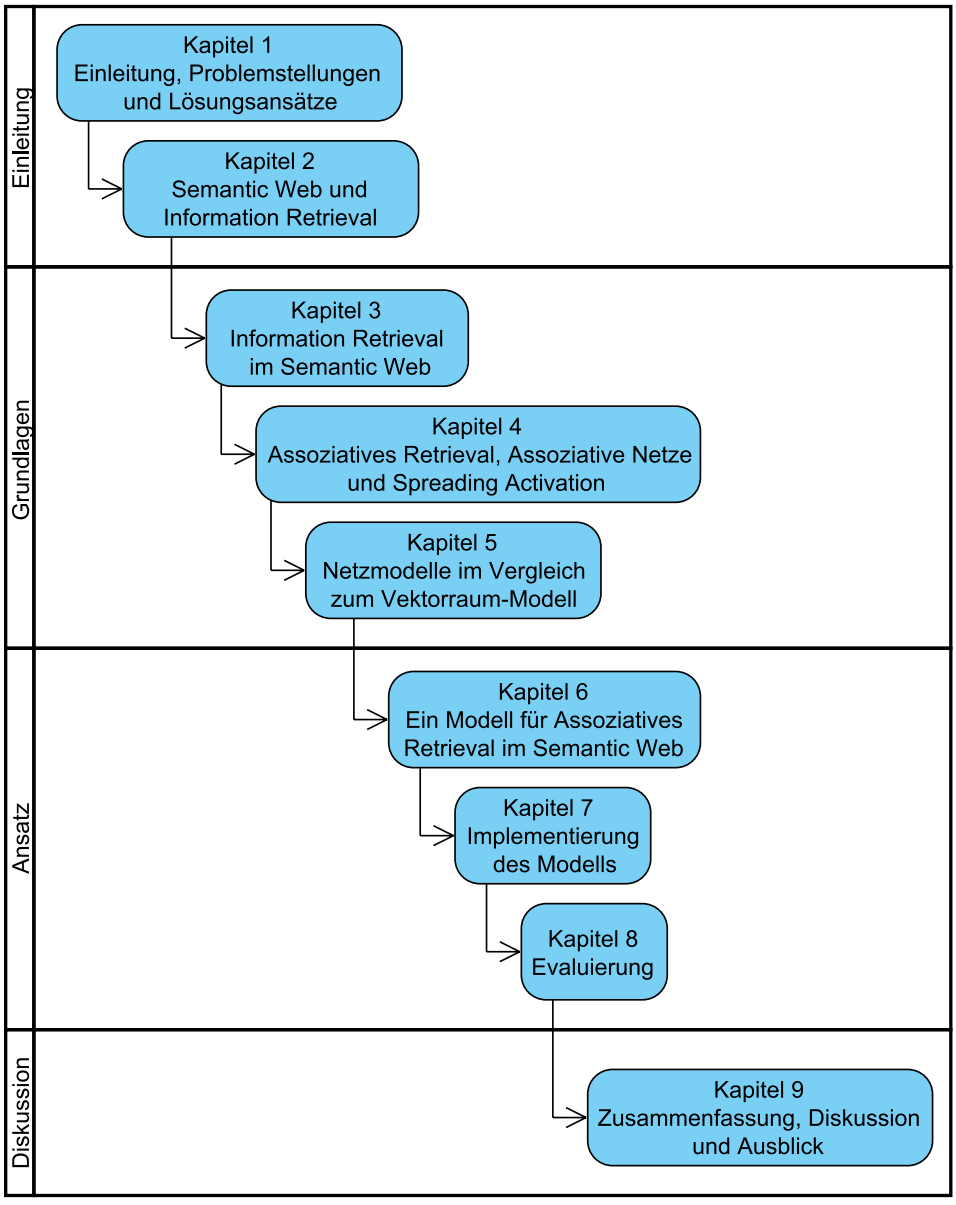

Abbildung 1.1: Die Abschnitte der Arbeit

2 Semantic Web und Information Retrieval

2.1 Einleitung

Dieses Kapitel beschreibt die zentralen Themengebiete der vorliegenden Arbeit. Es werden die Begriffe *Semantic Web* und *Information Retrieval* eingeführt.

Die Vorstellung des Semantic Web erfolgt einerseits durch die Beschreibung der Ziele dieses Vorhabens, andererseits anhand der Technologien die im Semantic-Web-Umfeld Verwendung finden. Im Rahmen der Vorstellung des Semantic Web wird des Weiteren auf das Konzept des *Semantic Desktop* eingegangen und der *Ontologie*-Begriff beleuchtet.

2.2 Semantic Web

Das *Semantic Web* (zu Deutsch auch das *Semantische Web*[1]) stellt eine Erweiterung zum derzeit existieren *World Wide Web* (Berners-Lee u. a., 1999) dar. Informationen im Web werden mit Maschinen-interpretierbaren Daten versehen um eine bessere Verarbeitbarkeit durch Computer zu ermöglichen. Während im derzeitigen Web Informationsbeschaffung eine Aufgabe ist, welche von Menschen durchgeführt wird (während der Navigation durch das

[1] Bei *Web* handelt es sich um eine aus dem Englischen übernommene Kurzform für *World Wide Web*. Die wörtliche Übersetzung von Semantic Web - *Semantisches Netz* - hat eine andere Bedeutung.

Web mittels Browsers), sollen derartige Aufgaben im Semantic Web durch eine Maschine erledigt oder zumindest unterstützt werden:

> *The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. ... In the near future, these developments will usher in significant new functionality as machines become much better able to process and "understand" the data that they merely display at present.*
>
> Berners-Lee u. a. (2001)

Um den Nutzen des Semantic Web für den Menschen zu verdeutlichen, beschreiben Berners-Lee u. a. (2001) ein Szenario, in dem ein Semantic-Web-Agent bei der Organisation der Physiotherapie für die kranke Mutter hilft. Bei diesem Agenten handelt es sich um ein Computer-Programm, welches mit dem Semantic Web interagiert und anstelle eines Menschen komplexe Aufgaben durchführt. Der Agent sucht ausgehend von der verordneten Behandlung, Therapiezentren, die diese anbieten, nicht weiter als 20 Meilen entfernt liegen und durch die Versicherung der Mutter gedeckt sind. Zusätzlich stimmt der Agent die Therapiezeiten mit den Kalendern der beiden Kinder ab, welche die Mutter zur Therapie bringen und von dort auch wieder abholen sollen.

Gerne wird auch das anschauliche Beispiel, der Buchung einer Urlaubsreise als Anwendungsfall von Programmen, welche auf Semantic-Web-Technologie aufsetzen gebracht: Einem Agenten werden Rahmenbedingungen einer Reise, wie Preis, Jahreszeit oder Region, genannt. Dieser Agent ermittelt dann aufgrund des Angebots unterschiedlicher Reiseanbieter eine Auswahl an Reiseangeboten, die dem Profil (und dem Terminplan) des Reisenden entsprechen. Dieses Szenario stellt eine technologisch nicht allzu ferne Vision des Semantic Web dar. Schon jetzt gibt es Portale, welche die Angebote verschiedener Reiseanbieter miteinander integrieren und vergleichbar machen. Zum gegenwärtigen Zeitpunkt bieten allerdings Reiseanbieter ihre Daten nicht mit Hilfe von Technologien für das Semantic Web an. Inhalte werden ausschließlich menschenlesbar aufbereitet ohne sie zusätzlich maschinen-interpretierbar zu beschreiben.

Für eine detailliertere Einführung in das Thema Semantic Web sei auf (Antoniou u. van Harmelen, 2004), (Davies u. a., 2003), (Fensel u. a., 2005) oder (Passin, 2004) verwiesen.

2.2.1 Semantic Web Stack

Die Technologien, welche auf dem Weg zur Realisierung der Semantic Web Idee zum Einsatz kommen, können anhand des *Semantic Web Stack* (gelegentlich auch Semantic Web Layer Cake oder Semantic Web Tower genannt) anschaulich dargestellt werden. Dieser Stack präsentiert aktuelle und zukünftige Technologien im Semantic Web und stellt eine Art Roadmap für das Semantic Web dar.

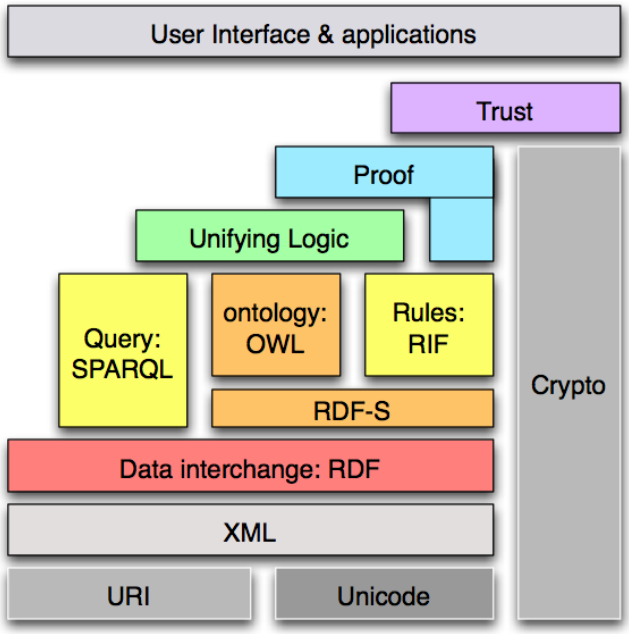

Abbildung 2.1: Der Semantic Web Stack 2006 (aus (Berners-Lee, 2006))

Mit der Weiterentwicklung der Technologien für das Semantic Web hat sich über die Zeit auch der Semantic Web Stack verändert. Neue Technologien sind dazugekommen und die Anordnung bestehender Technologien im Semantic Web Stack wurde angepasst. Eine detaillierte Beschreibung der Entwicklung des Semantic Web Stacks über die Jahre ist in (Gerber u. a.,

2006) zu finden. Im Folgenden wird auf die Elemente des Semantic Web Stacks in Abbildung 2.1 eingegangen und es werden Verweise auf zusätzliche Information angeführt.

Basistechnologien: Die Basis für die Technologien um das Semantic Web bilden die Standards für Unicode und URI. Bei Unicode handelt sich um einen Standard zur Kodierung von Schriftzeichen. *Unicode* (The Unicode Consortium, 2006) bietet für eine Vielzahl von international verwendeten Zeichen eine eindeutige Kodierung und wird kontinuierlich erweitert.

Die Identifikation von Ressourcen im Semantic Web erfolgt über *Uniform Resource Identifier* (*URI*) (Berners-Lee u. a., 2005). Bei URIs handelt es sich um die Verallgemeinerung der *Uniform Resource Locators* (*URL*s) (Berners-Lee u. a., 1994), welche für Ressourcen im World Wide Web entworfen wurden. Mit URIs können sowohl Web-Ressourcen, wie Webseiten oder Teile daraus, als auch physikalische Ressourcen, wie Bücher oder Personen, referenziert werden.

Als weitere grundlegende Technologie im Semantic Web kommt die *Extensible Markup Language* (*XML*[2]) zum Einsatz. In dieser Auszeichnungs-Sprache vom W3C werden Daten in Form eines Baums strukturiert. XML dient im Semantic Web primär zur Serialisierung der im Semantic Web Stack darüber liegenden Sprachen wie RDF, RDF Schema oder OWL.

RDF: Das *Resource Description Framework* (*RDF*) (Klyne u. Carroll, 2004) wurde 1999 vom W3C veröffentlicht. Ziel von RDF ist das Beschreiben von Ressourcen zu ermöglichen. Ressourcen werden über URIs adressiert, somit kann eine Ressource alles sein für das ein URI vergeben werden kann. Daher ist es auch möglich Ressourcen außerhalb des Webs zu adressieren. Ressourcen werden durch Eigenschaften (Properties) und zugehörige Eigenschaftswerte beschrieben.

Eine Ressource, eine Eigenschaft und deren Wert bilden zusammen eine Aussage (Statement). In RDF werden die Elemente eines Statements als Subjekt, Prädikat und Objekt bezeichnet. Beim Subjekt handelt es sich um die zu beschreibende Ressource. Der Teil der das Property identifiziert wird als Prädikat bezeichnet. Der Eigenschaftswert bildet das Objekt. Jedes RDF

[2] `http://www.w3.org/XML/` (18.05.2008)

Statement kann als ein Tripel (subject, predicate, object) ausgedrückt werden. Als Objekt eines Statements sind sowohl andere Ressourcen als auch *Literale* zulässig. Unter Literalen versteht man Werte wie Zahlen oder Zeichenketten. Ein Literal darf ausschließlich als Objekt eines RDF-Statements verwendet werden. Aus dem Zusammenschluss mehrerer Statements entsteht eine vernetzte Datenstruktur die als der RDF-Graph[3] bezeichnet wird. Dieser kann graphisch als Diagramm aus Knoten und Kanten (*nodes-and-arcs diagrams*) dargestellt werden (Manola u. Miller, 2004).

Der RDF-Standard definiert ein abstraktes Datenmodell zur Beschreibung von Ressourcen auf syntax-neutrale Art. Ausgehend von diesem Datenmodell existieren verschiedene Serialisierungsformen zu RDF, die es ermöglichen den RDF-Graphen auszudrücken. Die derzeit populärste ist *RDF/XML* (Beckett, 2004). Der zentrale Vorteil an RDF/XML ist die Vielzahl der Verfügbaren XML-Parser. Als Auszeichnungs-Sprache, welche Daten in Form eines *Baumes* strukturiert ist XML nicht optimal für die Serialisierung des RDF *Graphen*. Das Argument der guten Lesbarkeit von XML für den Menschen hält im Fall von RDF/XML nicht. Aus diesem Grund existieren weiter Sprachen für die Serialisierung von RDF wie Beispielsweise *Notation3* (*N3*) (Berners-Lee, 2006) und die zeilenbasierte Sprache *N-Triples* (Beckett, 2001).

RDF Schema: RDF ermöglicht es Aussagen über Ressourcen unter Zuhilfenahme von Properties und deren Werten zu treffen. Es besteht allerdings keine standardisierte Möglichkeit eine Aussage über den Typ von Ressourcen oder Properties zu treffen oder Beziehungen zwischen Typen von Ressourcen und Properties festzulegen. Dies wird mit einer Erweiterung zu RDF, der *RDF Vocabulary Description Language 1.0: RDF Schema* (kurz *RDF Schema*) (Brickley u. Guha, 2004) gelöst.

RDF Schema ist ein Vokabular, das es ermöglicht anwendungsspezifische Klassen und Properties zu definieren und festzulegen welche Klassen mit welchen Properties zusammen verwendet werden. Somit ist RDF Schema objekt-orientierten Programmiersprachen ähnlich. Ressourcen können als Instanzen einer oder mehrerer Klassen ausgezeichnet werden. Klassen

[3] beim RDF-Graphen handelt es sich um einen benannten, gerichteten Graphen

und Properties können hierarchisch organisiert werden, indem eine Klasse bzw. eine Eigenschaft als Unterklasse bzw. als Untereigenschaft einer anderen festgelegt wird. RDF Schema unterscheidet sich allerdings dadurch von objekt-orientierten Programmiersprachen, dass nicht Properties Teile einer Klassenbeschreibung sind, sondern für Properties festgelegt wird auf welche Klassen sie anwendbar sind.

RDF Schema ist als Satz an vordefinierten RDF-Ressourcen mit festgelegter Bedeutung realisiert. Somit sind Beschreibungen in RDF Schema gültige RDF-Graphen.

Sowohl bei RDF als auch bei RDF Schema handelt es sich um formale Sprachen, deren Semantik in (Hayes, 2004) definiert ist. Im Gegensatz zu OWL DL und OWL Lite (vgl. nächsten Abschnitt) besteht bei RDF und RDF Schema keine Verbindung zu Beschreibungslogiken (Baader u. a., 2003). Die Basis für logische Schlüsse in RDF und RDF Schema bilden sogenannte *Entailment Rules*[4]. Hierbei handelt es sich um Regeln, die auf die Menge an Tripel des RDF-Graphen angewendet werden um neue Tripel zu erzeugen, die in weiterer Folge dem RDF-Graphen hinzugefügt werden können. Eine detailliere Einführung zur Semantik von RDF und RDF Schema ist in (Hitzler u. a., 2008) zu finden.

OWL: Die *Web Ontology Language* (*OWL*[5]) ist eine Ontologie[6]-Sprache für das World Wide Web. Sie wurde Anfang 2004 als sechsteilige W3C Empfehlung veröffentlicht (McGuinness u. Harmelen, 2004) und stellt die offizielle Weiterentwicklung zu DAML+OIL (Connolly u. a., 2001) dar. OWL führt auf Basis von RDF Schema zusätzliches Vokabular ein, um größere Ausdrucksstärke zu erreichen.

OWL wurde als Beschreibungssprache für das Semantic Web entwickelt. Mit OWL können Klassen unterschiedlichen Typs (mit unterschiedlichen URIs) als gleich ausgezeichnet werden und neue Klassen als Schnitt- oder Vereinigungsmenge von bestehenden Klassen deklariert werden.

[4] entail: engl. für bedingen, zur Folge haben

[5] Die korrekte Abkürzung für Web Ontology Language wäre eigentliche WOL. Die Kurzform OWL existiert in Anlehnung an die Eule *Owl* im Märchen *Winnie the Pooh*, die ihren Namen mit W, O, L buchstabierte (Hendler, 2004).

[6] Eine detaillierte Einführung des Ontologie-Begriffs ist in Abschnitt 2.2.3 zu finden

OWL hat drei Untersprachen die unterschiedliche Ausdrucksstärke aufweisen und für unterschiedliche Anwendungsfälle entwickelt wurden (nach (McGuinness u. Harmelen, 2004) und (Smith u. a., 2004)):

1. *OWL Lite* bietet eine einfache Möglichkeit der Migration für bestehende Taxonomien und Thesauri. OWL Lite erlaubt die Definition von Klassenhierarchien mit einfachen Einschränkungen (Constraints) zu Properties und weist eine niedrigere formale Komplexität als OWL DL auf.
2. *OWL DL* trägt diesen Namen Aufgrund ihres Bezugs zu *Description Logics* (zu Deutsch, *Beschreibungslogiken*), welche laut Horrocks u. a. (2003) starken Einfluss auf die Ausgestaltung von OWL hatten. Das Forschungsfeld der Beschreibungslogiken (Baader u. a., 2003) beschäftigt sich mit der formalen Repräsentation von Wissen, welche das Ziehen logischer Schlüsse gestattet. OWL DL beinhaltet alle Sprachkonstrukte von OWL, allerdings werden einige Einschränkungen getroffen um die Berechenbarkeit von Schlüssen sicher zu stellen. Zum Beispiel darf eine Klasse nicht auch eine Instanz oder eine Property sein und eine Property darf nicht auch eine Klasse oder eine Instanz sein. Somit stellt OWL DL ein Maximum an Ausdrucksstärke zur Verfügung und garantiert zugleich *Vollständigkeit* (alle Schlüsse sind berechenbar) und *Berechenbarkeit* (alle Berechnungen enden in endlicher Zeit).
3. *OWL Full* bietet maximale Ausdrucksstärke und die syntaktische Freiheit von RDF, allerdings wird die Berechenbarkeit nicht mehr garantiert. OWL Full erlaubt die Erweiterung des durch RDF und OWL vordefinierten Vokabulars. Aus derzeitigem Standpunkt erscheint es als unwahrscheinlich jemals eine Software zu entwickeln, welche aus allen in OWL Full möglichen Sprachkonstrukten logische Schlussfolgerungen ziehen kann (vgl. (McGuinness u. Harmelen, 2004) und (Smith u. a., 2004)).

Sowohl in punkto Ausdrucksstärke als auch in der Möglichkeit gültige Schlüsse ziehen zu können ist jede Sprache eine Erweiterung eines seiner Vorgänger (vgl. (McGuinness u. Harmelen, 2004) und (Smith u. a., 2004)):

- Jede gültige OWL Lite Ontologie ist auch ein gültige OWL DL Ontologie.
- Jede gültige OWL DL Ontologie ist auch ein gültige OWL Full Ontologie.
- Jeder gültige Schluss in einer OWL Lite Ontologie ist auch ein gültiger Schluss in einer OWL DL Ontologie.
- Jeder gültige Schluss in einer OWL DL Ontologie ist auch ein gültiger Schluss in einer OWL Full Ontologie.

Wie bei DAML+OIL und RDF Schema ist jedes OWL Dokument (egal ob OWL Lite, DL oder Full) ein korrekter RDF-Graph. Auf der anderen Seite ist jedes RDF-Dokument ein gültiges OWL Full Dokument, da OWL auf RDF basiert und OWL Full die Erweiterung um beliebige RDF-Statements erlaubt. Allerdings ist, aufgrund der sprachlichen Einschränkungen, die in diesen Sub-Sprachen getroffen worden sind, nicht jedes RDF Dokument auch ein gültiges OWL DL oder OWL Lite Dokument. Einen exakten Überblick über die Einschränkungen geben Dean u. Schreiber (2004).

Die Semantik von OWL ist in (Patel-Schneider u. a., 2004) einerseits als alleinstehende modell-theoretische Spezifikation, andererseits als Erweiterung der Semantik von RDF und RDF Schema (Hayes, 2004) spezifiziert. Die alleinstehende modell-theoretische Spezifikation von OWL erfolgt für die beiden Sprachen OWL Lite und OWL DL und ist laut Patel-Schneider u. a. (2004) einfacher als die auf der Semantik von RDF und RDF Schema (Hayes, 2004) aufbauende Spezifikation. Eine detailliere Einführung zur Semantik von OWL ist in (Horrocks u. a., 2003) und (Hitzler u. a., 2008) zu finden.

SPARQL: Bei der *SPARQL Protocol And RDF Query Language* (*SPARQL*) handelt es sich um eine RDF-Abfragesprache die der Structured Query Language (SQL) nachempfunden ist. SPARQL kann als Nachfolger zur *RDF Data Query Language* (*RDQL*) von Seiten des W3C angesehen werden und ist seit Jänner 2008 eine W3C Recommendation (Prud'hommeaux u. Seaborne, 2008). RDQL wurde das erste Mal mit der Version 1.2.0 des Semantic

Web Frameworks *Jena*[7] veröffentlicht und im Anfang 2004 als Member Submission zur Standardisierung beim W3C eingereicht (Seaborne, 2004).

Rule Interchange Format: Mit dem *Rule Interchange Format* (*RIF*) soll eine Regel-Sprache für das Semantic Web geschaffen werden. RIF soll es ermöglichen aus Wenn-dann-Regeln Schlüsse zu ziehen, wie sie beispielsweise in Experten-Systemen Verwendung finden. Zum gegenwärtigen Zeitpunkt werden Use Cases und Requirements (Ginsberg u. a., 2006) für RIF gesammelt.

Unifying Logic: Die Aufgabe der Unifying-Logic-Schicht ist es die unterschiedlichen Ansätze von Logic die derzeit und in Zukunft im Semantic Web Verwendung finden (wie OWL-DL und RIF) miteinander zu vereinen und zugänglich zu machen.

Proof: In der Beweisschicht (Proof) des Semantic Web Stacks geht es darum zu verifizieren ob die in den darunterliegen Schichten getroffenen Aussagen wahr sind. Zu diesem Zweck werden Beweissprachen eingesetzt. Diese sollen es ermöglichen den Weg, auf dem eine Applikation zu einem Schluss gekommen ist, zu explizieren und dadurch von einer anderen Applikation überprüfbar zu machen.

Trust: Die Vertrauenswürdigkeit (Trust) von Information im Semantic Web hängt davon ab von welcher Quelle diese Information erhalten wurden. Hierbei ist relevant inwieweit dieser Quelle getraut werden kann und ob die Quelle auch die Quelle ist, für die sie sich ausgibt. Von einer Quelle sind somit deren *Vertrauenswürdigkeit* und *Authentizität* relevant (Gerber u. a., 2006).

Crypto: Die Kryptographieschicht im Semantic Web Stack hat zwei primäre Ziele. Auf der einen Seite besteht für die Trust-Schicht die Notwendigkeit die Authentizität einer Quelle eindeutig verifizierbar zu machen. Zu diesem Zweck können digitale Signaturen zum Einsatz kommen. Berners-Lee (2006) beschreibt diesen Anwendungsfall.

Auf der anderen Seite ist die Verschlüsselung von Daten eine relevante Aufgabenstellung, um den Datentransfer vor dem Abhören durch Dritte

[7] `http://jena.sourceforge.net/` (18.05.2008)

zu schützen. Dies hat zum Beispiel für E-Business-Anwendungen, in denen sensible oder persönliche Daten übertragen werden, Relevanz. Ein bereits bestehender Standard des W3Cs, welcher hierfür genutzt werden kann ist XML Encryption (Eastlake u. Reagle, 2002).

2.2.2 Semantic Desktop

Die Idee des *Semantic Desktop* ist aus der Semantic-Web-Bewegung gewachsen und hat zum Ziel Technologien, die für das Semantic Web entwickelt wurden, für Desktop-Computing nutzbar zu machen. Laut Sauermann u. a. (2005) wurde der Begriff des Semantic Desktop in (Decker u. Frank, 2004) geprägt und stellt einen Schritt auf dem Weg zur vollständigen (semantischen) Vernetzung des Arbeitsplatzcomputers mit dem Internet und somit mit anderen Arbeitsplatzcomputern dar.

Oft[8] wird gerade der Fokus auf das Web als zentrale Neuerung am Semantic Web gesehen. Nicht desto weniger hat die Semantic-Web-Bewegung in den letzten Jahren massiv zur Entwicklung neuer, standardisierter Sprachen zur Wissensrepräsentation beigetragen und die Entwicklung von Technologien zur Nutzbarmachung dieser Sprachen vorangetrieben. Diesen Entwicklungen nimmt sich das Semantic-Desktop-Paradigma an, um sie auf Desktop-Computing umzulegen, eine engere Verbindung zwischen Desktop und Web zu realisieren und im Desktop-Umfeld eine dem Semantic Web äquivalente Umgebung zu schaffen. Eine detailliert Einführung in das Themenfeld Semantic Desktop ist in (Sauermann u. a., 2005) zu finden.

Aspekte des Ansatzes Semantic-Web-Technologien für den Desktop bzw. für die Unternehmung zu nutzen sind nicht nur unter dem Namen *Semantic Desktop* bekannt. So präsentieren Maedche u. a. (2003) eine Architektur für ein *ontology-based knowledge management system* für den Einsatz im Unternehmen. Dieses System kann als Anwendung für den Semantic Desktop betrachtet werden, da Technologien, welche aus dem Semantic-Web-Umfeld entstanden sind, zur Realisierung des Systems herangezogen werden. Auch Uren u. a. (2006) betrachten die Verwendungen von Semantic-Web-Technologien für den Einsatz im Wissensmanagement. Uren u. a. (2006)

[8] `http://www.w3.org/2001/sw/SW-FAQ#weburi` (18.05.2008)

sehen den Akt der semantischen Annotation von Dokumenten (also der Anreicherung von Dokumenten mit Metadaten aus einer Ontologie) als einen zentralen Nutzen für eine (Dokument-zentrierte) Wissensmanagement-Strategie im Unternehmen. Sie untersuchen bestehende Ansätze zur semantischen Annotation und definieren Anforderungen an zukünftige Systeme unter dem Blickwinkel des Wissensmanagements.

Im Rahmen dieser Arbeit werden Systeme, welche Semantic-Web-Technologien in einer Desktop-Umgebung nutzen als Systeme für den Semantic Desktop bezeichnet. Diese Systeme zeichnen sich dadurch aus, dass durch die Verwendung von Technologien für das Semantic Web eine dem Semantic Web ähnliche Umgebung geschaffen wird und effizienter eine Verbindung mit dem Semantic Web hergestellt werden kann, als es auf Basis anderen Technologien möglich wäre.

2.2.3 Ontologien

Begriffsdefinition

Der Ontologiebegriff hat seine Wurzeln in der Philosophie, dort ist die Ontologie die Lehre von der Organisation der Welt. In den vergangen Jahren wurde der Begriff der *Ontologie* auch in der Computerwissenschaft, als Werkzeug zur Wissensrepräsentation populär. So schreiben Berners-Lee u. a. (2001):

> *„In philosophy, an ontology is a theory about the nature of existence, of what types of things exist; ontology as a discipline that studies such theories. Artificial-intelligence and Web researchers have co-opted the term for their own jargon, and for them an ontology is a document or file that formally defines the relations among terms. The most typical kind of ontology for the Web has a taxonomy and a set of inference rules."*
>
> Berners-Lee u. a. (2001)

Häufig wird in der Informatik auf die Ontologie-Definition von Gruber (1993) zurückgegriffen:

> *,,An ontology is an explicit specification of a conceptualization."*
>
> Gruber (1993)

Bei einer *Konzeptualisierung* (engl. conceptualization) handelt es sich um eine abstrakte, vereinfachte Darstellung der Welt. Nur die für eine bestimmte Anwendung relevanten Teile der Welt werden als Konzepte in das Modell übernommen und miteinander in Verbindung gebracht. Jede Anwendung die die Welt zu einem gewissen Grad modelliert, verwendet implizit oder explizit eine Konzeptualisierung (Gruber).

Die Bedeutung der Konzeptualisierung im Zusammenhang mit der Erstellung von Ontologien wird auch aus folgender Definition ersichtlich:

> *,,,Ontology' is the term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve the above problems ... An ontology necessarily entails or embodies some sort of world view with respect to a given domain. The world view is often conceived as a set of concepts (e.g. entities, attributes, processes), their definitions and their inter-relationships, this is referred to as a conceptualisation."*
>
> Uschold u. Grüninger (1996)

Somit handelt es sich bei einer Ontologie um ein Modell eines Teiles der Realität, einer oder mehrerer Personen. Die Modellbildung erfolgt durch Menschen, im letzten Ende in deren Gehirnen. Dieser Schritt wird als Konzeptualisierung bezeichnet. Durch die Externalisierung des internen Modells wird eine Ontologie erzeugt, die Konzeptualisierung wird explizit spezifiziert. Dies kann in Form eines Diagramms auf Papier sein oder mittels einer formalen Sprache wie OWL (vgl. Abschnitt 2.2.1) erfolgen. Bei beidem handelt es sich um die explizite Spezifikation einer Konzeptualisierung und somit um eine Ontologie. Für deren Anwendung im Semantic Web muss allerdings zur Spezifikation von Ontologien ein Formalismus verwendet werden, der es Maschinen ermöglicht deren Inhalt zu interpretieren (vgl. hierzu auch Abschnitt 2.2.3).

Bei Ontologien im Kontext des Semantic Web findet ein objektorientiertes Modellierungsparadigma Verwendung. Dinge in der Welt werden als Klassen und Instanzen dieser Klassen modelliert. Klassen sind Mengen von Dingen in der Welt mit gleichen Eigenschaften. Die Eigenschaften einer Klasse und die Relationen zwischen Klassen können bei der Spezifikation einer Ontologie festgelegt werden.

Nutzen von Ontologien

Der Prozess der Erstellung eines Modells einer Domäne ist mit Aufwand verbunden. Im Folgenden werden Argumente für die Erstellung von Ontologien angeführt, um den Nutzen der expliziten Spezifizierung einer Konzeptionalisierung zu verdeutlichen. Hierbei wird nach den von Noy u. McGuinness (2001) identifizieren Gründen für das Erstellen einer Ontologie vorgegangen. Zusätzlich sei auf (Gruber, 1993) und (Uschold u. Grüninger, 1996) hingewiesen, welche ähnliche Argumente für den Nutzen von Ontologien anführen.

Gemeinsames Verständnis der Struktur von Information zwischen Personen oder Software-Agenten: In der Ontologie wird ein einheitliches Vokabular für eine Domäne festgelegt und Relationen zwischen den Begrifflichkeiten definiert. Dies ermöglicht den Anwendern der Ontologie sich (effizienter) in noch nicht bekannten Informationen zurechtzufinden und Missverständnisse zu vermeiden. So kann ein Agent schneller ein Buch in einer (digitalen) Bibliothek auffinden, wenn die Bücher in einer ihm vertrauten Form geordnet sind.

Wiederverwendbarkeit von Domänenwissen: Wurde bereits eine Domäne von Experten modelliert, so kann auf dieses Modell zurückgegriffen werden. Bereits Vorhandenes kann also weiter verwendet werden. Hier sei angemerkt, dass existierende Ontologien ausschließlich das Weltbild ihrer Ersteller widerspiegeln. Haben zwei Gruppen eine unterschiedliche Sichtweise auf eine Domäne wird die Wiederverwendung von Wissen deutlich erschwert.

Explizit-Machen von Hypothesen in Bezug auf eine Domäne: Werden Annahmen zu einer Domäne in einer Ontologie-Schicht gekapselt, so können diese gegebenenfalls leichter verändert werden, als wenn diese Hypothesen beispielsweise fest im Quelltext eines Programms festgelegt

wären. Annahmen sind somit leichter auffindbar und können auch von Nicht-Programmierern gewartet werden.

Trennung von Domänenwissen und betriebsbedingtem Wissen: Wird anstatt Domänenwissen und Algorithmus in einem Programm zu kombinieren, die Domäne als Ontologie modelliert, so kann ein und dasselbe Programm auf unterschiedliche Domänen angewandt werden. Als Beispiel sei eine Fertigungslinie in einer Automobilfabrik genannt. Durch Änderung des Modells des zu fertigenden Produkts in der Steuerungssoftware der Roboter kann deren Funktionsweise angepasst und andere Produkte können gefertigt werden. Die Funktionsweise des Roboters bleibt allerdings die gleiche. In der Software ist die Domänen-Ontologie für das zu fertigende Produkt modelliert, während die Roboter das betriebsbedingte Wissen enthalten.

Analyse von Domänenwissen: In einer Ontologie werden explizit die Zusammenhänge in einer Domäne modelliert. Dadurch kann das Verständnis derselben unterstützt werden. Nach (Uschold u. Grüninger, 1996) können Ontologien somit beispielsweise dazu beitragen Anforderungen an ein (Software-)System abzuleiten, welches in dieser Domäne operieren soll.

Formalisierungsgrad von Ontologien

Die in Abschnitt 2.2.3 angeführten Definitionen des Ontologiebegriffs lassen offen in welchem Formalisierungsgrad diese Konzeptualisierung der Welt zu erfolgen hat. Es gibt unterschiedliche Ansätze welche formalen Kriterien das Modell der Wirklichkeit erfüllen muss um als Ontologie zu gelten.

Leo Obrst schreibt in einer Nachricht an eine Mailingliste[9]:

> *,,With respect to definitions of ontologies, I hope to send a portion of a briefing I made at the Army Knowledge Management Conference in Ft. Lauderdale late Aug/early Sept of 2004, that takes you through the ontology spectrum, from taxonomy (weak and strong) to thesaurus (a strong term taxonomy+) to conceptual model (weak ontology) to logical theory (strong ontology). The first is unstandardized, the second and third each has a set of standards associated with them, the*

[9] `http://ontolog.cim3.net/forum/ontolog-forum/2005-03/msg00005.html` (18.05.2008)

> *third and fourth have multiple representation languages supporting them, and the last has some logic behind the representation language, typically ranging from a description logic (OWL) to first-order logic (KIF, Common Logic) to a higher order logic.*
>
> *A logical theory is a formal ontology. The others range from informal to semi-formal. Other informal ontologies can be natural language sentences in a document. The key point about formal ontologies (logical theories) is that they are machine-interpretable, i.e., semantically interpretable by machine. The others are not, are only interpretable by human beings, though they may be machine-readable and machine-processable."*
>
> Obrst (2005)

Die Nachricht zeigt die Breite des *Ontologie-Spektrum*s, also die Vielzahl der Möglichkeiten zur Wissensrepräsentation welche als Ontologie gelten können. Ein zentraler Punkt in der oben angeführten Nachricht ist, dass nur formale Ontologien maschineninterpretierbar, und somit für das Semantic Web nutzbar sind. Eine detailliere Erläuterung der oben angeführten Materie ist in (Daconta u. a., 2003) zu finden.

McGuinness (2003) präsentiert ein Ontologie-Spektrum (siehe Abbildung 2.2), welches von Wortmengen über Thesauri und Klassenhierarchien (is-a) bis hin zu beliebigen logischen Konstrukten reicht. Dieses Spektrum enthält eine Trennlinie in der Mitte, welche informale (links) von formalen Ontologien (rechts) trennt. Nur formale Ontologien werden in (McGuinness, 2003) als Ontologien gesehen. Diese können sicher für das Ziehen von Schlüssen genutzt werden.

Das breite Spektrum in dem der Ontologiebegriff angesiedelt ist, spiegelt sich auch in der Semantic-Web-Bewegung wider. So wird dort diskutiert wie hoch der Formalisierungsgrad der semantischen Schicht über dem bisherigen Web ausfallen soll (vgl. Abschnitt 2.2.4).

Beispiel für die Anwendung von Ontologien im Semantic Web

Anhand des in der Einleitung von Abschnitt 2.2 beschriebenen Anwendungsfalls für das Buchen einer Reise über das Web durch einen Agenten soll nun

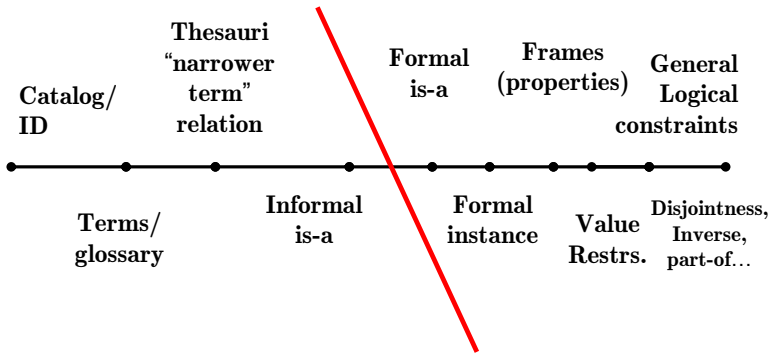

Abbildung 2.2: Das Ontology Spectrum nach McGuinness (2003)

ein Beispiel der Anwendung von Ontologien im Kontext des Semantic Web gegeben werden.

Ontologien für diesen Anwendungsfall müssen jene Informationen beinhalten welche es Agenten ermöglichen Schlüsse auf den vorhanden Daten zu ziehen. Ein Anbieter von Reisen über das Internet könnte also jenen Datenbestand der derzeit bereits in seinen Datenbanken vorhanden ist über das Web zugänglich machen. Neben den Daten als solches wird zusätzlich auch das Schema der Datenbank veröffentlicht, dieser Datenbestand ist mit einer Ontologie gleichsetzbar. Tabellen in der Datenbank können als Klassen betrachtet werden, Spalten in der Datenbank als Eigenschaften der Klassen. Zeilen in Tabellen der Datenbank als die Instanzen dieser Klassen.

Tabelle 2.1 zeigt ein Beispiel einer Tabelle, welche Reiseangebote enthält. Diese Tabelle kann als (Teil einer) Ontologie aufgefasst werden, da sie einen Ausschnitt der Welt für einen bestimmten Anwendungsfall repräsentiert. Als Ontologie betrachtet gibt es eine zur Tabelle äquivalente Klasse *Reise*, deren Instanzen die Eigenschaften *Ort*, *Anreise*, *Abreise* und *Preis* haben. Ontologie-Sprachen für das Semantic Web erlauben es nun, sowohl Informationen über Schema der Tabelle als auch über die Instanzen im Web verfügbar zu machen.

Ein Agent der Daten von Reiseanbietern auswerten soll, muss nun in der Lage sein die Semantik dieser Daten zu interpretieren. Beispielsweise muss die Bedeutung der Spalte bzw. der Eigenschaft *Ort* klar interpretierbar sein. In dieser Hinsicht ist ein Agent für das Semantic Web identisch mit aktu-

Ort	Anreise	Abreise	Preis
London	21.2.2008	24.2.2008	200
Wien	21.2.2008	23.2.2008	100
Rom	22.2.2008	27.2.2008	220
Mailand	23.2.2008	28.2.2008	500
⋮	⋮	⋮	⋮

Tabelle 2.1: Beispielhafte Datenbanktabelle mit Reiseangeboten

ellen Anwendungen, welche mit Daten aus einer Datenbank arbeiten, auch hier muss die Semantik der Daten in der Datenbank für die Programmschicht eindeutig festgelegt sein. Soll der Agent Daten von verschiedenen Reiseanbietern verarbeiten können, muss dieser in der Lage sein Daten aus unterschiedlichen Datenquellen miteinander zu vereinen. Auch diese Situation ist im Kontext von großen Unternehmensanwendungen nicht neu, auch hier ist es nötig Informationen aus unterschiedlichen Quellen miteinander zu vereinen. Die Herausforderung im Semantic Web liegt nun darin, diese Unterfangen in einer offenen Umgebung wie dem Web zu realisieren.

Ontologiesprachen für das Semantic Web ermöglichen es Daten im Web zu veröffentlichen. Teilweise wird damit auch die Semantik dieser Daten eindeutig interpretierbar, beispielsweise legt OWL klar fest wie sich eine Klasse zu einer Instanz verhält. Für spezielle Anwendungen, wie die Abbildung von Reisen, muss allerdings der Agent die Fähigkeit besitzen Eigenschaften von Klassen zu interpretieren, da deren Semantik nicht durch OWL festgelegt ist. Z.B. muss das Computerprogramm die Eigenschaft *Ort* richtig interpretieren können, um darauf aufbauend eine weitere Verarbeitung der Daten gewährleisten zu können. Dies wird nicht durch Technologien des Semantic Web übernommen. Ebenfalls sind Methoden zur Integration von Daten aus verschieden Datenquellen mit unterschiedlichen Schemata derzeit noch Gegenstand der Forschung im Alignment bzw. Mapping von Ontologien (Euzenat u. Shvaiko, 2007).

Daten die dazu dienen eine Reise zu beschreiben können wiederum auf Information in anderen Ontologie verweisen. So können zur Identifikation von Orten Identifikatoren aus einer anderen Ontologie verwendetet werden,

welche geographische Informationen enthält, beispielsweise, dass die Stadt London in England liegt. Dies ermöglicht es Agenten, die vom Reiseanbieter erhaltenen Daten mit zusätzlichen Informationen anzureichern. Diese Art der Verweise, über Ontologiegrenzen hinweg, wird ebenfalls durch Ontologiesprachen für das Semantic Web unterstützt.

2.2.4 Formalisierungsgrad des Semantic Web

Wie in der Einleitung von Kapitel 2.2 beschrieben wurde, ist es ein erklärtes Ziel der Semantic-Web-Bewegung Informationen im Web eine wohldefinierte Bedeutung zu geben (*information is given well-defined meaning*). Zu diesem Zweck soll eine semantische Schicht über dem bisherigen Web geschaffen werden. Derzeit gibt es allerdings (noch) unterschiedliche Sichtweisen, wie formal diese semantische Schicht ausfallen soll (Hendler, 2007): Am einen Ende des Spektrums steht die Ansicht, dass ein starker Formalisierungsgrad der Metadaten zu den Ressourcen im Web nötig ist und Zusammenhänge zwischen diesen Metadaten über Ontologien (in OWL-DL) beschrieben werden sollen.

Am anderen Ende des Spektrums steht der Ansatz die Zusammenhänge zwischen Informationsobjekten weniger formal zu beschreiben (zum Beispiel mittels RDF). Diese niedrige Formalisierung sei ausreichend um den ersten Schritt in Richtung Semantic Web zu tun. Der Fokus solle darauf gelegt werden, semantische Beschreibungen in naher Zukunft in großer Vielfalt im Web verfügbar zu machen.

Ein oft gebrauchtes Zitat in diesem Zusammenhang, welches auch das Kredo der vorliegenden Arbeit bildet, stammt von James Hendler:

> *,,A Little Semantics Goes A Long Way"*
>
> James Hendler[10]

Die unterschiedlichen Ansichten über den Formalisierungsgrad im Semantic Web sind mit dem unterschiedlichen Formalisierungsgrad von Ontologien vergleichbar, auf den in Abschnitt 2.2.3 eingegangen wird. So wie der

[10] `http://www.mindswap.org/blog/2006/12/13/the-dark-side-of-the-semantic-web/` (18.05.2008)

Formalisierungsgrad von Ontologien in der Form eines Spektrums abgebildet werden kann (vgl. Abbildung 2.2), existiert für den Formalisierungsgrad des Semantic Web ein fließender Übergang. Wichtig für die Umsetzung des Semantic Web ist allerdings die Maschinen-Interpretierbarkeit der semantischen Schicht über dem bisherigen Web. Aus diesem Grund kommen nur formale Ontologien, in Abbildungen 2.2 auf der rechten Seite zu finden, für die Realisierung des Semantic Web in Frage.

2.3 Information Retrieval

Information Retrieval, zu Deutsch auch *Informationswiedergewinnung* genannt, beschäftigt sich mit dem (Wieder-)auffinden von Information, sei es in Digitalen Bibliotheken, im Internet oder am Arbeitsplatzrechner. Baeza-Yates u. Ribeiro-Neto (1999) bezeichnen diesen Prozess als die Befriedigung des *Informationsbedürfnis* (engl. *information need*) des Benutzers eines Information-Retrieval-Systems.

Für eine umfassende Einführung in das Thema Information Retrieval sei auf (Rijsbergen, 1979), (Baeza-Yates u. Ribeiro-Neto, 1999), (Ferber, 2003) und (Fuhr, 2006) verwiesen.

Wie in (Ferber, 2003), (Fuhr, 2006) und (Granitzer, 2006) soll auch hier die Definition der Fachgruppe Information Retrieval (FGIR[11]) der Gesellschaft für Informatik (GI[12]) angeführt werden:

> *Im Information Retrieval (IR) werden Informationssysteme in bezug auf ihre Rolle im Prozeß des Wissenstransfers vom menschlichen Wissensproduzenten zum Informations-Nachfragenden betrachtet. Die Fachgruppe „Information Retrieval“ in der Gesellschaft für Informatik beschäftigt sich dabei schwerpunktmäßig mit jenen Fragestellungen, die im Zusammenhang mit vagen Anfragen und unsicherem Wissen entstehen. Vage Anfragen sind dadurch gekennzeichnet, daß die Antwort a priori nicht eindeutig definiert ist. Hierzu zählen neben Fragen mit unscharfen Kriterien insbesondere auch solche, die*

[11] `http://www.uni-hildesheim.de/fgir/` (18.05.2008)
[12] `http://www.gi-ev.de/` (18.05.2008)

nur im Dialog iterativ durch Reformulierung (in Abhängigkeit von den bisherigen Systemantworten) beantwortet werden können; häufig müssen zudem mehrere Datenbasen zur Beantwortung einer einzelnen Anfrage durchsucht werden. Die Darstellungsform des in einem IR-System gespeicherten Wissens ist im Prinzip nicht beschränkt (z.B. Texte, multimediale Dokumente, Fakten, Regeln, semantische Netze). Die Unsicherheit (oder die Unvollständigkeit) dieses Wissens resultiert meist aus der begrenzten Repräsentation von dessen Semantik (z.B. bei Texten oder multimedialen Dokumenten); darüber hinaus werden auch solche Anwendungen betrachtet, bei denen die gespeicherten Daten selbst unsicher oder unvollständig sind (wie z.B. bei vielen technisch-wissenschaftlichen Datensammlungen). Aus dieser Problematik ergibt sich die Notwendigkeit zur Bewertung der Qualität der Antworten eines Informationssystems, wobei in einem weiteren Sinne die Effektivität des Systems in bezug auf die Unterstützung des Benutzers bei der Lösung seines Anwendungsproblems beurteilt werden sollte.

Fachgruppe Information Retrieval der Gesellschaft für Informatik

Information-Retrieval-Systeme sind somit eine Untergruppe von Informationssystemen. Aus der oben angeführten Definition geht die Breite des Gebiets Information Retrieval hervor. Ferber (2003) sieht *Vagheit*, *Unschärfe* oder *Unsicherheit* als zentrale Charakteristika des Information Retrieval nach dieser Definition. Information Retrieval baut nicht nur auf die statistischer Analyse von Text-Dokumenten, sondern auch Elemente aus der Wissensrepräsentation wie Fakten, Regeln und semantische Netze können nach der Definition der GI Teil eines Information-Retrieval-Systems sein.

Abbildung 2.3 zeigt ein allgemeines Modell zum Information Retrieval. Das hier präsentierte Modell ist ein überarbeitete Version[13] des in (Kuropka, 2004) zu findenden Modells. Es formalisiert die in (Rijsbergen, 1979), (Baeza-Yates u. Ribeiro-Neto, 1999) und (Fuhr, 2006) textuell vorliegenden Beschreibungen zu Information Retrieval auf grafische Weise.

[13] Zusätzlich zur inhaltlichen Überarbeitung findet ein Aktivitätsdiagramm der UML Verwendung um das Modell zu repräsentieren.

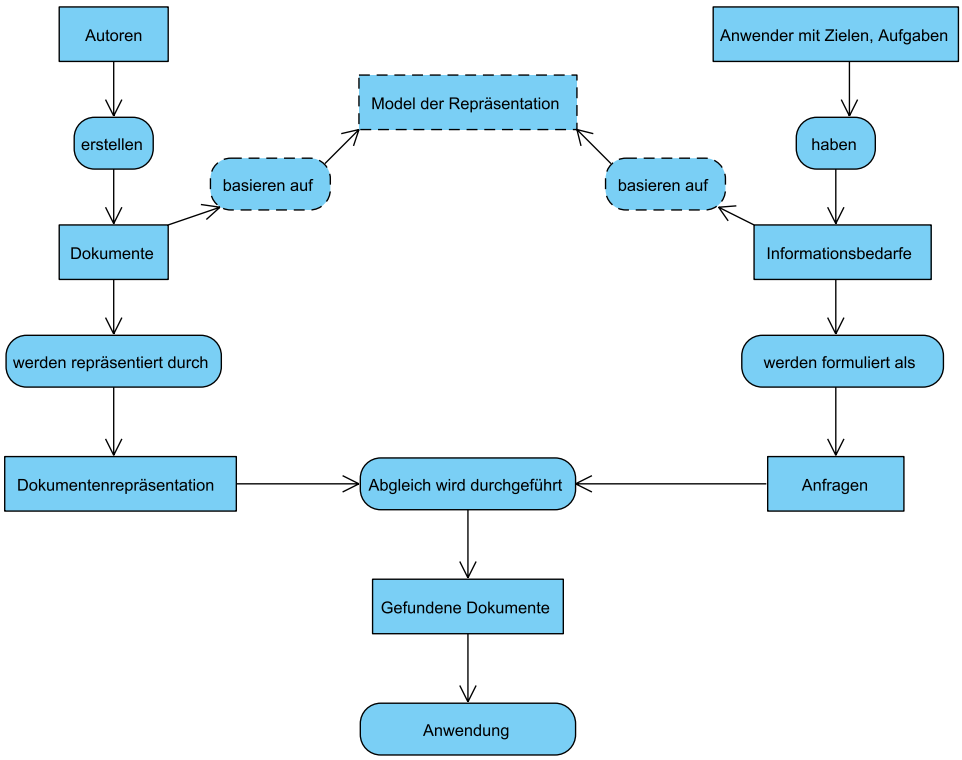

Abbildung 2.3: Ein allgemeines Modell zum Information Retrieval (nach (Kuropka, 2004))

Im Information Retrieval sind zwei (gegebenenfalls auch unterschiedliche) Personengruppen beteiligt, Autoren und Anwender. Die Gruppe der Autoren erstellt Dokumente, welche in das Information-Retrieval-System eingepflegt werden. Die Aktivität des Einstellens in das System kann aktiv erfolgen, z.B. durch Speichern in einem speziellen Ordner, oder auch passiv, beispielsweise, wenn das Information-Retrieval-System den Email-Verkehr einer Firma ausliest, oder eine Internetsuchmaschine Webseiten indiziert.

Die im System vorhandenen Dokumente werden für die interne Verarbeitung in eine Dokumentenrepräsentation umgewandelt, welche auf einem dem System zugrunde liegenden Modell der Repräsentation basiert. Wurde als Modell der Repräsentation im System beispielsweise das Vektorraummodell (vgl. Abschnitt 5.2) gewählt, werden die sich im System befindlichen Dokumente in Form von Vektoren repräsentiert.

Die zweite Personengruppe im Information Retrieval ist die Gruppe

der Anwender. Diese haben bestimmte Ziele bzw. Aufgaben, zu deren Bewältigung sie bestimmte, von ihrer gegenwärtigen Situation abhängige, Information benötigen. Man spricht hier vom Informationsbedarf der Anwender. Diesen Informationsbedarf drücken sie durch Anfragen an das System aus. Die Repräsentation der Anfragen im System basiert wiederum auf dem, dem System zugrunde liegenden Modell der Repräsentation.

Nachdem eine Anfrage an das Information-Retrieval-System geschickt wurde, führt dieses einen Abgleich zwischen der Anfrage und der Menge an Dokumenten im System durch. Daraufhin retourniert das System die Liste jener Dokumente, welche zur Anfrage des Benutzers passen. Diese Dokumente können nun vom Benutzer des Information-Retrieval-Systems für die Bewältigung seiner Aufgaben herangezogen werden.

2.3.1 Information Retrieval und Data Retrieval

Literatur zum Thema Information Retrieval ist oft bemüht eine Abgrenzung zum *Data Retrieval* zu geben. So auch Baeza-Yates u. Ribeiro-Neto (1999):

> *Data retrieval, in the context of an IR system, consists mainly of determining which documents of a collection contain the keywords in the user query which, most frequently, is not enough to satisfy the user information need. In fact, the user of an IR system is concerned more with retrieving information about a subject than with retrieving data which satisfies a given query. A data retrieval language aims at retrieving all objects which satisfy clearly defined conditions such as those in a regular expression or in a relational algebra expression. Thus, for a data retrieval system, a single erroneous object among a thousand retrieved objects means total failure. For an information retrieval system, however, the retrieved objects might be inaccurate and small errors are likely to go unnoticed. ...*

> *Data retrieval, while providing a solution to the user of a database system, does not solve the problem of retrieving information about a subject or topic. To be effective in its attempt to satisfy the user information need, the IR system must somehow 'interpret' the contents of the information items (documents) in a collection and rank them according to a degree of relevance to the user query.*
>
> Baeza-Yates u. Ribeiro-Neto (1999)

Ein zentraler Punkt des Information Retrieval im Gegensatz zum Data Retrieval ist die Interpretation der Informationsobjekte und deren Reihung abhängig von ihrer Relevanz für den Nutzer.

2.4 Zusammenfassung

Dieses Kapitel hat die Begriffe Semantic Web und Information Retrieval vorgestellt. Somit wurde die Grundlage für Kapitel 3 geschaffen, welches auf dem hier Angeführten aufbaut und existierende Ansätze zum Information Retrieval im Semantic Web und am Semantic Desktop vorstellt. Ebenfalls wurde in diesem Abschnitt der Ontologie-Begriff eingeführt und das Konzept des Semantic Desktop beleuchtet, welche beide im weiteren Verlauf dieser Arbeit Verwendung finden.

3 Information Retrieval im Semantic Web

3.1 Einleitung

Dieses Kapitel untersucht Information Retrieval im Kontext des Semantic Web. Neben der Betrachtung des in der Literatur postulierten Nutzens von Semantic-Web-Technologien für Information Retrieval erfolgt ein Überblick über aktuelle Suchansätze für das Semantic Web und den Semantic Desktop. Um eine fundierte Basis für eigene Entwicklungen zu schaffen wird eine Vielzahl von Systemen untersucht. Abschließend werden die untersuchten Suchansätze miteinander verglichen und es wird eine Definition von Information Retrieval im Semantic Web gegeben.

3.2 Schwächen gegenwärtiger Suchansätze

Antoniou u. van Harmelen (2004) sehen Web-Suchmaschinen, welche auf der Suche nach Schlüsselwörtern basieren, wie AltaVista[1], Google[2] oder Yahoo[3], als eines der zentralen Elemente des derzeitigen Webs an. Trotzdem lassen sich nach wie vor Probleme in Zusammenhang mit den derzeit für die Suche im Web verwendeten Technologien identifizieren (Antoniou u. van Harmelen, 2004):

[1] `http://www.altavista.com/` (18.05.2008)
[2] `http://www.google.com/` (18.05.2008)
[3] `http://www.yahoo.com/` (18.05.2008)

1. *Hoher Recall, niedrige Precision.*[4] Es werden zwar relevante Seiten zu einer Anfrage gefunden, allerdings auch eine hohe Menge andere, nicht relevante Seiten. Zuviel kann in diesem Zusammenhang ähnlich schlecht wie zu wenig sein.
2. *Niedriger oder kein Recall.* Keine Seiten die zur Anfrage passen werden gefunden oder relevante Seiten werden nicht gefunden. Obwohl eher selten, ist dieses Problem vorhanden und kann mit dem folgenden Punkt korrelieren.
3. *Resultate sind stark abhängig vom verwendeten Vokabular.* Oftmals werden mit der ersten Anfrage nicht genau die Seiten gefunden, die der Benutzer erwartet, da relevante Dokumente eine andere Terminologie enthalten als jene, die der Benutzer in der Anfrage verwendet. Auf die semantische Ähnlichkeit von Wörtern wird nicht eingegangen.
4. *Resultate sind einzelne Web-Seiten.*[5] Wenn Information benötigt wird, welche über mehrere Dokumente verteilt ist, muss der Benutzer diese Information selbst zusammentragen, oftmals unter Verwendung mehrerer Anfragen.

Ähnlich zu den von Antoniou u. van Harmelen (2004) beschrieben Problemen bei der aktuellen Suche im Web sehen Davies u. a. (2003) Defizite an gegenwärtigen Suchansätzen, welche im Unternehmensumfeld eingesetzt werden. Sie zählen folgende Schwächen aktueller Suchansätze in Wissensmanagement-Systemen auf:

1. *Irrelevante Information wird gefunden*, da bestimmte Wörter unterschiedliche Bedeutungen haben.
2. *Relevante Information wird nicht gefunden*, wenn unterschiedliche Wörter für einen gleichbedeutenden Inhalt verwendet werden.

[4] Eine detaillierte Einführung der Evaluierungsmaße Recall und Precision erfolgt in Abschnitt 8.2.1. Im Kontext von (Antoniou u. van Harmelen, 2004) wird unter Recall die Menge an relevanten Dokumenten in einem Suchergebnis verstanden und unter Precision die Menge der relevanten Dokumente im Verhältnis zur Menge aller Dokumente im Suchergebnis.

[5] Aktuelle Web-Suchmaschinen liefern natürlich nicht nur Web-Seiten sondern auch andere Dokumenttypen wie beispielsweise PDFs. Das Problem, dass Information über mehrere Ergebnisse verteilt ist bleibt allerdings bestehen.

3. Traditionelle Suche fokussiert auf den Zusammenhang zwischen einer Suchanfrage und einer Menge an Information. Die *Zusammenhänge zwischen Informationen* im System werden allerdings *nicht betrachtet.*

3.3 Chancen durch die Verwendung von Semantic-Web-Technologien

Heflin u. Hendler (2000) gehen davon aus, dass durch die Verwendung von semantischem Markup die Suche im Web verbessert werden kann:

> *However, if machines could "understand" the content of web pages, then searches with high precision and recall would be possible.*
>
> Heflin u. Hendler (2000)

Auch Iofciu u. a. (2005) sehen Vorteile für den Fall, in dem Metadaten zu Ressourcen, in der Form von RDF-Statements, in den Suchprozess nach Ressourcen eingebunden werden. Im Vergleich zur reinen Volltext-Suche könnten mehr Ergebnisse geliefert werden (also der Recall erhöht werden) und eine bessere Reihung der Suchergebnisse erzielt werden (also die Precision erhöht werden).

Ähnlich zum Precision/Recall-Argument von Iofciu u. a. (2005) führen Guha u. a. (2003) folgende Möglichkeiten bei der Verwendung einer *semantischen Suche*[6] an:

1. Die *Erweiterung der Ergebnisse* traditioneller Suchmaschinen durch Daten aus dem Semantic Web. Beispielsweise könnten Suchergebnisse zu einem Musiker mit dessen Bild, seinen Tour-Daten, oder seinen Alben angereichert werden.
2. Ein *besseres Verständnis der Anfrage* und darauf aufbauend bessere Suchergebnisse. Dies soll durch das Wissen um die Bedeutung der in der Anfrage verwendeten Wörter ermöglicht werden.

[6] Guha u. a. (2003) bezeichnen ihren Ansatz zur Suche, der in Abschnitt 3.4 vorgestellt wird, als *Semantic Search*

Davies u. a. (2003) beschreiben Vorteile durch die Verwendung von Semantic-Web-Technologien für eine Suche im Unternehmensumfeld und gehen davon aus, dass mit Hilfe von Ontologien es möglich wäre sonst isolierte Informationen zueinander in Beziehung zu setzen.

3.4 Aktuelle Ansätze zur Suche im Semantic Web und am Semantic Desktop

Im Rahmen dieser Arbeit wird eine explorative Studie zu aktuellen Suchansätzen für das Semantic Web durchgeführt. Diese dient auf der einen Seite dazu neue Richtungen für eigene Forschung zu identifizieren, auf der anderen Seite dazu die Problemdomäne zu strukturieren und die Konstruktvalidität des durchgeführten Experiments (vgl. Abschnitt 9.3.1) zu erhöhen. Es wird eine Literaturrecherche durchgeführt und die identifizierten Suchansätze und deren Methoden beschrieben. Ziel der durchgeführten Studie ist es einen Großteil der aktuell existierenden Systeme zur Suche im Semantic Web und am Semantic Desktop zu erfassen und deren Eigenschaften zu dokumentieren, allerdings kann nicht davon ausgegangen werden, dass alle gegenwärtig existierenden Systeme beschrieben werden. Basierend auf der durchgeführten Untersuchung werden abschließend eine eigene Definition von Information Retrieval im Semantic Web und eine Klassifikation der untersuchten Ansätze basierend auf deren Eigenschaften angeführt:

QUEST: *QUEST* (QUErying Semantically Tagged documents on the web) (Bar-Yossef u. a., 1999) ist eines der frühsten System zur semantischen Suche im World Wide Web. QUEST ermöglicht es nach semantischen Auszeichnungen (engl. Markup) in Web-Dokumenten zu suchen, welche mit Hilfe von *OHTML* (Kogan u. a., 1998) in diese Dokumente eingebettet wurden. OHTML erweitert HTML mit Auszeichnen, welche es ermöglichen eine Objektstruktur zu definieren und somit ein semantisches Modell zu erzeugen. QUEST stellt eine grafische Benutzerschnittstelle zur Verfügung um Anfragen zu formulieren. Mit Hilfe dieser grafischen Abfragesprache können Graphen konstruiert werden, deren Kantentypen und Knotentypen bestimmte Bedingungen erfüllen müssen. Der durch den Benutzer erstellte Anfragegraph wird dann mit der vorhandenen Datenbasis, einer Menge an OHTML-

Dokumenten, verglichen und ein Ergebnisgraph wird retourniert. Dieser Ergebnisgraph wird wiederum grafisch dargestellt.

Obwohl QUEST nicht aus dem Semantic-Web-Umfeld kommt wird in diesem System einen sehr ähnlicher Weg zu RDF und RDF-Abfragesprachen beschritten, um semantische Auszeichnungen zu Web-Dokumenten festzulegen und zu suchen.

SHOE: Einen weiteren Vorläufer zu aktuellen Systemen bildet *SHOE* (*Simple HTML Ontology Extensions*) (Heflin u. Hendler, 2000). Auch hier werden semantische Auszeichnungen in HTML-Dokumente eingebettet. Der *SHOE Knowledge Annotator* ist ein grafisches Werkzeug zur semantischen Annotation von HTML-Dokumenten. *SHOE Search* bietet eine grafische Benutzerschnittstelle für die Suche basierend auf den semantischen Auszeichnungen an.

Am Anfang des Suchprozesses steht das Auswählen der Ontologie, welche als Ausgangsbasis für die Suche verwendet werden soll. Darauf folgend wird eine Klasse (im Original als *Kategorie* bezeichnet) aus der Ontologie selektiert. SHOE Search ermöglicht es nun nach Instanzen dieser Klasse zu suchen. Zusätzlich können Werte für die Eigenschaften der gewählten Klasse festgelegt werden. SHOE Search liefert dann in tabellarischer Form all jene Instanzen zurück, bei denen die Property-Werte mit den durch den Benutzer eingegebenen Werten übereinstimmen.

SEAL (Stojanovic u. a., 2001): Stojanovic u. a. (2001) präsentieren *SEAL* (A Framework for Developing SEmantic PortALs). Eine wichtige Anforderung an ein semantisches Portal stellt laut Stojanovic u. a. (2001) die Suche nach Information dar. Da auch bei ausgefeilten Suchmöglichkeiten eine hohe Wahrscheinlichkeit besteht zuviel Information zu finden, wird ein Ansatz zur Reihung von Suchergebnissen vorgestellt.

Als Basis für die Reihung wird eine Ähnlichkeitsfunktion, der *Object Match* (OM) definiert, welcher auf der *Aufwärts-Cotopy* (*upwards cotopy* (UC)) basiert. Bei der *semantic cotopy* (SC) eines Konzepts in einer Ontologie handelt es sich um die Menge aller seiner Super- und Subkonzepte. Bei der Aufwärts-Cotopy handelt es sich um die Menge aller Superkonzepte eines Konzepts. Eine detaillierte Beschreibung der semantic cotopy und upwards cotopy ist in (Maedche u. Staab, 2002) zu finden.

Der Object Match basiert auf der Menge gleicher Superkonzepte zweier Konzepte O_1 und O_2 und somit auf deren Position in einer gemeinsamen Klassenhierarchie H. Der Object Match ist 1 wenn beide Konzepte die gleichen Superkonzepte besitzen und geht umso stärker gegen 0 je stärker sich die Superkonzepte der beiden Konzepte unterscheiden.

Um die Ergebnisse zu einer Anfrage reihen zu können wird äquivalent zum Object Match ein *Predicate Match* (PM) definiert. Hierbei handelt es sich um ein Ähnlichkeitsmaß zwischen zwei Prädikaten, das, äquivalent zum Object Match, auf deren Position in der Hierarchie der Relationen H_R aufbaut. Zusätzlich zieht der Predicate Match noch die Object-Match-Werte aller in den beiden zu vergleichenden Prädikaten vorkommenden Konzeptinstanzen (hier I_n und J_n) in Betracht. Im Predicate Match findet der geometrische Mittelwert Verwendung, daraus resultiert, dass wenn eine der Komponenten des Predicate Match gegen 0 geht, der gesamte Predicate Match gegen 0 geht.

Mit Hilfe des Predicate Match ist es nun möglich die Ergebnisse zu einer Anfrage an eine Wissensbasis zu reihen. Hierzu werden die Prädikate, welche als Ergebnis auf eine Anfrage erhalten wurden, in Abhängigkeit von ihrer Übereinstimmung mit den in der Wissensbasis enthaltenen Prädikaten gereiht[7]. Prädikate in der Ergebnismenge, welche exakt den Prädikaten in der Wissensbasis entsprechen werden höher gereiht. Für ein anschauliches Beispiel der Berechnung der Reihung auf Basis des Predicate Match sei auf (Stojanovic u. a., 2001) verwiesen.

QuizRDF: *QuizRDF* (Davies u. Weeks, 2004) (Davies u. a., 2002) kombiniert Schlüsselwort-basierte Suche mit der Suche in und der Navigation durch RDF-Annotationen von Ressourcen. Der Indizierungsprozess von QuizRDF basisiert auf so genannten Inhaltsbeschreibungen (content descriptors), welche sowohl Wörter, welche im Volltext der Dokumente vorkommen, als auch Literale aus den RDF Properties zu einer Ressource, sein können. Um indiziert werden zu können werden numerische Literale in Zeichenketten

[7] Bei den Prädikaten aus der Ergebnismenge handelt es sich um Prädikate, welche aus dem abgeleiteten Modell stammen (z.B. Superklassen-Beziehungen werden aufgelöst), während es sich bei den Relationen in der Wissensbasis um das ursprüngliche Modell handelt.

konvertiert.

QuizRDF ermöglicht es dem Benutzer eine Suchanfrage im Freitext zu formulieren. In diesem Fall wird im Volltext der Dokumente gesucht. Zusätzlich kann der Typ der Ressourcen, welche zurück geliefert werden sollen, eingeschränkt werden, bzw. können vorhandene Suchergebnisse nach diesem Typ gefiltert werden. Hierfür wird dem Benutzer eine Auswahlliste mit all jenen Klassen angeboten, zu denen sich Ressourcen in der Ergebnismenge befinden. Vor der ersten Suche werden in dieser Liste alle vorhandenen Klassen angeboten. Wird eine Klasse ausgewählt, werden alle Properties dieser Klasse angezeigt und es besteht die Möglichkeit Einschränkungen für die Eigenschaftswerte einer Ressource zu treffen und somit die Ergebnismenge weiter einzuschränken.

OWLIR: Finin u. a. (2005); Mayfield u. Finin (2003); Shah u. a. (2002) beschreiben das System *OWLIR*. OWLIR (*Ontology Web Language and Information Retrieval*) ist ein Ansatz Dokumente zu suchen, welche sowohl Text als auch semantisches Markup enthalten. Als Wissensbasis dient in OWLIR eine Ontologie, welche mit Hilfe von DAML+OIL (Connolly u. a., 2001) modelliert ist. Im in (Shah u. a., 2002) vorgestellten Anwendungsfall wird eine Ereignis-Ontologie modelliert, mit deren Hilfe Ankündigen für Veranstaltung an einer Universität, welche in textueller Form vorliegen, annotiert werden. Für diesen Annotationsprozess wird AeroText[8], ein Sprachverarbeitungswerkzeug für natürlichsprachliche Texte verwendet. WONDIR (Shah u. a., 2002) und Haircut (Mayfield u. Finin, 2003), zwei Suchmaschinen für Textdokumente, dienen als Basis für die Suchmaschine von OWLIR. *DAMLJessJB* (Kopena u. Regli, 2003) wird als Wissensbasis verwendet.

Die Autoren verfolgen den Ansatz ganze RDF-Tripel, welche als Auszeichnung zu einem Dokument dienen, zu indizieren und somit statistische Assoziationen zwischen Markup und Textdokumenten für die Suche zu nutzen. Anfragen an das System können in natürlichsprachlicher Form gestellt werden, und es dürfen komplexe Datentypen wie Nummern, geographische Orte oder Uhrzeiten verwendet werden. Wie die Verarbeitung natürlichsprachlicher Anfragen realisiert ist, wird in keiner der oben an-

[8] `http://www.lockheedmartin.com/products/AeroText/index.html` (18.05.2008)

geführten Publikationen genauer spezifiziert. DAMLJessKb wird für das Ziehen von Schlüssen in der Wissensbasis genutzt. Hierbei wird auf der einen Seite Subsumtion-Inferenz auf der Klassenhierarchie der Ontologie verwendet, auf der anderen Seite werden im Vorfeld Domänen-spezifische Regeln definiert, welche für Regel-basiertes Schließen über die Domäne genutzt werden.

SCORE (Sheth u. a., 2002): Die *Semantic Content Organization and Retrieval Engine* (*SCORE*) (Sheth u. a., 2002) wurde als kommerzielles Produkt der Firma *Voquette* in Zusammenarbeit mit der *University of Georgia* realisiert. SCORE verwendet Klassifizierungs- und Information-Extraktion-Techniken um Metadaten aus textuellen Quellen zu erzeugen. Diese Metadaten werden in weiterer Folge für die Suche verwendet. Ein Benutzer kann die Kategorie und ein oder mehrere Attributwerte der Metadaten des zu suchenden Dokuments festlegen.

Laut Sheth u. a. (2002) wurde SCORE nicht für die Volltext-Indizierung einer großen Dokumentenmenge entworfen, kann allerdings so konfiguriert werden, dass auch der textuelle Inhalt von Dokumenten indiziert wird.

DOSE: (Bonino u. a., 2003) beschreiben *DOSE* (*Distributed Open Semantic Elaboration*), eine Plattform zur semantischen Annotation von XML- oder XHTML-Dokumenten und Dokumentteilen. Ein Suchdienst ermöglicht es Anfragen an das *Annotation Repository* zu stellen.

Für die Suche wird eine Menge an Konzepten oder eine Menge an Schlüsselwörtern akzeptiert. Eine Anfrage in textueller Form wird an die Komponente *Semantic Mapper* weitergeleitet um eine Menge an Konzepten zu generieren. Diese Komponente liefert zu einer Zeichenkette eine Menge an Konzepten. Diese Konzepte dienen als eigentliche Anfrage an das Annotation Repository. Eine gereihte Liste von Annotationen wird retourniert.

In (Bonino u. a., 2004) wird detailliert auf die Suche in DOSE eingegangen. Diese basiert auf dem Vektorraummodell, die Menge aller Konzepte, die für die Annotation von Dokumenten verwendet wurden, spannen einen Vektorraum auf. Ein Anfragevektor besteht aus einer Menge an Konzepten. Es wird die Verwendung eines Query-Expansion-Ansatzes vorgestellt, welcher die Menge der Konzepte in der Anfrage um Konzepte erweitert, welche Super- oder Sub-Klassen der Konzepte in der ursprünglichen Anfrage sind.

Query Expansion wird nur verwendet, wenn die Anzahl der gefundenen Ergebnisse unter einem Schwellwert liegt. Für die Erweiterung der Anfrage wird zwischen Fokussierung (focalization) und Generalisierung (generalization) unterschieden. Bei Fokussierung wird der Query um Unterkonzepte der ursprünglichen Konzeptmenge erweitert, bei einer Generalisierung um Überkonzepte. Wann Fokussierung und wann Generalisierung verwendet wird, wird durch einen zweiten Schwellwert geregelt. Die Autoren gehen davon aus, dass, wenn die Anzahl der gefundenen Dokumente über diesem Schwellwert liegt Fokussierung der geeignete Ansatz ist. Ist die Anzahl der gefundenen Dokumente unter dem Schwellwert, stellt Generalisierung den richtigen Ansatz dar.

TAP: *TAP* wurde als eine Infrastruktur für Semantic-Web-Anwendungen entwickelt und ist als Modul für den Webserver Apache[9] realisiert. Guha u. a. (2003) beschreiben einen Ansatz zur *semantischen Suche* auf Basis von TAP:

Zusätzlich zur Anfrage an eine Web-Suchmaschine wird die Anfrage auch an TAP gesendet. Die Suche startet von einem Ankerknoten (engl. *anchor node*) im RDF-Graphen, wobei dieser Knoten zu einem der in der Suchanfrage vorkommenden Wörter korrespondieren muss. Ausgehend vom Ankerknoten wird eine Breitensuche (engl. breath first search) im RDF-Graphen durchgeführt. Guha u. a. (2003) präsentieren zwei Varianten für die Suche: (1) Eine Suche die den Kantentyp außer Acht lässt und (2) eine Suche welche den Kantentyp berücksichtigt.

In (1) werden alle Eigenschaften im Graphen gleich behandelt. Ausgehend vom Ankerknoten werden die ersten N Tripel gesammelt, wobei es sich bei N um ein vorher definiertes Limit handelt. Auch in (2) werden die ersten N Tripel gesammelt, allerdings werden hier nur jene Tripel berücksichtigt deren Prädikat einem bestimmten Typ entspricht.

Auch ein gemischter Ansatz ist laut Guha u. a. (2003) denkbar: Erst wird Ansatz (2) durchgeführt, wenn nicht genügen Tripel mit diesem Ansatz gefunden werden konnten wird Strategie (1) verfolgt.

(Soo u. a., 2003): Soo u. a. (2003) präsentieren eine Ansatz für die au-

[9] `http://httpd.apache.org/` (18.05.2008)

tomatische Annotation von und die Suche nach Bildern. Ausgehend von textuellen Beschreibungen zu Bildern werden mit Hilfe von Sprachverarbeitungstechniken und einem Fall-basierten Lernalgorithmus Bilder mit RDF-Metadaten beschrieben. Eine Anfrage an das System wird im Freitext gestellt, diese wird ebenfalls mit Sprachverarbeitungstechniken verarbeitet, um RDF-Tripel zu generieren. Eine Suche erfolgt über einen Vergleich der Tripel der Anfrage mit den Tripeln der Bilder. Wie dieser Vergleich genau stattfindet wird nicht beschrieben.

(Stojanovic u. a., 2003): Stojanovic u. a. (2003) präsentieren einen Ranking-Ansatz für Ergebnisse auf Anfragen im Semantic Web. Sie sehen gegenwärtiges Information Retrieval im Web und die Suche im Semantic Web als unterschiedliche Aufgabengebiete an, da: (1) Alle Ergebnisse einer Suche im Semantic Web formal (in der Anfrage) definierten Eigenschaften entsprechen müssen. Da jedes Ergebnis relevant für die Anfrage sein muss haben Maße wie Precision und Recall wenig Bedeutung für die Suche im Semantic Web. (2) Auch, wenn bei der Suche im Semantic Web alle Ergebnisse zu einer Anfrage relevant sind, erscheint es Stojanovic u. a. (2003) als unrealistisch, dass Applikation (bzw. die Menschen die mit diesen Applikation arbeiten) die potentiell sehr große Menge an Resultaten verarbeiten werden können. Da die Suche im Semantic Web auf das Ziehen von logischen Schlüssen in Ontologien basiert, kann diese Information verwendet werden um die Suchergebnisse zu reihen.

Für die Reihung der Ergebnisse zu einer Anfrage werden im in (Stojanovic u. a., 2003) vorgestellten Ansatz zwei Maße in Betracht gezogen. (1) Die *Spezifität* einer Relation in der Ontologie und (2) der Ableitungsprozess auf dem das Ergebnis erzielt wurde. Es wird von einer Ontologie ausgegangen, welche Regeln enthält[10], um die Ontologie zu repräsentieren wird *F-Logic* (Kifer u. a., 1995) verwendet.

Die Spezifität (vgl. (1)) einer Instanz einer Relation ist umso höher je weniger oft die Instanzen der Konzepte, welche in dieser Relation vorkommen, in Instanzen anderer Relationen vorkommen. Den maximalen Wert (in

[10]Dies ist mit der derzeitigen Version von OWL nicht möglich. Um die Verwendung von Regeln im Semantic Web zu ermöglichen wird an RIF gearbeitet (vgl. Kapitel 2.2.1).

Stojanovic u. a. (2001) ist dieser 1,0) erreicht die Spezifität, wenn die Instanzen der Konzepte, welche in ihr vorkommen in keiner anderen Instanz einer Relation vorkommen.

Für die Ableitung von Schlüssen (vgl. (2)) aus den in der Ontologie vorkommenden Regeln, wird ein *UND-ODER-Baum* aufgebaut. Die Verarbeitung des *Ableitungsbaum*s entspricht laut Stojanovic u. a. (2001) Ansätzen für semantische Netze, welche darauf beruhen, dass sich Relevanz durch den Baum ausbreitet. Die Relevanz eines Knotens resultiert aus der Relevanz seiner Kinder, wobei ein UND-Knoten einer Addition und ein ODER-Knoten einer Multiplikation von Relevanz entspricht.

Stojanovic u. a. (2003) vergleichen ihren Ansatz mit dem Ranking-Ansatz aus Stojanovic u. a. (2001) (vgl. Abschnitt 3.4), welcher ausschließlich die Klassenhierarchie einer Ontologie in Betracht zieht und kommen zum Schluss, dass der gewählte Ansatz bessere Ergebnisse liefert[11]. Der Ansatz ist im *Ontobroker*-System (Decker u. a., 1998) und im Portal des Instituts AIFB[12] implementiert.

(Bamba u. Mukherjea, 2004): Bamba u. Mukherjea (2004) präsentieren einen Ansatz für die Reihung von Suchergebnissen zu Anfragen an *ein* Semantic Web. Um jene Tripel, welche von einer RDQL-Anfrage retourniert werden zu reihen, kombinieren Bamba u. Mukherjea (2004) eine adaptierte Version von Kleinbergs HITS-Algorithmus (Kleinberg, 1999) mit zusätzlichen Gewichtungsfaktoren.

Anstatt der Hub- und Authority-Werte des HITS-Algorithmus, verwenden Bamba u. Mukherjea (2004) *Subjectivity*- und *Objectivity*-Werte (Scores) um den globalen Rang der Knoten im Graphen zu berechnen. Ein Knoten im Graph, welcher einen hohen Subjectivity-Wert aufweist, ist Subjekt in vielen Tripeln. Ein Knoten welcher einen hohen Objectivity-Wert besitzt, ist das Objekt vieler Tripel. Der ursprüngliche HITS-Algorithmus operiert auf einem Graphen, dessen Kanten alle gleich gewichtet sind. Bamba u. Mukher-

[11] Hier muss allerdings beachtet werden, dass durch die Verwendung von Regeln mehr Information in das System eingebracht wird, als es ohne Regeln der Fall wäre. Um den vorgestellte Ranking-Ansatz voll ausschöpfen zu können, muss ein Wissensmodell zum Einsatz kommen, welches Regeln enthält.

[12] `http://www.aifb.de/` (18.05.2008)

jea (2004) erweitern den HITS-Algorithmus um gewichtete Kanten, wobei die Kantengewichte von der Wichtigkeit des Kantentyps abhängig gemacht werden. Die Wichtigkeit (importance) eine Klasse hängt zusätzlich vom Subjectivity- und Objectivity-Wert noch von der Wichtigkeit ihrer Superklasse ab. Die Wichtigkeit einer Instanz hängt zusätzlich vom Subjectivity- und Objectivity-Wert noch von der Wichtigkeit ihrer Klasse ab.

Um eine Reihung der Ergebnismenge vorzunehmen, wird für jeden Graphen, welcher als Ergebnis auf die RDQL-Anfrage zurückgeliefert wird, ein Relevanzwert berechnet. Dieser wird aus den Wichtigkeitswerten der Knoten und der Kanten der Graphen zusammensetzt. Auf die Berechnung der Wichtigkeit eines Knoten wurde bereits zuvor eingegangen. Die Wichtigkeit der Kanten wird über die *inverse property frequency* bestimmt, jene Kanten mit einem Kantentyp der sehr häufig in der Wissensbasis vorkommt erhalten eine niedrigeres Gewicht. Diese Kantengewichtung ist an die *inverse document frequency* (vgl. Abschnitt 5.2.2) angelehnt. Schließlich kann noch das Ziehen von Schlüssen, mit in die Gewichtung von Ergebnis-Graphen mit einbezogen werden, auch hierfür wird ein Gewichtungsfaktor eingeführt. Dieser ermöglicht es eine hohen Anzahl von gezogenen Schlüssen sowohl positiv als auch negativ zu bewerten.

Der Reihungsansatz wird laut Bamba u. Mukherjea (2004) im *BioPatenetMiner* System eingesetzt. Welche Parametrierung dort verwendet wird, bleibt allerdings unklar. Auch wird nicht erklärt wie die Gewichtung von Typen von Eigenschaften erfolgen soll. Im Zusammenhang mit der Bewertung der Wichtigkeit des Ziehens von Schlüssen wird auf die Rolle eines *Knowledge Administrator* verwiesen, welcher diese Gewichtung für das System vornehmen soll.

KIM: *KIM* (Popov u. a., 2003) (Kiryakov u. a., 2003) (Kiryakov u. a., 2004) ist eine Plattform zur semantische Annotation, semantischen Indizierung und für semantisches Retrieval. KIM baut zu einem gewichtigen Teil auf Information-Extraction-Mechanismen auf, mit deren Hilfe Entitäten (Wörter) aus Dokumenten mit Klassen aus einer Ontologie verbunden werden. Hierzu werden die Wörter als Instanzen in die Ontologie eingefügt. Nach dem initialen Schritt der Informationsextraktion und semantischen Annotation, werden die Dokumente im System auf spezielle Art indiziert: Jede Instanz aus der Ontologie welche in einem Dokument vorhanden ist,

wird durch eine spezielle Kennung (welche kürzer als ein URI in der Ontologie ist) repräsentiert. Diese eindeutige Kennung wird vor der Indizierung in das Dokument eingefügt. Ein Beispiel für diese Vorgehensweise stellt die Passage *,,Paris a1b2c3 is full of tourists"* dar, hier repräsentiert die Zeichenkette *a1b2c3* die URI die in der Ontologie für *Paris* steht. Für Synonyme wird die gleiche Kennung verwendet, dies geschieht um eine Verbesserung der Retrieval-Leistung zu erzielen. Die Indizierung der Daten erfolgt dann mittels Lucene[13].

Anfragen an KIM werden mit Hilfe von Konzepten und Relationen aus der Ontologie formuliert. Eine beispielhafte Anfrage könnte *,,give me all documents referring to a Person that hasPosition spokesman within a Company, locatedIn a Location with name UK"* lauten. Anfragen an KIM können programmatisch über eine API oder von Hand über eine Webanwendung abgesetzt werden. Die Retrieval-Leistung von KIM wurde laut Kiryakov u. a. (2004) nicht mit der von klassischen Suchmaschinen verglichen.

InWiss: Priebe u. a. (2004) stellen eine *Search Engine for RDF Metadata* vor, welche im *InWiss* Wissensportal implementiert wurde. Der Suchansatz basiert auf der Anfragesprache SeRQL. Vor dem Absetzen einer Anfrage wird der RDF-Graph, der dem Wissensportal zugrunde liegt erweitert indem transitiven Eigenschaften gefolgt wird und die so gefunden Eigenschaftswerte direkt mit den Ressourcen verbunden werden, von denen ausgegangen wird. Der selbe Ansatz wird auch für die Erweiterung von Suchanfragen verfolgt. Für die Suche wird eine Anfrage mit SeRQL absetzt, welche mit der erweiterten Anfrage auf dem erweiterten RDF-Graphen operiert. Die Menge an Suchergebnissen wird basierend auf einem Mengen-theoretischen Ansatz gereiht: Der Relevanzwert für ein Suchergebnis wird ermittelt indem die Anzahl der übereinstimmenden Eigenschaftswerte der Anfrage und eines Suchergebnisses durch die Gesamtanzahl der Eigenschaften der Anfrage dividiert wird.

(Rocha u. a., 2004a): Rocha u. a. (2004a,b) stellen einen *Hybrid Approach for Searching in the Semantic Web* vor. Sie präsentieren zwei Prototypen, welche für die Suche auf einer Webseite Volltextsuche mit einem

[13] `http://lucene.apache.org/` (18.05.2008)

Spreading-Activation-Ansatz in einer Ontologie kombinieren.

Die Suche startet, wie bei herkömmlichen Suchansätzen für Webseiten, mit der Eingabe von Schlüsselwörtern durch den Benutzer. Diese werden als Anfrage an eine Volltext-Suchmaschine (Lucene) gesendet, welche eine gereihte Liste an Ergebnissen zurück liefert. Die Volltextsuche dient dazu die Eigenschaftswerte der Instanzen in der Ontologie zu durchsuchen. In (Rocha u. a., 2004a) werden die Instanzen der Ontologien dazu verwendet dynamische Webseiten zu generieren, somit ist der textuelle Inhalt einer Instanz zur Klasse Person ähnlich zum Inhalt einer Webseite, welche Daten über diese Person enthält. Die Suchergebnisse aus diesem Schritt bilden die Ausgangsbasis für den Spreading-Activation-Prozess.

Der Spreading-Activation-Ansatz in (Rocha u. a., 2004a,b) operiert auf einem Graph, welcher im Vorfeld aus der Ontologie erzeugt wird. Dieses Netz wird als *Hybrid Graph* bezeichnet, da es sowohl symbolische (benannte) als auch subsymbolische (gewichtete) Kanten enthält. Die symbolischen Kanten werden direkt aus den Instanzen von Relationen in der Ontologie übernommen, während die subsymbolischen Kanten durch ein Gewichtungsschema namens *Weight Mapping* erzeugt werden.

Weight Mapping berechnet ein Gewicht für eine Instanz einer Relation in der Ontologie auf Basis der Struktur der Relationen in der Wissensbasis. Rocha u. a. (2004a) stellen zwei Maße für das Weight Mapping vor, das *Cluster Measure* und das *Specifity Measure*. Darauf aufbauend präsentieren sie auch ein *Combined Measure*, welches durch die Multiplikation der beiden zuvor genannten Maße entsteht.

Der vorgestellte Ansatz wurde in zwei Prototypen erprobt. Im ersten Anwendungsfall diente eine Ontologie über universitäre Forschung als Basis für die Suche in der Webseite des *Department of Informatics* der *PUC-Rio*. Die hierbei verwendetet Ontologie bestand aus 2630 Instanzen. Im zweiten Anwendungsfall wurde eine Suche für eine Webseite über den brasilianischen Maler *Candido Portinari* entwickelt. Die zugrunde liegende Ontologie bestand aus ca. 16000 Instanzen. Welcher Formalismus für die Erstellung der Ontologien verwendet wurde, wird nicht erwähnt, nur, dass es sich um einen Formalismus handelt, welcher einfach auf RDF abgebildet werden kann (*„can be easily mapped to RDF"*).

(Bangyong u. a., 2005): Bangyong u. a. (2005) präsentieren ein Modell

für *Association Search in the Semantic Web*. Sie schlagen vor aus einer Ontologie ein Bayessches Netz zu generieren und dieses in weiterer Folge dazu zu nutzen um zu einer Menge an gefundenen Instanzen weitere Instanzen zu finden, welche mit diesen assoziiert sind.

Die initiale Suche erfolgt über traditionelle Suchtechnologie für das Web (traditional web search technology). Basierend auf der Menge an so gefunden Instanzen werden unter Zuhilfenahme des Bayesschen Netzes assoziierte Instanzen gesucht.

Bangyong u. a. (2005) gehen nicht darauf ein welche traditionelle Suchtechnologie verwendet und wie die Ontologie in ein Bayessches Netz umgewandelt werden soll.

(Song u. a., 2005): Song u. a. (2005) stellen ein *Ontology-Based Information Retrieval Model for the Semantic Web* vor, dabei gehen sie auf keinerlei Standards des Semantic Web ein. Sie schlagen vor semantische Indexterme (semantic index terms) für die Indizierung von Dokumenten zu verwenden. Dies bedeutet für unterschiedliche Wörter, welche die gleiche Bedeutung haben denselben Indexterm zu verwenden und für gleiche Wörter, welche eine unterschiedliche Bedeutung haben unterschiedliche Indexterme zu verwenden. Auch die Benutzeranfrage soll mittels semantischer Indextermen formuliert werden. Somit könnte für jede Anfrage maximaler Recall und maximale Precision erreicht werden.

Wie eine Benutzeranfrage mit Hilfe semantischer Indexterme ausgedrückt wird und wie von Wörtern im Dokument auf semantische Indexterme geschlossen werden kann, wird allerdings nicht ausgeführt.

(Zhang u. a., 2005): Zhang u. a. (2005) stellen ein *Enhanced Model for Searching in Semantic Portals* vor. Sie schlagen vor, text-basierte Suche mit einer unscharfen (fuzzy) Variante der Description Logic $\mathcal{ALC}$ zu kombinieren.

In ihrem Modell werden Anfragen an ein Information-Retrieval-System und die so gefundenen Dokumente als Teil der Wissensbasis repräsentiert. Hierzu wird ein unscharfes Konzept D_q eingeführt, welches für die Menge von Dokumenten steht, die zur Anfrage q relevant sind. Die zur Anfrage q relevanten Dokumente bilden die Instanzen zum Konzept D_q. Das Retrieval Status Value (RSV) einer text-basierten Suche gibt darüber Auskunft wie

relevant ein Suchergebnis zu einer Anfrage ist. Dieses RSV wird als Grad der Unschärfe der Instanzen des Konzepts D_q in der Wissensbasis verwendet, welche für die Anfrage q gefunden wurden.

Das Modell wurde anhand eines Semantic-Web-Service realisiert, an welches programmatisch Anfragen gestellt werden können.

CORESE: *CORESE* (Corby u. a., 2006) ist eine Ontologie-basierte Suchmaschine, welche intern auf der Basis von *Conceptual Graphs* (Sowa, 2000) arbeitet. Anfragen an CORESE werden durch ein oder mehrere RDF-Tripel formuliert. Die Abfragesprache von CORESE ist Sprachen wie SPARQL, SeRQL oder RDQL ähnlich, erlaubt allerdings zusätzliche eine approximative Suche. Diese basiert auf der semantischen Distanz zwischen zwei Klassen in einer gemeinsamen Hierarchie und dem `rdfs:seeAlso`-Property.

Die Relevanz eines Suchergebnisses wird an dessen Ähnlichkeit zur Anfrage gemessen. Wie diese Ähnlichkeit berechnet wird, wird in (Corby u. a., 2006) nicht erläutert. Auch wird nicht darauf eingegangen ob Suchergebnisse gereiht werden oder die Ähnlichkeitsberechnung dazu dient einen Schwellwert zu realisierten und nur Ergebnisse, welche diesen Schwellwert überschreiten retourniert werden.

(Choudhury u. Phon-Amnuaisuk, 2006): Choudhury u. Phon-Amnuaisuk (2006) präsentieren ein Suchsystem, welches nach ihren Aussagen auf Technologien für das Semantic Web basiert. Der vorgestellte Prototyp weißt ähnliche Eigenschaften zu dem in Rocha u. a. (2004a) präsentierten System auf, in welchem ebenfalls Schlüsselwort-basierte Suche mit einer Spreading-Activation-Suche in einer Ontologie kombiniert wird.

Dem System liegt eine Ontologie mit 6 Klassen und ca. 400 Instanzen zu Grunde, welche die Domäne der universitären Forschung modelliert. Diese Ontologie wird in einen Graph transformiert, dessen Kanten mit dem Cluster Measure aus (Rocha u. a., 2004a) gewichtet werden. Für die Schlüsselwort-basierte Suche wird Lucene verwendet, welche Daten allerdings indiziert werden geht aus (Choudhury u. Phon-Amnuaisuk, 2006) nicht hervor. Vermutlich werden, wie in (Rocha u. a., 2004a) die Eigenschaftswerte der Instanzen in der Ontologie indiziert und für die Schlüsselwort-basierte Suche verwendet.

Zu den gefunden Instanzen werden dann über die Spreading-Activation-

Suche verwandte Instanzen gesucht. Hierfür werden die über das Cluster Measure bestimmten Assoziationen zwischen Instanzen genutzt. Semantische Zusammenhänge zwischen Entitäten der Ontologie werden, im Gegensatz zu (Rocha u. a., 2004a), nicht für die Suche genutzt.

OntoSearch (Jiang u. Tan, 2006): Jiang u. Tan (2006) bezeichnen *OntoSearch* als *a Full-Text Search Engine for the Semantic Web*, allerdings sind im vorgestellten Ansatz keinerlei Technologien des Semantic Web zu finden. Die Suche in OntoSearch startet mit einer Term-basierten Anfrage durch den Benutzer. Zur Menge an Anfragetermen wird eine Liste an Dokumenten gesucht, welche die Suchterme des Benutzers beinhalten. Von diesen Dokumenten werden nun jene Konzepte, welche als Metadaten für sie vergeben wurden ermittelt. Hierbei wird davon ausgegangen, dass zu jedem Dokument eine Menge an Konzepten als Metadaten vergeben wurde.

Im nächsten Schritt folgt eine Erweiterung dieser Menge an Konzepten durch einen Spreading-Activation-Ansatz in der Ontologie. Um die Kanten zwischen den Konzepten in der Ontologie zu gewichten, wird die Häufigkeit einer Relation in der Ontologie verwendet. Häufig vorkommende Relationen erhalten ein höheres Gewicht. Die Gewichtungen werden auf den Wertebereich [0..1] normalisiert. Jiang u. Tan (2006) gehen zwar genau auf die theoretischen Grundlagen von Spreading Activation ein, führen allerdings nicht an, welche Parametrisierung für ihren Suchansatz verwendet wird.

Die Konzepte welche im Spreading-Activation-Schritt hinzugewonnen wurden, werden nach diesem Schritt, neben den ursprünglichen Suchtermen, dazu genutzt um die Menge an Ergebnissen aus der Volltextsuche zu reihen. Somit erfolgt eine Reihung nach den Suchtermen des Benutzers und zusätzlich nach jenen Konzepten, welche über Spreading Activation in der Ontologie gewonnen wurden. Für die Reihung der Suchergebnisse wird das Kosinusmaß verwendet. In jedem Dokumentvektor sind zusätzlich zu den Termen, welche im Dokument vorkommen auch noch jene Konzepte enthalten, welche als Metadaten zum Dokument vergebenen wurden. Die durch den Spreading-Activation-Schritt gewonnen Konzepte werden zum Anfragevektor hinzugefügt.

Jiang u. Tan (2006) nutzen mit dem *ACM Computing Classification Sys-*

tem (*CCS*[14]) eine wenig formale Ontologie um ihren Suchansatz zu testen. Zusätzlich enthält diese Ontologie nur zwei Typen von Relationen (*ist Unterkategorie von* und *ist verwandt mit*).

Beagle++: Eine Anwendung semantischer Suche für den (Semantic) Desktop stellt *Beagle++* dar. Beagle++ ist eine Erweiterung zur Desktop-Suchmaschine Beagle[15], welche Teil der GNOME[16] Desktop-Umgebung ist.

Beagle++ (Chirita u. a., 2006, 2005; Iofciu u. a., 2005) berücksichtigt semantische Metadaten während des Indizierungs- und Suchprozesses. Dies geschieht um (1) mehr Ergebnisse zu finden (also den Recall zu verbessern) und (2) eine bessere Reihung der Suchergebnisse zu ermöglichen (also die Precision zu erhöhen) (Iofciu u. a., 2005). Für jedes Dokument, welches indiziert wird, wird ein zusätzliches Feld für semantische Metadaten im Index angelegt. Prädikate und Objekte aller RDF-Statements, in denen das Dokument vorkommt, werden in diesem Feld gespeichert. Zusätzlich werden Prädikatpfade (in (Iofciu u. a., 2005) *predicate paths* genannt) verwendet um den Kontext des Dokuments zu beschreiben. Die Prädikatpfade werden durch jene Prädikate gebildet, die durchlaufen werden, wenn der RDF-Graph ausgehend vom Dokument traversiert wird.

Die Autoren von Beagle++ verfolgen den Ansatz, dass Menschen Dinge mit einem Kontext assoziieren, dieser Kontext soll für die Suche genutzt werden. Metadaten werden anhand des E-Mail-Kontexts, des Datei-Hierarchie-Kontexts und des Web-Cache-Kontexts gesammelt. Relationen zwischen den Elementen in einem Kontext werden als Ontologien modelliert. Beispielsweise ist eine besuchte Webseite mit der Webseite von der der Benutzer kam bevor er die Webseite besuchte und mit der Webseite zu der der Benutzer ging nachdem er die Webseite besucht hatte, verbunden (Chirita u. a., 2005). Metadaten werden aufgrund von Ereignissen im Dateisystem generiert, beispielsweise wenn eine Datei verschoben wird. Dies wird anhand der *inotify*[17] Linux-Kernel Funktionalität realisiert.

Die Suche mit Beagle++ ist ähnlich zu einer klassischen Suche reali-

[14] `http://www.acm.org/class/` (18.05.2008)
[15] `http://beagle-project.org/` (18.05.2008)
[16] `http://www.gnome.org/` (18.05.2008)
[17] `http://www.kernel.org/pub/linux/kernel/people/rml/inotify/README` (18.05.2008)

siert, allerdings wird zusätzlich zum Volltext-Index auch der Metadaten-Index durchsucht.

Beagle++ integriert die Suche anhand kontextueller Information mit ObjectRank (Balmin u. a., 2004), einem Ranking-Algorithmus ähnlich zu PageRank (Brin u. Page, 1998). Die Autoren argumentieren, dass das Hauptproblem aktueller Ansätze zur Desktopsuche das Fehlen von Ranking-Algorithmen sei, welche vergleichbar gute Ergebnisse wie PageRank liefern. PageRank kann allerdings nicht benutzt werden, da Informationen am Desktop nicht (wie im Web) miteinander verlinkt sind. Dies ändert sich mit dem Semantic Desktop, auf dem Informationen über semantischen Relationen (welche in Ontologien abgebildet sind) verbunden sind. Beagle++ kombiniert tf*idf (vgl. Abschnitt 5.2.2) mit ObjectRank um die Suchergebnisse zu reihen (Chirita u. a., 2006).

MESH: Castells u. a. (2007) bezeichnen ihren Ansatz als eine *Adaptation des Vektorraummodells für Ontologie-basiertes Information Retrieval*. Der Hauptbeitrag ihrer Arbeit liegt (1) in der Gewichtung von semantischen Annotation von Dokumenten, (2) der Verwendung dieser Gewichtung für eine Reihung der gefundenen Dokumente und (3) der Kombination dieses Maßes mit einer textbasierten Suche über Lucene.

Eine Anfrage an das in (Castells u. a., 2007) präsentierte System wird mit SPARQL (vgl. Abschnitt 2.2.1) formuliert und retourniert eine Menge an Dokumenten. Die Ergebnismenge der Anfrage wird mit Hilfe einer Variation des tf*idf Maßes (vgl. Abschnitt 5.2.2) gereiht. Ein Dokument erhält ein höheres Ranking je öfter ein Konzept aus der Anfrage in diesem Dokument vorkommt und je weniger oft das Konzept in allen anderen Dokumenten zu finden ist. Die gereihten Suchergebnisse aus diesem Schritt werden mit dem Ergebnis einer Volltextsuche kombiniert. Im Vorfeld werden die zu durchsuchenden Dokumente mit Lucene indiziert. An Lucene wird dann eine Anfrage abgesetzt, welche auf der SPARQL-Anfrage basiert. Hierzu wird der Wert der `rdf:label` Properties jener Klassen, welche (1) mit `rdf:type` in der SPARQL-Anfrage vorkommen und (2) aller Instanzen welche in der Anfrage angeführt sind, als Suchterme verwendet.

Die Ergebnisse der beiden Suchanfragen werden mittels der *CombSUM* (Lee, 1997) Strategie, einer gewichteten Linearkombination, miteinander kombiniert.

Das System wurde mit Hilfe einer erweiterten Version der Domänenontologie von KIM (vgl. Abschnitt 3.4) evaluiert und verwendet einen Testkorpus der aus Dokumenten der CNN-Webseite[18] besteht. Die Kombination von Ergebnissen aus einer Schlüsselwort-basierten Suche und einer semantischen Suche sehen Castells u. a. (2007) als schwierig an. Während die Ergebnisse der Volltextsuche das präsentierte System auch Ergebnisse finden lassen, wenn über eine rein-ontologie-basierte Suche keine Ergebnisse gefunden werden, führt dieser Ansatz gleichzeitig zu einer Verringerung der Precision des Systems.

3.4.1 Diskussion der untersuchten Systeme

Bei der Suche im Semantic Web handelt es sich derzeit um ein inhomogenes Feld. Um diesen Themenbereich für die vorliegende Arbeit zu strukturieren und die Konstruktvalidität des durchgeführten Experiments (vgl. Abschnitt 9.3.1) zu erhöhen wird im Rahmen dieser Arbeit eine explorative Studie zu aktuellen Suchansätzen für das Semantic Web durchgeführt. Der folgende Abschnitt gibt eine Zusammenfassung der Funde der durchgeführten Studie:

Wenige der aktuell für die Suche verwendeten Modelle wurden direkt für das Semantic Web entwickelt, sondern basieren auf existierenden Modellen, welche für den Umgang mit Semantic-Web-Technologien erweitert wurden. Ein Teil der untersuchten Systeme nutzen Wissensbasen um die Suchprozess zu modellieren, andere Ansätze basieren auf Information-Retrieval-Modellen, wie beispielsweise dem Vektorraummodell (vgl. Spalten WBS und IRS in Tabelle 3.1). Auch die Art wie Anfragen an Systeme formuliert werden variiert. In einem Teil der untersuchten System bestehen Anfragen aus Konzepten, in einem anderen Teil aus Wörtern (vgl. Spalten KBA und TBA in Tabelle 3.1). Viele der vorgestellten Systeme verwenden Methoden aus dem Information Retrieval um von einer Anfrage zu einer gereihten Menge an Ergebnissen zu kommen (vgl. Spalte IR in Tabelle 3.1). Die in den vorgestellten Ansätzen durchgeführten Experimente lassen vermuten, dass die Anwendung von Methoden aus dem Information Retrieval einen geeigneten Weg für die Suche im Semantic Web darstellt.

[18] `http://dmoz.org/News/Online_Archives/CNN.com/`

Generell wird in den untersuchten Ansätzen von Situationen mit vollständig vorhandenen semantischen Annotationen von Ressourcen ausgegangen (vgl. Abschnitt 1.1.2). Ansätze zur assoziativen Suche (vgl. Kapitel 4), welche semantische Ähnlichkeit zwischen Konzepten und inhaltsbasierter Ähnlichkeit zwischen Ressourcen nutzen um zusätzliche, relevante aber nicht annotierte Ressourcen zu identifizieren, sind aktuell wenig untersucht und bieten daher Potential für eigene Forschung: Suche unter Berücksichtigung semantischer Ähnlichkeit wird derzeit wenig adressiert (vgl. Spalten SA in Tabelle 3.1), obwohl mit Ontologien eine Quelle für die Berechnung von Assoziationen zwischen Konzepten vorhanden wäre. Der Großteil jener Ansätze, die Ähnlichkeiten zwischen Konzepten in die Suche einfließen lassen, nutzen diese Funktionalität zur Identifikation von Entitäten aus einer Wissensbasis, welche für eine Anfrage relevant sind, bzw. um die Relevanz einer Ressource, welche mit Konzepten annotiert ist, für eine Anfrage zu bestimmen. Nur in wenigen Fällen dient semantische Ähnlichkeit zur Erweiterung der Anfrage um Konzepte (vgl. (Rocha u. a., 2004a), (Corby u. a., 2006) oder (Bonino u. a., 2003)). Für die Bestimmung der Ähnlichkeit zwischen Konzepten existieren unterschiedliche Ansätze: Rocha u. a. (2004a) und Choudhury u. Phon-Amnuaisuk (2006) bauen auf einem strukturellen Ähnlichkeitsmaß basierend auf einem Graphen, welcher aus der Ontologie erzeugt wird, auf, Corby u. a. (2006) nutzen ein Maß für semantische Ähnlichkeit und Bonino u. a. (2003) und Priebe u. a. (2004) fügen Über- und Unterklassen der Konzepte in der Anfrage zur Anfrage hinzu. Keiner der untersuchten Ansätze nutzt die Bestimmung von inhaltsbasierter Ähnlichkeit zwischen Ressourcen zur Erweiterung der Ergebnismenge (vgl. Spalten IA in Tabelle 3.1). Maße zur Bestimmung der inhaltsbasierten Ähnlichkeit werden ausschließlich zur Berechnung der Ähnlichkeit zwischen einer textuellen Anfrage und dem textuellen Inhalt von Ressourcen verwendet. Manche der untersuchten Systeme nutzen diesen Ansatz auch zur Bestimmung der Relevanz von einem Dokument zu einer Anfrage basierend auf textuellen Annotationen der Dokumente, welche im Vorfeld indiziert werden.

Wenige der untersuchten Systeme verwenden sowohl eine Dokumentenrepräsentation als eine Wissensbasis für die Suche. Systeme die assoziative Suche in Wissensbasen nutzen, wie z.B. (Rocha u. a., 2004a) kombinieren diese nicht mit assoziativer Suche basierend auf textueller Information. Kei-

nes der vorgestellten Systeme kombiniert die Suche nach Ressourcen, die assoziative Suche nach Konzepten und die assoziative Suche nach Ressourcen in einem Modell.

Zusammengefasst lässt die durchgeführte Studie folgende Schlüsse zu:

- Ansätze aus dem Information Retrieval stellen einen geeigneten Weg für die Suche im Semantic Web dar.
- Assoziative Suchansätze sind im Kontext Suche im Semantic Web derzeit wenig untersucht und bieten daher Potential für eigene Forschung.

Eine Definition von Information Retrieval im Semantic Web

Die durchgeführte Studie lässt erkennen, dass derzeit eine verschwommene Sicht auf den Begriff des Information Retrieval im Semantic Web besteht. Für die vorliegenden Arbeit werden folgende Kriterien als notwendig für ein Information-Retrieval-System für das Semantic Web erachtet (vgl. (Scheir u. a., 2007c)):

1. Das System ist ein Information-Retrieval-System (nicht ein Data-Retrieval-System)
2. Das System basiert auf Technologien für das Semantic Web
3. Das System operiert im Semantic Web

Zu Kriterium 1: Wie bereits in Abschnitt 2.3.1 beschrieben wird in der Information-Retrieval-Literatur auf den Unterschied zu Data Retrieval hingewiesen. Ein Teil der untersuchten Ansätze zur Suche im Semantic Web retournieren keine Relevanzwerte zu den Suchergebnissen und sind somit Data Retrieval im Sinn der Definition aus Abschnitt 2.3.1 (vgl. auch (Castells u. a., 2007) für eine Diskussion). Dies trifft ebenso für SQL-ähnliche Abfragesprachen für das Semantic Web, wie die SPARQL Query Language for RDF (SPARQL) (vgl. Abschnitt 2.2.1), zu: Für eine gegebene Anfrage werden jene RDF-Tripel retourniert, welche die Anfrage erfüllen, alle Tripel sind *gleich relevant* für die Anfrage.

Zu Kriterium 2: Ein zentrales Element von Information-Retrieval-Ansätzen für das Semantic Web ist, dass sie auf Technologien basieren, welche für das Semantic Web entwickelt wurden und somit einfach in das

Semantic Web integriert werden können. Beispiele derartiger Technologien sind die Standards, welche sich im Semantic Web Stack (vgl. Abschnitt 2.2.1) wiederfinden. Nicht alle der betrachteten Systeme basieren auf Technologien für das Semantic Web (vgl. Abschnitt 2.2). Teilweise wird Ontologie-basiertes Information Retrieval und Information Retrieval für das Semantic Web in den vorgestellten Ansätzen synonym gebraucht, obwohl die beiden Konzepte nicht gleichsetzbar sind. Ontologie-basiertes Information Retrieval nutzt Ontologien für Retrieval-Zwecke, zum Beispiel um die Retrieval-Leistung zu erhöhen indem Information, welche in einer Ontologie gekapselt ist, in den Suchprozess mit eingebunden wird. Während viele Information-Retrieval-Systeme für das Semantic Web Information Retrieval basierend auf Ontologien betreiben, muss Ontologie-basiertes Information Retrieval nicht notwendigerweise auf das Semantic Web abzielen.

Zu Kriterium 3: Viele der untersuchten Systeme operieren in einer Desktop-Umgebung. Da sie nicht nach Ressourcen suchen, welche sich im Web befinden, können sie nach den oben genannten Kriterien nicht als Information-Retrieval-System für das Semantic Web bezeichnet werden. Allerdings können diese Systeme als Schritt in die Richtung von Information-Retrieval-Systemen für das Semantic Web gesehen werden. Viele der Algorithmen die derzeit für die Suche im Web Verwendung finden, wurden ursprünglich im Desktop-Umfeld erprobt und die Inkubation von Suchtechnologien im Desktop-Umfeld ist eine bekannte Situationen im Information Retrieval. Baut ein System für den Desktop auf Technologien für das Semantic Web auf (vgl. Kriterium 2), ist eine Verbindung mit dem und eine Portierung in das Semantic Web mit weniger Aufwand möglich als, wenn dies nicht der Fall ist. Derartig Systeme werden im Rahmen dieser Arbeit als Systeme für den Semantic Desktop (vgl. Abschnitt 2.2.2) bezeichnet.

Klassifikation der untersuchten Systeme

In Tabelle 3.1 werden die in diesem Abschnitt untersuchten Systeme basierend auf ihren Eigenschaften gegenübergestellt. Es wird angeführt ob der vorgestellte Suchansatz für das Web (W) entwickelt wurde oder ob die Suche am Desktop (D) abläuft. Diese Klassifikation beruht auf dem zuvor diskutiertem Kriterium 3. Weiters wird unterschieden ob es sich bei den Anfragen

an die untersuchten Modelle und Systeme um Konzepte bzw. Elemente von Wissensrepräsentationen (KBA) handelt oder ob die Anfrage aus Wörtern (Termen) besteht (TBA). Diese Unterscheidung beruht auf Kriterium 1. Auch wird unterschieden ob die Suche in einem wissensbasierten System (WBS) abläuft, also eine Wissensrepräsentation durchsucht wird, oder die Suche in einem Information-Retrieval-System (IRS) stattfindet, also eine Dokumentenrepräsentation durchsucht wird. Auch diese Klassifikation beruht auf Kriterium 1. Weiters wird aufgelistet ob der vorgestellte Ansatz Data Retrieval (DR) betreibt, also alle Ergebnisse gleich relevant sind, oder Information Retrieval (IR) durchgeführt und eine nach Relevanz gereihte Liste an Suchergebnissen retourniert wird. Diese Unterscheidung korrespondiert ebenfalls zu Kriterium 1. Ebenfalls wird angeführt ob ein System oder Modell semantische Ähnlichkeit (SA) zwischen Konzepten oder inhaltsbasierte Ähnlichkeit (IA) zwischen Ressourcen für die Suche nutzt. Auch diese Unterscheidung beruht auf Kriterium 1. Schließlich wird noch der Typ der verwendeten Wissensrepräsentation (WR) angeführt. Diese Eigenschaft korrespondiert zu Kriterium 2. Wird ein Charakteristikum unterstützt, wird dies durch ein X in der Zelle der Tabelle vermerkt. (X) signalisiert Grenzfälle die in Fußnoten erklärt werden. Für die Beschriftung der Spalten werden folgende Abkürzungen verwendet:

- W ... (Semantic) Web - Die Suche findet im Web-Umfeld statt
- D ... (Semantic) Desktop - Die Suche findet am Desktop statt
- KBA ... Konzept-basierte Anfrage - Die Suchanfrage besteht aus Konzepten bzw. Elementen einer Wissensrepräsentation
- TBA ... Term-basierte Anfrage - Die Suchanfrage besteht aus Wörtern (Termen)
- WBS ... Wissensbasiertes System - Ein wissensbasiertes System wird verwendet, eine Wissensrepräsentation wird durchsucht
- IRS ... Information-Retrieval-System - Ein Information-Retrieval-System wird verwendet, eine Dokumentenrepräsentation wird durchsucht
- DR ... Data Retrieval - Alle Ergebnisse zu einer Suche sind relevant und gleichermaßen relevant

- IR ... Information Retrieval - Eine nach Relevanz zur Anfrage gereihte Liste von Ergebnissen wird auf eine Suche retourniert
- SA ... Semantische Ähnlichkeit - Semantische Ähnlichkeit zwischen Konzepten wird für die Suche genutzt
- IA ... Inhaltsbasierte Ähnlichkeit - Inhaltsbasierte Ähnlichkeit zwischen Ressourcen wird für die Suche genutzt
- WR ... Wissensrepräsentation - Typ (Format) der verwendeten Wissensrepräsentation

3.5 Zusammenfassung

Dieses Kapitel befasste sich mit Information Retrieval im Kontext des Semantic Web. Basierend auf existierender Literatur wurde der potentielle Nutzen von Semantic-Web-Technologien für Information Retrieval beschrieben. Aktuelle Modelle und Systeme für die Suche im Semantic Web und am Semantic Desktop wurden untersucht und gegenübergestellt, und eine Definition für Information Retrieval im Semantic Web wurde erarbeitet.

	W	D	KBA	TBA	WBS	IRS	DR	IR	SA	IA	WR
(Bar-Yossef u. a., 1999)	X		X		X		X				OHTML
(Heflin u. Hendler, 2000)	X		X		X		X				SHOE
(Stojanovic u. a., 2001)	(X)[19]		X		X			X	(X)[20]		F-Logic
(Davies u. a., 2002)		X	(X)[21]	X	X	X		X		(X)[22]	RDF & RDFS
(Shah u. a., 2002)		X	X[23]	X	[24]	X		X		(X)[25]	DAML+OIL
(Sheth u. a., 2002)		X	X	[26]	X		X				unbekannt
(Bonino u. a., 2003)		X	X			X		X	X[27]		RDF
(Guha u. a., 2003)	X		(X)[28]		X		X				RDF
(Soo u. a., 2003)		X[29]		X	X			X	(X)[30]		RDF
(Stojanovic u. a., 2003)	(X)[19]		X		X			X	(X)[20]		F-Logic

[19] Suche in einem semantischen Portal
[20] Ähnlichkeit zwischen zwei Wissensbasen
[21] Filtern der Ergebnisse nach Typ und Suche in Eigenschaftswerten
[22] Ähnlichkeit zwischen textueller Anfrage und textuellem Inhalt von Dokumenten und Eigenschaftswerten
[23] RDF-Tripel mit Platzhaltern
[24] Inferenz in Wissensbasis um die Menge an Tripel zu erweitern, welche gemeinsam mit Text indiziert wurden
[25] Ähnlichkeit zwischen textueller Anfrage und textuellem Inhalt von Dokumenten
[26] Nicht für Volltextsuche entworfen, aber möglich
[27] Ähnlichkeit zwischen Anfrage und semantischen Annotationen, Super- und Subklassen zur Erweiterung der Anfrage
[28] Knoten in einem RDF-Graph
[29] Textuelle Metadaten zu Bildern
[30] Ähnlichkeit zwischen konzeptueller Anfrage und semantischen Annotationen

	W	D	KBA	TBA	WBS	IRS	DR	IR	SA	IA	WR
(Bamba u. Mukherjea, 2004)		X	X		X			X			RDF & RDFS
(Kiryakov u. a., 2004)		X	X	X	X	X	X[31]	X		(X)[22]	RDFS
(Priebe u. a., 2004)	(X)[32]		X		X			X	(X)[33]		RDF
(Rocha u. a., 2004a)	(X)[34]			X	X	X		X	X[35]		unbekannt[36]
(Bangyong u. a., 2005)	[37]			X	X	X		X			unbekannt
(Song u. a., 2005)	[37]		X			X		X	(X)[30]		none
(Zhang u. a., 2005)	(X)[19]		X	(X)[38]	X	X		X			fuzzy DL $\mathcal{ALC}$
(Corby u. a., 2006)		X	X		X			X	X[33]		RDF, RDFS, OWL Lite
(Choudhury u. Phon-Amnuaisuk, 2006)		X		X	X	X		X	X[35]		unbekannt
(Jiang u. Tan, 2006)		X		X	X	X		X	X[39]	(X)[25]	Taxonomie[40]
(Chirita u. a., 2006)		X		X		X		X		(X)[22]	RDF

[31] Entitäten aus einer Wissensbasis können gesucht werden
[32] Suche in einem Wissensportal
[33] Ähnlichkeit zwischen Anfrage und Entitäten in einer Wissensbasis
[34] Suche auf eine Webseite
[35] Ähnlichkeit basierend auf der Struktur eines Graphen, welcher aus der Ontologie erzeugt wurde, zur Erweiterung der Anfrage
[36] ,,can be easily mapped to RDF"
[37] Ein Modell wird vorgestellt, kein System
[38] Term-basierte Anfragen werden als Instanzen von Konzepten in einer Wissensbasis modelliert
[39] Konzepte sind ähnlicher je öfter jene Relationen, welche Konzepte miteinander verbinden in der Wissensbasis vorkommen
[40] ACM Computing Classification System

	W	D	KBA	TBA	WBS	IRS	DR	IR	SA	IA	WR
(Castells u. a., 2007)		X	X		X	X		X	$(X)^{30}$	$(X)^{25}$	OWL

Tabelle 3.1: Klassifikation der vorgestellten Systeme zur Suche im Semantic Web

4 Assoziatives Retrieval, Assoziative Netze und Spreading Activation

4.1 Einleitung

In diesem Kapitel wird das in dieser Arbeit angewandte Suchparadigma des Assoziativen Retrieval vorgestellt. Ebenfalls erfolgt die Einführung in das Themengebiet der Assoziativen Netze, die für diese Form der Suche herangezogen werden können. In Folge wird mit Spreading Activation jener Verarbeitungsansatz von Netzen betrachtet, welcher in dieser Arbeit Verwendung findet. Im Anschluss an diese Einführung der grundlegenden Konzepte erfolgt ein Überblick über Systeme, welche auf Assoziativem Retrieval, Assoziativen Netzen oder Spreading Activation aufbauen. Diese Systeme werden anhand ihrer Charakteristika einander gegenübergestellt.

4.2 Assoziatives Retrieval

Die Idee des Assoziativen Retrievals reicht in die 1960er Jahre zurück als Forscher im Information Retrieval versuchten ihre Systeme dadurch zu verbessern, dass Assoziation zwischen Dokumenten oder Indextermen, welche im Vorfeld ermittelt wurden, für die Suche mit in Betracht gezogen werden.

Salton (1963) stellt einen Ansatz vor, in dem bibliographische Daten in den Suchprozess in einem Dokument-Retrieval-System integriert werden. Dokumente, welche vom selben Autor stammen, werden miteinander assozi-

iert. Zusätzlich werden auch Dokumente, welche von anderen Dokumenten referenziert werden mit dem referenzierenden Dokument assoziiert.

Einige Jahre später beschreibt Salton (1968) eine Methode zum *Linear Associative Retrieval* als Erweiterung zum Vektorraummodell. Er zeigt wie Term-zu-Term-Assoziationen und Dokument-zu-Dokument-Assoziationen, welche im Vorfeld durch die statistische Analyse von Dokumenten und den darin vorkommenden Wörtern ermittelt wurden, für die Suche genutzt werden können.

Crestani (1997a) formuliert eine generischere Sicht auf *Assoziatives Retrieval* (engl. *Associative Retrieval*). So sieht er es als eine Variante des Information Retrieval (vgl. Kap 2.3) in der versucht wird, von einer Menge gegebener Information ausgehend, welche als relevant gilt, verwandte (assoziierte) Information zu finden, welche ebenfalls relevant ist.

> *The idea behind this form of information retrieval is that it is possible to retrieve relevant information by retrieving information that is "associated" with some information the user already retrieved and that is know it to be relevant. The associations between information can either be static and already existing at the time of the query session or dynamic and determined at run time. In the first case, associations among information items (document or parts of documents, extracted terms, index terms, concepts, etc.) are created before the query session, and they make use of semantic relationships between these items, such as for example thesaurus-like relationships among index terms, bibliographic citations among documents, or statistical similarity among documents or terms. In the last case, instead, the system determines associations between information items through interaction with the user, for example by retrieving documents that are similar to documents the user points out to be relevant ...*
>
> Crestani (1997a)

Es werden also sich im Retrieval-System befindende Informationsobjekte im Vorfeld oder während der Laufzeit miteinander in Verbindung gesetzt. Diese Assoziationen sollen dazu dienen relevante Information zu finden. Bei den assoziierten Informationsobjekten kann es sich um (Teile von) Dokumente(n), Index-Terme, aber auch um Konzepte handeln.

4.3 Assoziative Netze

Wie in Abschnitt 4.2 beschrieben, werden beim Assoziativen Retrieval sich im Retrieval-System befindende Informationsobjekte miteinander in Verbindung gesetzt. Oft werden diese Verbindungen als Netz bzw. Graph repräsentiert, in welchem Informationsobjekte als Knoten und Assoziationen als Kanten zwischen den Knoten dargestellt werden. Dieses Assoziative Netz wird dann für den Vorgang der Suche genutzt.

Crestani (1997a) und Berger u. a. (2003) sehen ein *Assoziatives Netz* (engl. *Associative Network*) als generisches Netz von Informationsobjekten, welche durch Knoten in einem Graphen dargestellt werden. Die Kanten zwischen den Knoten können benannt und/oder gewichtet werden, wobei das Gewicht für die Stärke der Assoziation zwischen den Informationsobjekten steht:

> *This is a generic network of information items in which information items are represented by nodes, and links express sometimes undefined and unlabeled associative relations among information items. In modern IR, where statistical techniques are used in the indexing phase to associate weights to index terms, the relationships among information items are sometimes weighted, so adding to the network a measure of the strength of associations.*
>
> Crestani (1997a)

Heylighen u. Bollen (2002) geben eine engere Definition eines Assoziativen Netzes. Sie sehen es als gewichteten, gerichteten Graphen in dem Knoten für Dokumente stehen und die Gewichte der Kanten für die Stärke der Assoziation zwischen den Dokumenten stehen:

> *An associative network is a weighted, directed graph, whose nodes d_i represent documents, and whose weights represent the association between nodes. It can be represented as a matrix whose components $a_{ij} \in [0, 1]$ correspond to the connection weights between nodes d_i and d_j.*
>
> Heylighen u. Bollen (2002)

Heylighen u. Bollen (2002) repräsentieren das Assoziative Netz als quadratische Matrix, welche es erlaubt jeden Koten mit jedem andere Knoten zu verbinden. Die Werte in der Matrix korrespondieren mit der Stärke der Assoziation zwischen je zwei Knoten. Im Gegensatz zur Definition von Crestani (1997a) erlauben sie keine benannten Kanten.

In dieser Arbeit wird ein Assoziatives Netz wie in (Crestani, 1997a) verstanden. Die Assoziationen zwischen den Knoten des Netzes werden nicht wie in (Heylighen u. Bollen, 2002) in der Form einer Matrix gespeichert, da diese Adjazenzmatrix nur dünn besetzt wäre. Auf die Datenstruktur zur Verwaltung des Netzes wird in den Abschnitten 7.6.1 und 7.7.1 eingegangen.

4.3.1 Ermittlung der Assoziationen

Wie Crestani (1997a) beschreibt dienen im Information Retrieval statistische Maße wie tf*idf (vgl. Abschnitt 5.2.2) dazu die Kantengewichte zwischen Term- und Dokumentknoten in Assoziativen Netzen zu ermitteln, welche eine Dokumentensammlung repräsentieren. Dies geschieht während der Indexierung der Dokumente.

Auch benannte Kanten zwischen Konzepten können Teil eines Assoziativen Netzes sein. Diese stammen aus einer Wissensrepräsentation[1], wie es zum Beispiel in (Berger u. a., 2003) der Fall ist.

Heylighen (2001) beschreibt die experimentelle Erstellung von Assoziativen Netzen in der Psychologie: Eine Versuchsperson wird mit einem Wort (z.B. Katze) konfrontiert und gefragt, welche anderen Wörter ihr einfallen (z.B. Hund, Maus, Milch). Je öfter ein Wort in der Antwortmenge zu einem andern Wort vorkommt, je stärker wird der Grad der Assoziation zwischen den beiden Wörtern.

4.3.2 Assoziative und Semantische Netze

Semantische Netze haben als Werkzeug zur symbolischen Wissensrepräsentation eine lange Geschichte in der Künstlichen Intelligenz. Der

[1] Hier sei auf die Definition zu Information Retrieval der FGIR in Kapitel 2.3 hingewiesen, die Wissensrepräsentationen als mögliche Bestandteile eines Information-Retrieval-Systems sieht.

Grundgedanke dieser Form der Wissensrepräsentation ist es Konzepte als Knoten in einem Netz zu modellieren und diese mit symbolischen Kanten miteinander zu verbinden. Sowohl Knoten als auch Kanten im Netz sind also benannt.

Gerne (so auch in (Sharples u. a., 1989)) wird auf (Quillian, 1968) als den Ursprung der Semantischen Netze hingewiesen, da dort Netze als ein Modell für die Repräsentation semantischer Information im menschlichen Gedächtnis verwendet werden, um die Verarbeitung von Sprache im menschlichen Gehirn zu simulieren.

Wie Sharples u. a. (1989) ausführen ist die Idee eines Semantischen Netzes, als Netz von assoziativ verbundenen Konzepten, allerdings viel älter. So verweisen sie auf Anderson u. Bower (1973), welche die Wurzeln des Assoziationismus auf Aristoteles zurückführen. Hierbei verweisen sie auf sein Essay *Über Gedächtnis und Erinnerung* (*On Memory and Reminiscence*). Auch Anderson (1998) weißt darauf hin, dass der Grundstein für assoziative Berechnungen mit Computern im 4 Jahrhundert vor Christus mit den Werken von Aristoteles gelegt wurde. Nach dessen Theorien besteht das menschliche Gedächtnis aus einer Menge von elementaren Einheiten, welche miteinander verbunden sind.

Oft werden die Begriffe Semantisches Netz und Assoziatives Netz synonym gebraucht (Crestani, 1997a). Hier sei darauf hingewiesen, dass der Fokus in Semantischen Netzen darauf liegt den Verbindungen zwischen zwei Begriffen eine Bedeutung zu geben. Bei Assoziativen Netzen liegt der Fokus darauf Elemente des Netzes in Beziehung zueinander zu setzen, sei es aufgrund von semantischer Ähnlichkeit zwischen Begriffen oder basierend auf einem statistischen Ähnlichkeitsmaß für Texte. Zusätzlich zur Benennung der Kanten im Netz erlauben es Assoziative Netze die Stärke der Assoziation zwischen zwei Elementen anhand eines Kantengewichts zu modellieren. Semantische Netze können also als Teilmenge von Assoziativen Netzen gesehen werden.

Crestani u. van Rijsbergen (1993, 1997) und Berger u. a. (2004b) stellen fest, dass Kanten in Assoziativen Netzen nicht typisiert sein müssen (allerdings typisiert sein können), da ihr Fokus nicht in der Beschreibung der Semantik der Assoziation liegt, sondern viel mehr auf der Beschreibung der Stärke der Assoziation. Rocha u. a. (2004a), O'Riordan u. Sorensen (1995)

und Montaner u. a. (2003) sehen Kanten in Assoziativen Netzen generell als nicht typisiert an, da die Kanten im Assoziativen Netz ausschließlich über die Stärke der Assoziation Auskunft geben.

4.4 Spreading Activation

Spreading Activation (zu Deutsch *Aktivierungsausbreitung*) beschreibt eine Familie von Algorithmen, welche ihren Ursprung in der Kognitivpsychologie (vgl. (Anderson u. Pirolli, 1984)) haben. Dort bieten sie einen Erklärungsmechanismus wie Wissen im menschlichen Gehirn repräsentiert und verarbeitet wird. Vereinfacht ausgedrückt wird das Gedächtnis als ein Netz aus Knoten, welche für Begriffe stehen, gesehen. Verbindungen zwischen Begriffen werden als Kanten zwischen diesen Knoten modelliert. Wird ein Teil des Netzes aktiviert, breitet sich die Aktivierung in angrenzende Teile des Netzes aus. Die Geschwindigkeit des Zugriffs auf einen Begriff und die Wahrscheinlichkeit, dass auf ein Konzept zugegriffen wird, hängen von der Höhe der Aktivierung ab, welche sich durch das Netz ausbreitet. Diese wiederum basiert darauf wie häufig dieser Teil des Gedächtnisses zuletzt genutzt wurde (Sharifian u. Samani, 1997).

4.4.1 Ausbreitung der Aktivierung

Die Aktivierungsausbreitung eines Knotens findet laut Crestani (1997a) in vier Schritten statt, welche zyklisch ablaufen. Vor und nach dem eigentlichen Schritt der *Weitergabe von Aktivierung* können optional ein Schritt der *Vorbereitung* bzw. der *Nachbereitung* stehen. In diesen Schritten kann zusätzliche Funktionalität in den Algorithmus integriert werden. So kann im Schritt der Vorbereitung die Überprüfung erfolgen ob eine Kante von einem bestimmten Typ ist und somit traversiert werden soll. Im Schritt der Nachbereitung kann ein Teil der Aktivierung, die der Knoten derzeit aufweist verfallen, um ein sich Aufschaukeln der Aktivierungen der Knoten im Netz zu vermeiden. Am Ende eines Zyklus steht die *Prüfung einer Abbruchbedingung*. Ist diese erfüllt stoppt die Aktivierungsausbreitung. Abbildung 4.1 zeigt die vier Schritte der Aktivierungsausbreitung.

Crestani (1997a) definiert die Berechnung der Aktivierung eines Knotens wie in Gleichung 4.1 beschrieben. Als typische Beispiele für die Funktion f

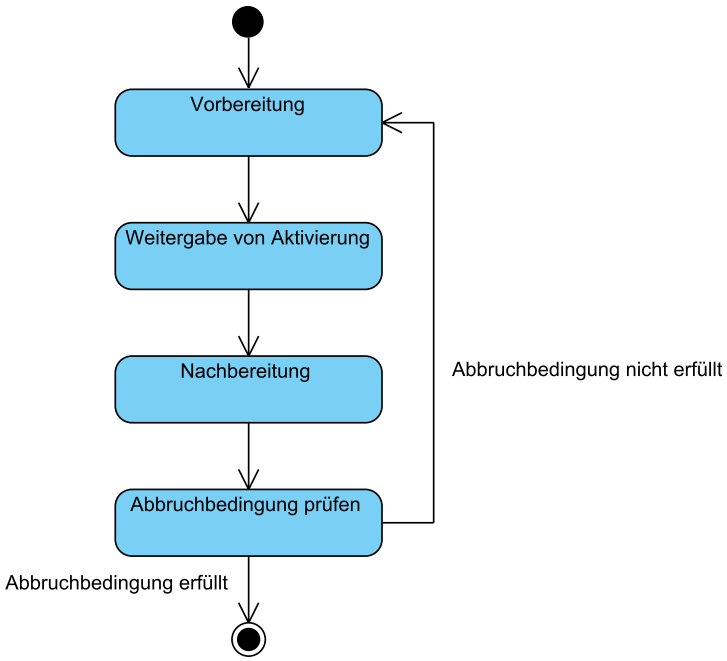

Abbildung 4.1: Die Schritte von Spreading Activation (nach Crestani (1997a))

führt er lineare, sigmoide und Schwellwertfunktionen an. Die Gleichungen 4.2, 4.3 und 4.4 zeigen exemplarisch diese drei Aktivierungsfunktion. Die Abbildungen 4.2, 4.3 und 4.4 geben jeweils ein Beispiel für den Verlauf dieser Aktivierungsfunktion für eine Eingangsaktivierung im Wertebereich $[0, 1]$.

$$A_j = f(I_j) \tag{4.1}$$

Wobei gilt:

- A_j ... Aktivierung von Knoten j
- I_j ... Eingangsaktivierung von Knoten j
- f ... Aktivierungsfunktion

$$A_j = a \cdot I_j + b \tag{4.2}$$

$$A_j = \begin{cases} 0: & I_j < t_j \\ 1: & I_j \geq t_j \end{cases} \tag{4.3}$$

$$A_j = \frac{1}{1 + e^{-I_j}} \tag{4.4}$$

Wobei gilt:

- A_j ... Aktivierung von Knoten j
- I_j ... Eingangsaktivierung von Knoten j
- a ... Konstante
- b ... Konstante
- t_j ... Schwellwert von Knoten j

Crestani (1997a) definiert die Berechnung der Eingangsaktivierung eines Knotens bei Spreading Activation in seiner einfachsten Form wie in Gleichung 4.5. Die Eingangsaktivierung des Knotens j ist abhängig von seinen Nachbarn i die ihm Aktivierung zuführen und den Gewichten der Kanten durch welche der Knoten j mit seinen Nachbarn verbunden sind.

$$I_j = \sum_i O_i \cdot w_{ij} \tag{4.5}$$

Wobei gilt:

- I_j ... Eingangsaktivierung von Knoten j
- O_i ... Ausgangsaktivierung von Knoten i
- $w_{i,j}$ ... Gewicht der Kante zwischen Knoten i und Knoten j

Laut Crestani (1997a) besteht für gewöhnlich kein Unterschied zwischen der Aktivierung eines Knotens und der Aktivierung die er an seine Nachbarknoten weitergibt (Ausgangsaktivierung des Knotens). Dies ist in Gleichung 4.6 beschrieben.

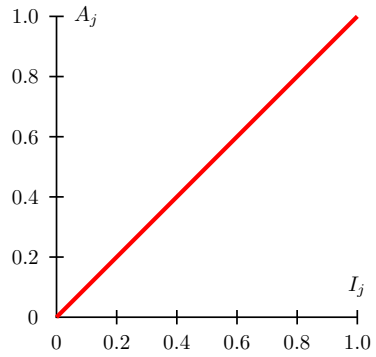

Abbildung 4.2: Eine lineare Aktivierungsfunktion

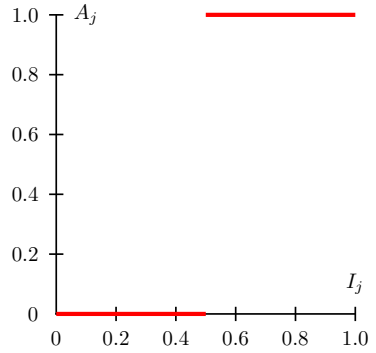

Abbildung 4.3: Eine Schwellwert-Aktivierungsfunktion

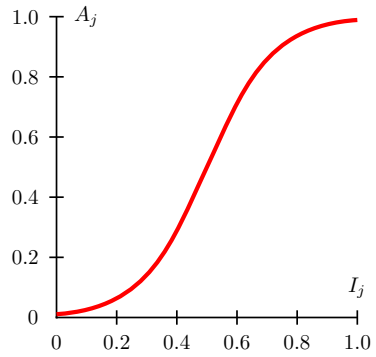

Abbildung 4.4: Eine Sigmoid-Aktivierungsfunktion

$$O_j = A_j \tag{4.6}$$

Wobei gilt:

- O_j ... Ausgangsaktivierung von Knoten j
- A_j ... Aktivierung von Knoten j

Der Spreading-Activation-Algorithmus wirkt in seiner grundlegenden Beschreibung einfach, kann aber in der Praxis zu komplexen Problemstellungen führen. So muss ein Mittelmaß zwischen einer sich ausdünnenden Aktivierung, wie es bei Graphen mit sehr vielen Kanten der Fall ist, und einer Überaktivierung, welche bei zyklischen Graphen auftritt, gefunden werden. Dies kann durch verschiedene *Constraints* (vgl. Abschnitt 4.4.2) realisiert werden.

Beim in Gleichung 4.5 vorgestellten Spreading-Activation-Ansatz handelt es sich um ein rein numerisches Verfahren, welches den Aktivierungswert eines Knoten in Abhängigkeit von seinen Nachbarn berechnet. Semantische Relationen zwischen Konzepten können in dieser einfachen Form, nur dahingehend berücksichtigt werden, dass abhängig vom Typ der jeweiligen Relation ein anderes Gewicht zwischen den Konzepten verwendet wird. Somit geht der angeführte Ansatz nicht auf die Möglichkeiten in Semantischen Netzen ein (vgl. Abschnitt 4.4.5).

4.4.2 Constrained Spreading Activation

Die Aktivierungsausbreitung kann durch mehrere Maßnahmen eingeschränkt bzw. kontrolliert werden. Dies wird als *Constrained Spreading Activation* bezeichnet. Der Begriff Constrained Spreading Activation geht auf Cohen u. Kjeldsen (1987) und das von ihnen vorgestellt System GRANT (vgl. Abschnitt 4.5) zurück. Zusätzlich zu den drei Constraints aus GRANT, *Distance Constraint*, *Fan-out Constraint* und *Path Constraint*, führt Crestani (1997a) noch das *Activation Constraint* an. Im Folgenden werden diese vier Constraints als Beispiele für mögliche Einschränkungen angeführt.

Distance Constraint: Die Aktivierungsausbreitung eines Knoten stoppt, wenn sie eine gewisse Entfernung vom ausgehenden Knoten erreicht hat. Beispielsweise breitet sich die Aktivierung von Knoten A über den benach-

barten Knoten B zu dessen Nachbarknoten C aus. Anstatt sich von Knoten C auch zu dessen Nachbarn auszubreiten, stoppt die Ausbreitung hier. Dieser Constraint ist mit der semantischen Distanz von Konzepten in Thesauri vergleichbar, welche auf der Anzahl der Knoten und Kanten zwischen zwei Knoten basiert.

Fan-out Constraint: Knoten welche eine hohe Anzahl an ausgehenden Kanten haben, geben keine Aktivierung ab. Somit wird eine Ausbreitung der Aktivierung zu weit im Netz verhindert. In einem symbolischen Netz kann dieses Constraint für Knoten mit einer Kantenanzahl, welche einen bestimmten Wert überschreitet aktiv werden. Eine Variante dieses Constraints in Netzen mit reeller Aktivierungsausbreitung kann durch die Verwendung von Aktivierungsfunktionen, welche nach dem Prinzip der Aktivierungserhaltung agieren, realisiert werden. Das heißt ein Knoten kann nur soviel Aktivierung abgeben wie er aufnimmt. Alternativ dazu kann die ausgehende Aktivierung auch von der Anzahl an ausgehenden Kanten abhängig gemacht, beispielsweise durch diese dividiert werden.

Path Constraint: Aktivierung darf sich über gewisse Kanten ausbreiten über andere nicht. Das System GRANT basiert hier auf *Path Endorsement*, das heißt manche Pfade werden bevorzugt für die Aktivierungsausbreitung genutzt. Path Constraints können einerseits über hohe bzw. niedrige Gewichtung der Kanten im Vorfeld realisiert werden, andererseits, bei typisierten Kanten, über einen Spreading-Activation-Algorithmus, welcher den Typ der Kante mit in Betracht zieht.

Activation Constraint: Für jeden Knoten oder für jede Gruppe an Knoten kann eine individuelle Schwellwertfunktion festgelegt werden. Nur wenn der Schwellwert der Aktivierung eines Knotens überschritten wird, gibt der Knoten Aktivierung an seine Nahbaren ab. Legt man dieses Constraint auf ein symbolisches Netz um, könnten in die Knoten Regeln implementiert werden, welche die Informationen, die die Knoten von ihren Nachbarn (über Marker, vgl. Abschnitt 4.4.4) erhalten auswerten.

4.4.3 Spreading-Activation-Algorithmen

Wie die Systembeschreibungen in Abschnitt 4.5 zeigen, gibt es alleine im Information Retrieval viele Anwendungsfälle des Spreading-Activation-

Verfahrens. Allerdings werden nicht immer alle Details zur verwendeten Implementierung angeführt. Auch existieren kaum gegenüberstellende Vergleiche von verschiedenen Ansätzen.

Alani u. a. (2002), Alani u. a. (2003) und Rocha u. a. (2004a) führen Pseudocode zum verwendeten Algorithmus an, welcher auf einfachem Niveau gehalten ist. Eine umfangreiche Sammlung an aktuellen Spreading-Activation-Methodiken stellt (Alves u. Jorge, 2005) dar.

Einige der wenigen Evaluierungen unterschiedlicher Spreading-Activation-Methodiken sind in (Chen u. Ng, 1995) (1) und (Huang u. a., 2004a) (2) zu finden:

(1) Chen u. Ng (1995) vergleichen einen *Branch-and-Bound-Algorithmus* mit einem *Hopfield-Net-Algorithmus* für die Suche in Thesauri. Dabei greifen sie auf den in (Chen u. Dhar, 1991) vorgestellten Branch-and-Bound-Algorithmus zurück. Dieser schätzt für jeden zu aktivierenden Pfad die Kosten dieser Operation und aktiviert dann jenen Pfad der am wenigsten Kosten verursacht. In (Chen u. Dhar, 1991) basiert die Schätzfunktion darauf die relevantesten Terme in einem Thesaurus zu finden, hierzu werden die Knoten in Abhängigkeit ihrer Aktivierung gereiht. Die Aktivierungsausbreitung endet, wenn eine vom Benutzer definierte Menge an Suchergebnissen gefunden wurde. Der Hopfield-Net-Algorithmus stammt aus (Chen u. a., 1993), er beruht auf einer sigmoiden Aktivierungsfunktion. Die Aktivierungsausbreitung wird so lange fortgesetzt bis keine Änderung mehr in den Aktivierungswerten der Knoten stattfindet. Chen u. Ng (1995) kommen zum Schluss, dass währen der Branch-and-Bound-Algorithmus schneller ist, der Hopfield-Net-Algorithmus bessere Ergebnisse liefert. Chen u. Ng (1995) führen dies auf die parallele Aktivierungsausbreitung des Hopfield-Net-Algorithmus zurück, während sich beim Branch-and-Bound-Algorithmus die Aktivierung ungleichmäßig in Richtung der höchstgereihten Knoten ausbreitet.

(2) Huang u. a. (2004a) vergleichen ebenfalls den Branch-and-Bound-Algorithmus aus (Chen u. Dhar, 1991) mit dem Hopfield-Net-Algorithmus aus (Chen u. a., 1993) im Kontext von Recommender-Systemen. Zusätzlich nehmen sie noch einen Algorithmus auf, den sie als *Leaky-Capacitor-Modell* bezeichnen und für dessen Herkunft sie auf (Anderson, 1983) verweisen[2].

[2]Der Begriff *leaky capacitor model* kommt weder in (Anderson, 1983) noch in (Anderson u.

Sie modellieren die Datenbasis eines Recommender-Systems als Assoziatives Netz (vgl. Abschnitt 4.5) und stellen fest, dass alle drei Algorithmen bei dünnen Graphen bessere Ergebnisse liefern als Empfehlungsansätze auf Basis des Vektorraummodells. Dabei liefert der auf dem Leaky-Capacitor-Modell basierende Algorithmus die besten Ergebnisse. Bei dichten Graphen verschlechtert sich die Leistung der Spreading-Activation-Algorithmen. Alle drei getesteten Spreading-Activation-Algorithmen führen mehrere Iterationen durch, wodurch es zu Überaktivierung im Graphen kommen kann. Auf diese Überaktivierung führen Huang u. a. (2004a) negative Ergebnisse in der Evaluierung zurück. Der Algorithmus auf Basis des Leaky-Capacitor-Modells war am empfindlichsten auf Überaktivierung und zeigte die stärkste Verschlechterung.

4.4.4 Marker Passing

Marker Passing ist eine Variante von Spreading Activation. Bei Markern handelt es sich um symbolische Objekte, welche von unterschiedlichem Typ sein können. Knoten können somit aufgrund des Markers den sie empfangen unterschiedliche Reaktionen setzen. Des Weiteren können Marker auch Information beinhalten, wie beispielsweise der Knoten von dem sie ausgingen oder die Kanten über die sie zum aktuellen Knoten gekommen sind (Barnden u. a., 2002). Marker Passing lässt somit zusätzliche Möglichkeiten zur Kontrolle der Aktivierungsausbreitung im Netz zu.

4.4.5 Spreading Activation in Semantischen und Neuronalen Netzen

Barnden u. a. (2002) geben einen Überblick über die Verarbeitung von Semantischen Netzen mit Spreading Activation und Marker Passing und vergleichen in diesem Zusammenhang die Verarbeitung von Neuronalen Netzen mit Hilfe von Spreading Activation. Sie stellen fest, dass während sich in Semantischen Netzen Aktivierung binär ausbreitet, also ein Knoten entweder aktiviert oder nicht aktiviert wird, sich in Neuronalen Netzen Aktivierung in der Regel kontinuierlich ausbreitet. Auch Crestani (1997a) stellt fest, dass in

Pirolli, 1984) vor. Er wird allerdings von Pirolli u. a. (1996) verwendet.

Semantischen Netzen häufig eine binäre Aktivierungsfunktion genutzt wird. Die Aktivierung eines Knotens kann somit nur 0 oder 1 bzw. -1 oder 1 annehmen. Zusätzlich führen Barnden u. a. (2002) als eine mögliche Variante von Spreading Activation in Semantischen Netzen an, nur Kanten eines speziellen Link-Typs zu folgen. Dies wird beispielsweise durch das *Path Constraint* wie im System GRANT (vgl. 4.5) realisiert.

Auch Rose u. Belew (1989) stellen Spreading Activation in konnektionistischen Ansätzen und Semantischen Netzen gegenüber und stellen fest, dass es sich dabei aus oben genannten Gründen um unterschiedliche Ansätze handelt. Chen u. a. (1993) kommen zum selben Schluss, werfen aber ein, dass durch die Entwicklung von komplexeren, hybriden Systemen diese Grenzen verschwimmen.

Sowohl ACTE (Anderson, 1976) als auch ACT* (Anderson u. Pirolli, 1984) sind Theorien über die Verarbeitung von Gedanken im menschlichen Gedächtnis. Beide greifen auf so genanntes deklaratives Wissen zurück, dieses kann mit Semantischen Netzen verglichen werden. Ein interessant Punkt in diesem Zusammenhang ist, dass das ACT*-Modell kontinuierliche Aktivierungswerte für die Konzepte im Gedächtnis annimmt, während das ACTE-Modell noch auf binärer Aktivierung basiert (Anderson u. Pirolli, 1984).

4.5 Assoziative-Retrieval-Systeme

Im Folgenden werden Systeme beschrieben, welche von ihren Autoren als Assoziatives-Retrieval-System bezeichnet werden und / oder deren Funktionalität auf Assoziativen Netzen beruht. Die anführten Suchansätze wurden im Rahmen einer Literaturrecherche identifiziert, die mit dem Ziel durchgeführt wurde einen großen Teil der aktuell existierenden Systeme und deren Methoden zu beschreiben. Im Gegensatz zu der in Abschnitt 3.4 durchgeführten explorativen Studie dient die hier beschriebene Untersuchung nicht zum Auffinden von neuen Richtungen für eigene Forschung, sondern ausschließlich dazu vorhandene Methoden zu strukturieren und die Konstruktvalidität des durchgeführten Experiments (vgl. Abschnitt 9.3.1) zu erhöhen.

(Salton, 1963): Im von Salton (1963) vorgestellten Ansatz werden bibliographische Daten in den Suchprozess in einem Dokument-Retrieval-System auf Basis des Vektorraummodells integriert. Dokumente, welche vom selben Autor verfasst wurden, werden miteinander assoziiert. Dokumente welche von anderen Dokumenten referenziert werden, werden mit dem referenzierenden Dokument assoziiert. Die Ergebnismenge auf eine Suchanfrage wird mit den Dokumenten, welche mit den Dokumenten in der Ergebnismenge assoziiert sind erweitert.

Linear Associative Retrieval (Salton, 1968): Salton (1968) beschreibt *Linear Associative Retrieval* als Erweiterung zum Vektorraummodell. Er zeigt wie Term zu Term und Dokument zu Dokument Assoziationen, welche im Vorfeld, statistisch ermittelt wurden in den Retrieval-Prozess integriert werden können. In vorangegangenen Ansätzen stammten Assoziationen zwischen Indextermen aus einem Thesaurus und Assoziationen zwischen Dokumenten wurden aus bibliographische Daten ermittelt (vgl. (Salton, 1963)). Im Linear Associative Retrieval werden diese Assoziationen aus der Dokument-Term-Matrix gewonnen.

Dokumente, welche dieselben Terme beinhalten und Terme, welche in denselben Dokumenten vorkommen, werden miteinander assoziiert. Term zu Term Assoziationen werden dazu genutzt um die Menge an Anfragetermen zu erweitern. Ausgehend von den Anfragetermen des Benutzers werden assoziierte Terme in die Anfrage eingebracht. Diese erweiterte Anfrage wird nun an das Retrieval-System gesendet. Die Dokument zu Dokument Assoziation werden verwendet um die Ergebnismenge zu erweitern. Nachdem eine Ergebnismenge auf eine Anfrage an das System zurückgeliefert wurde, wird diese Ergebnismenge um Dokumente erweitert, welche mit den Dokumenten in der ursprünglichen Ergebnismenge assoziiert sind.

SAM (Preece, 1981): Laut Crestani (1997a) handelt es sich bei der Arbeit von Preece (1981) um einen der ersten Ansätze assoziativer Suche mittels Spreading Activation im Information Retrieval. Preece (1981) stellt ein *Spreading Activation Network Model for Information Retrieval* (SAM) vor. Dieses Modell basiert laut Preece (1981) auf der Beobachtung, dass Datenbanken (und Dokumentensammlungen) als Netze von Knoten, welche Dokumente repräsentieren, modelliert werden können. Eine natürliche Art

Netze zu durchsuchen ist das Traversieren von Kanten. Dies wird im vorgestellten Ansatz durch einen Spreading-Activation-Algorithmus realisiert. Preece (1981) bezeichnet die Arbeit als Assoziative-Retrieval-Methode, welche es ermöglicht Dokumente zu finden, welche für ein Thema relevant sind, allerdings nicht diesem Thema zugeordnet wurden.

GRANT (Cohen u. Kjeldsen, 1987): Das Expertensystem *GRANT* (Cohen u. Kjeldsen, 1987; Kjeldsen u. Cohen, 1987) ermöglicht die Suche nach Förderagenturen, welche potentiell ein bestimmtes Forschungsvorhaben unterstützen. Zu diesem Zweck werden Förderagenturen und Forschungsthemen als Semantisches Netz modelliert, welches mittels Spreading Activation durchsucht wird.

(Cohen u. Kjeldsen, 1987) prägen mit GRANT den Ansatz der *Constrained Spreading Activation* (siehe Abschnitt 4.4.2). Bei diesem Spreading-Activation-Algorithmus werden auf Grund einer Menge an zuvor definierten Regeln manche Pfade im Netz bevorzugt, wodurch eine bessere Kontrolle der Aktivierungsausbreitung ermöglicht wird.

AIR (Belew, 1987): Belew (1987, 1987, 1989) stellt das Assoziative-Retrieval-System *AIR* (*Adaptive Information Retrieval*) vor. AIR ist einer der ersten konnektionistischen Ansätze für text-basiertes Information Retrieval. Aus einer Menge an Dokumenten, deren Autoren und den in ihnen vorkommenden Wörtern wird ein Assoziatives Netz aufgebaut, in dem mittels Spreading Activation gesucht wird. Neben Ähnlichkeiten in den Wortmengen der Textdokumente im System, verwendet AIR zusätzlich den Autor der Dokumente um Assoziationen zwischen Dokumenten festlegen zu können.

Zusätzlich ermöglicht AIR auf Basis von Rückmeldungen der Benutzer über die Qualität der Suchergebnisse die dem System zugrunde liegende Netzstruktur unter Verwendung einer Variante des Hebbian Learning (vgl. (Medler, 1998)) zu modifizieren.

I^{3}R (Croft u. a., 1988): *I^{3}R* (*Intelligent Intermediary for Information Retrieval*) (Croft, 1986; Croft u. a., 1988; Croft u. Thompson, 1987) versucht ein text-basiertes Information-Retrieval-System mit einer Wissensbasis (wie in GRANT, vgl. Abschnitt 4.5) zu kombinieren. Zu diesem Zweck baut I^{3}R ein Netz auf, welches Knoten für Konzepte, Dokumente und Terme beinhaltet und mittels Spreading Activation durchsucht wird. I^{3}R folgt

dem Konzept der *multiple sources of evidence*. Der Suchende erhält mehrere Möglichkeiten Indizien für ein Dokument zu spezifizierten. Ein Dokument ist umso relevanter, je mehr Indizieren es für seine Relevanz gibt.

Die Suche in I^3R findet iterativ statt, der Benutzer kann jene Terme, Dokumente und Konzepte spezifizieren, welche für ihn relevant sind. Anhand dieser Knoten im Netz breitet sich Aktivierung aus. Der Benutzer bekommt in weiterer Folge ein Suchergebnis aus Dokumenten und Konzepten angezeigt, aus dem er erneut ein Auswahl treffen kann.

Im in (Croft u. a., 1988) evaluierten Ansatz wird die Wissensbasis allerdings außer Acht gelassen. Die Suche startet mit einem text-basierten, probabilistischen Suchansatz. Ausgehend von den so gefunden Dokumenten wird eine Suche im Netz durchgeführt, welche ähnliche Dokumente (anhand von Inhalt oder bibliographischen Daten) findet.

SCALIR (Rose u. Belew, 1989): *SCALIR* (*Symbolic and Connectionist Approach to Legal Information Retrieval*) (Rose u. Belew, 1989, 1991) ist ein Rechts-Information-Retrieval-System für Urheberrecht. Das System modelliert Fälle, Gesetze auf die in den Fällen verwiesen wird und die in den Entscheidungen der Fälle vorkommenden Wörter als Netz. SCALIR baut nach den Aussagen der Autoren auf AIR (vgl. Abschnitt 4.5) auf. Zusätzlich zu sub-symbolischen Kanten, welche Fälle mit denen in ihnen vorkommenden Wörtern verbinden, schlagen die Autoren vor auch symbolischen Kanten in das Netz aufzunehmen. Diese Kanten sind dazu gedacht Fälle oder Gesetze miteinander zu verbinden, um beispielsweise festzulegen, welches Urteil durch welchen Prozess aufgehoben wurde oder welches Gesetz auf ein anderes verweist. Zur Zeit der Erstellung des in (Rose u. Belew, 1989) beschriebenen Systems war ausschließlich Funktionalität für sub-symbolischen Kanten implementiert, in (Rose u. Belew, 1991) befanden sich die symbolischen Kanten im experimentellen Stadium.

(Wilkinson u. Hingston, 1991): Wilkinson u. Hingston (1991, 1992) stellen ein Dokument-Retrieval-System vor, welches auf einem Neuronalen-Netz-Modell basiert. Das Modell besteht aus drei Schichten, einer Schicht für Anfrageterme, einer Schicht für Dokumentterme und einer Schicht für Dokumente. Die Kanten zwischen den Knoten des Netzes werden so gewichtet, dass die Suchergebnisse des Modells denen eines Vektorraummodells

entsprechen. Die Suche im System basiert auf Spreading Activation. Das System wird detailliert in Kapitel 5.3 beschrieben.

(Crestani u. van Rijsbergen, 1993): Crestani u. van Rijsbergen (1993, 1997) stellen ein System auf Basis eines Neuronalen-Netz-Modells für die Suche nach Text-Dokumenten vor. Die Besonderheit an diesem System ist, dass es versucht Domänenwissen zu akquirieren, dieses in einer subsymbolischen Wissensrepräsentation zu speichern und für die Suche nach Dokumenten nutzbar zu machen. Das Modell basiert auf einem dreischichtigen Feedforward-Modell in dem eine Schicht die Anfragen repräsentiert, eine Schicht für das Wissensmodell steht und eine Schicht zur Repräsentation der Dokumente im System dient. Die Wissensrepräsentation im Netz wird durch Konzepte gebildet, welche durch Wörter beschrieben werden. Diese Wörter werden durch Knoten in der Anfrage- und Dokumentschicht repräsentiert. Konzepte werden durch Knoten in der Wissensrepräsentationsschicht repräsentiert.

Um Domänenwissen zu lernen dient die Anfrageschicht als Eingabeschicht und die Dokumentschicht als Ausgabeschicht für das Neuronale Netz, welches anhand einer Backpropagation-Regel (vgl. (Medler, 1998)) Domänenwissen lernt. Trainiert wird das Netz mit einer Menge von Dokumenten, für welche Relevanzbewertungen vorhanden sind.

(Crouch u. a., 1994b): Crouch u. a. (1994b,b) stellen ein Dokument-Retrieval-System vor, welches auf einem konnektionistischen Netzmodell basiert. Das Modell besteht aus drei Schichten, einer Schicht für Anfragen, einer Schicht für Terme und einer Schicht für Dokumente. Die Kanten zwischen den Knoten des Netzes werden so gewichtet, dass die Suchergebnisse des Modells denen eines Vektorraummodells entsprechen. Die Suche im System basiert auf Spreading Activation. Das System wird detailliert in Kapitel 5.3 beschrieben.

SYRENE (Mothe, 1994): Mothe (1994) präsentiert das assoziative Information-Retrieval-System *SYRENE*, welches auf einem Neuronalen-Netz-Modell basiert, für die Suche nach Text-Dokumenten. Das Modell besteht aus zwei Schichten, eine Schicht steht für Terme, die andere Schicht steht für Dokumente. Die Suche im System basiert auf Spreading Activation. Das System wird im Detail in Kapitel 5.3 beschrieben.

KDSA (Wolverton u. Hayes-Roth, 1994): Wolverton (1994); Wolverton u. Hayes-Roth (1994, 1995); Wolverton (1994) präsentieren einen Ansatz namens *Knowledge-Directed Spreading Activation* (*KDSA*) zur Suche in einer Wissensbasis. Einsatzgebiet ist die Suche von Analogien in Wissensbasen, welche eine hohe semantische Distanz aufweisen, um kreative Prozesse, wie das Entwickeln von Produktideen zu unterstützen. Bei der von Wolverton u. Hayes-Roth (1994) vorgestellten Version von Spreading Activation wird vor jedem Aktivierungsschritt jene Richtung für die Aktivierungsausbreitung selektiert, welche das Ziel der Suche am besten unterstützt. Somit handelt es sich hier um eine, dem in (Chen u. Dhar, 1991) vorgestellten Branch-and-Bound-Algorithmus ähnliche, gerichtete Version von Spreading Activation.

CodeFinder (Henninger, 1995): Henninger (1995, 1996, 1997) stellt *CodeFinder*, ein Retrieval-System für Software-Komponenten vor. Im Vorfeld extrahiert das Programm *Peel* aus Definitionen im Source-Code von Software-Komponenten Wörter, welche in Kommentaren und Funktionsnamen der Komponenten vorkommen. Aus diesen Informationen generiert CodeFinder ein Assoziatives Netz, welches aus zwei Schichten besteht. Eine Schicht repräsentiert Komponenten, die andere Schicht die extrahierten Wörter.

CodeFinder durchsucht das Netz mit Spreading Activation und ermöglicht Komponenten zu finden, welche den Suchtermen entsprechen oder aufgrund der Netzstruktur zu ihnen verwandt sind. CodeFinder kann auf der Basis von Relevanzbewertungen das Netz adaptieren, die Lernregel hierfür ist aus (Rose u. Belew, 1991) übernommen.

(Crestani, 1997b): Crestani (1997b) präsentiert ein System, welches auf einem Assoziativen Netz operiert und zur Suche in Hypertext dient. Der Hypertext kann durch den Benutzer navigiert werden und das Hypertext-Netz kann mit Spreading Activation durchsucht werden. Das System besteht aus drei Schichten, einer Schicht für Konzepte, einer Schicht für Indexterme und einer Schicht für Dokumente. Für die Erstellung des Netzes verweist Crestani (1997b) auf (Agosti u. Crestani, 1993) und (Agosti u. a., 1996). Dort wird beschrieben wie Term zu Term, Term zu Dokument und Dokument zu Dokument Assoziationen auf Grund statistischer Maße erstellt werden. Für

die Verbindungen zwischen Konzepten wird ein Thesaurus vorgeschlagen. Für die Verbindungen zwischen Konzepten und Termen wird auf (Agosti u. Marchetti, 1992) verwiesen.

WebSCSA (Crestani u. Lee, 2000): Crestani u. Lee (2000) präsentieren mit dem System *WebSCSA* (*Web Search by Constrained Spreading Activation*), eine Variation von Assoziativem Retrieval, welches auf Dokumente im Web abzielt. Hyperlinks zwischen Web-Seiten werden als Grundlage für deren Assoziation gesehen, alle Hyperlinks haben den gleichen Assoziationsgrad (d.h. werden gleich gewichtet). Im in (Crestani u. Lee, 2000) vorgestellten Prototyp wählt ein Benutzer eine Menge an für ihn relevanten Webseiten aus, welche das System als Ausgangspunkt für die Suche nutzt. Den Hyperlinks in den Seiten der Ausgangsmenge folgend, werden neue Seiten gefunden. Die Relevanz einer neu gefundenen Seite zur Ausgangsmenge wird anhand des Kosinusmaßes (vgl. Gleichung 5.4) berechnet.

(Woodruff u. a., 2000): Woodruff u. a. (2000) präsentieren ein Recommender-System für digitale Bücher. Ziel dieses Systems ist es aufgrund einer Menge an gelesenen Dokumenten neue Dokumente zu finden, welche für den Benutzer relevant sind. Über mehrere Ähnlichkeitsmaße, wie textbasierte Ähnlichkeit von Dokumenten oder bibliographische Information (Referenzierung oder Überlappung von Referenzen) werden Dokumente miteinander assoziiert. Hierzu wird ein Netz aufgebaut, in dem Knoten für Dokumente stehen, welche aufgrund der zuvor genannten Ähnlichkeitsmaße miteinander assoziiert sind. Ausgehend von Knoten, welche bereits gelesene Dokumente repräsentieren wird dieses Netz mittels Spreading Activation durchsucht.

(Heylighen u. Bollen, 2002): Heylighen u. Bollen (2002) stellen den Entwurf eines Recommender-Systems zur Verwendung im Umfeld Digitaler Bibliotheken vor. Dokumente werden miteinander assoziiert, wenn sie vom selben Benutzer konsultiert werden. Die Assoziationen der Dokumente werden mittels eines Assoziatives Netzes abgebildet. Mittels eines Spreading-Activation-Ansatzes können für jedes Dokument assoziierte Dokumente gefunden werden.

Ein weiterer Aspekt am vorgestellten System ist die Orientierung am Hebb'schen Lernen (Medler, 1998). Die Regel von Hebb besagt, dass Begriffe

im Gehirn miteinander assoziiert werden, wenn sie zur gleichen Zeit aktiviert werden. Ausgehend von der Hebb-Regel wird ein Verfallsfaktor in die Formel zur Berechnung der Stärke der Assoziation eingebaut. Das heißt je länger zwei Dokumente nicht mehr gemeinsam aktiviert wurden, desto niedriger ist ihr gemeinsamer Assoziationsgrad.

TextIllustrator (Hartmann u. Strothotte, 2002): Hartmann (2002); Hartmann u. a. (2002); Hartmann u. Strothotte (2002) beschreiben mit *TextIllustrator* einen Spreading-Activation-Ansatz zur Illustration von multimedialen Lehrbüchern. Ihr Ziel ist es die Komplexität von Illustrationen im Bereicht der Anatomie zu reduzieren. Ausgehend von einem kompletten 3D-Modell eines Körperteils, versuchen sie nur jene Teile des Modells zu visualisieren, welche für den Nutzer des Lehrbuches gegenwärtig relevant sind.

Die Textpassagen mit denen der Benutzer interagiert (anzeigen, klicken) lösen eine Spreading-Activation-Suche in der Wissensbasis aus, welche anatomische Konzepte und deren Zusammenhänge modelliert (Muskel, Knochen, etc.). Jene Konzepte mit der höchsten Aktivierung werden im 3D-Modell visualisiert. Zur Identifikation der zu aktivierenden Konzepte in der Wissensbasis existiert zu jedem Konzept eine Menge an textuellen Ausprägungen, welche mit der aktuellen Textpassage verglichen wird.

Ontocopi (Alani u. a., 2003): Das System *Ontocopi* (*Ontology-Based Community of Practice Identifier*) (Alani u. a., 2003, 2002; O'Hara u. a., 2002) versucht in Ontologien mittels Spreading Activation *Communities of Practice* zu identifizieren. Ausgehend von einer Menge an initial aktivierten Instanzen aus der Ontologie breitet sich Aktivierung über deren Eigenschaftswerte, in der Ontologie aus.

Selektiert beispielsweise der Nutzer eine Person in der Ontologie, retourniert Ontocopi Instanzen aus der Ontologie, welche mit diese Person in Verbindung stehen. Das können Co-Autoren, Kollegen oder Personen sein, welche auf derselben Veranstaltung waren. Individuen, welche mehrere oder wichtigere Eigenschaften teilen (stärker miteinander verbunden sind) werden höher gereiht. Die Gewichte der Relationen zwischen den Individuen können aufgrund des Typs der Relation, im Vorfeld, manuell festgelegt werden oder automatisch, in Abhängigkeit der Häufigkeit der Relation in der

Ontologie gewichtet werden.

Berger u. a. (2003): Berger u. a. (2003); Berger (2003); Berger u. a. (2004a,b) beschreiben ein adaptives Information-Retrieval-System auf der Basis eines Assoziativen Netzes, welches in ein Tourismus-Informationssystem integriert ist. Die Wissensbasis des Systems ist als Assoziatives Netz modelliert, welches Begriffe aus dem Tourismuswesen miteinander verbindet. Die Konzepte im Assoziativen Netz sind mit Datensätzen in einer Datenbank, welche Unterkünfte für Besucher beschreiben, verknüpft.

Beispielsweise kann der Datenbankeintrag zu einem bestimmten Hotel mit den Begriffen *Sauna* und *Tischtennis* aus der Wissensbasis verknüpft sein. *Sauna* ist des Weiteren mit dem Begriff *Wellness* verbunden, der wiederum mit *Whirlpool* und *Solarium* assoziiert ist. Sucht ein Benutzer nach einer Unterkunft welche eine Sauna anbietet, stehen die Chancen gut, dass zusätzlich auch Unterkünfte gefunden werden, welche alternative Wellness-Angebote offerieren.

Die Suche wird mittels eines Spreading-Activation-Ansatzes durchgeführt, die Kanten zwischen den Konzepten im Assoziativen Netz werden händisch, im Vorfeld gewichtet.

Huang u. a. (2004a): Huang u. a. (2004a) führen den Begriff des Assoziativen Retrievals im Kontext von *Recommender-Systeme*n ein. So sehen sie *Kollaboratives Filtern* (collaborative filtering) als einen Anwendungsfall von Assoziativem Retrieval, in dem Kunden mit Produkten und Produkte mit Kunden assoziiert sind. Durch das Folgen der Verbindungen zwischen Kunden und Produkten ist es möglich, Kunden mit gleichen Produktinteressen oder Produkte, welche die gleiche Kundengruppe haben zu identifizieren. Sie vergleichen drei verschiedene Spreading-Activation-Algorithmen, das Leaky-Capacitor-Modell, einen Branch-and-bound-Algorithmus und ein Hopfield-Netz (vgl. Abschnitt 4.4.3).

Die Arbeit in (Huang u. a., 2004a) basiert auf (Huang u. a., 2002) und (Huang u. a., 2004b), auch dort wird ein Recommender-System entwickelt, welches auf Basis eines Spreading-Activation-Algorithmus operiert, allerdings wird in beiden Fällen ausschließlich ein Hopfield-Netz verwendet. Zusätzlich zum Vergleich mehrerer Spreading-Activation-Ansätze wird in (Huang u. a., 2004a) Kollaboratives Filtern im Kontext von Assoziativem

Retrieval betrachtet.

(Rocha u. a., 2004a): Rocha u. a. (2004a,b) präsentieren zwei Prototypen, welche für die Suche auf einer Webseite, Volltextsuche mit der Suche in einer Ontologie kombinieren. Die Ontologie wird hierzu in ein Assoziatives Netz transformiert und mit einem Spreading-Activation-Ansatz durchsucht. Eine detailliere Beschreibung ist in Abschnitt 3.4 zu finden.

4.5.1 Zusammenfassung und Klassifikation

Tabelle 4.1 gibt einen Überblick über die Eigenschaften der in diesem Abschnitt vorgestellten Systeme. Alle der angeführten Systeme (bis auf (Salton, 1963) und (Salton, 1968)) teilen sich die Eigenschaft, dass es sich bei der zugrundeliegenden Repräsentationsform um eine Netzstruktur handelt, welche mit Spreading Activation (SA) durchsucht wird. Konkret wird in Tabelle 4.1 angeführt ob der vorgestellte Suchansatz durch dessen Autoren als Modell auf der Basis von Assoziativen Netzen (AN), Neuronalen Netzen (NN) oder Semantischen Netzen (SN) bezeichnet wird und ob Assoziatives Retrieval (AR) genutzt wird. Weiters wird gezeigt ob es sich beim vorgestellten System um ein System zur Suche in einer Wissensrepräsentation (WR) oder in einer Dokumentensammlung (DR) handelt. Zusätzliche Anmerkungen (Anm.) werden als Fußnoten angeführt. Wird ein Charakteristikum unterstützt wird dies durch ein X in der Zelle der Tabelle vermerkt. (X) signalisiert Grenzfälle, die ebenfalls in Fußnoten erklärt werden. Für die Beschriftung der Spalten werden folgende Abkürzungen verwendet:

- AR ... Assoziatives Retrieval
- AN ... Assoziative Netze
- NN ... Neuronale Netze
- SN ... Semantische Netze
- SA ... Spreading Activation
- WR ... Wissensrepräsentation
- D ... Dokumente
- Anm. ... Anmerkung

	AR	AN	NN	SN	SA	WR	D	Anm.
(Salton, 1963)	X				[3]		X	
(Salton, 1968)	X				[3]		X	
(Preece, 1981)	X				X		X	
(Belew, 1987)	X	X			X	(X)[4]	X	
(Cohen u. Kjeldsen, 1987)				X	X	X		
(Croft u. Thompson, 1987)				[5]	X	(X)[6]	X	
(Rose u. Belew, 1989)	X				X	X[7]	X	
(Wilkinson u. Hingston, 1991)			X		X		X	
(Crouch u. a., 1994b)	X		(X)[8]		X		X	
(Mothe, 1994)			X		X		X	
(Wolverton, 1994)				X	X	X		
(Crestani u. van Rijsbergen, 1993)	X	X	X		X	[9]	X	
(Henninger, 1995)		X			X	[10]	X	
(Crestani u. van Rijsbergen, 1997)	X	X	X		X	[9]	X	
(Crestani, 1997b)	X	X			X	[11]	X	
(Crestani u. Lee, 2000)	X	[12]			X		X	
(Woodruff u. a., 2000)	X	(X)[13]			X	(X)[14]	X	
(Hartmann u. Strothotte, 2002)				X	X	X		[15]
(Heylighen u. Bollen, 2002)		X			X		X	
(Alani u. a., 2003)					X	X		
(Berger u. a., 2003)		X			X	X		
(Huang u. a., 2004a)	X				X	[16]		

[3] Auf Basis des Vektorraummodells
[4] Es gibt Knoten für Autoren, Metadaten werden verwendet
[5] Wissensbasis soll durch Semantisches Netz realisiert werden
[6] Dokumente und bibliographische Informationen werden genutzt
[7] Hybrider Ansatz - experimentelle Implementierung symbolischer Kanten
[8] Crouch u. a. (1994b) bezeichnen ihr Netz nicht als Neuronales Netz sondern als konnektionistisches Netz
[9] Versucht Wissensbasis mit einem Neuronal Netz zu lernen
[10] Zusammenhänge zwischen Komponenten aus dem Modell der Software werden nicht genutzt
[11] Thesaurus als Wissensrepräsentation
[12] Suche in Hypertext
[13] Als *Activation Network* bezeichnet und durch Matrizen realisiert
[14] Bibliographische Informationen
[15] Suche in Wissensrepräsentation ausgehend von Text
[16] Recommender-System

	AR	AN	NN	SN	SA	WR	D	Anm.
(Rocha u. a., 2004a)		X			X	X		

Tabelle 4.1: Die vorgestellten Systeme im Überblick

4.6 Zusammenfassung

Dieses Kapitel führte die für diese Arbeit grundlegenden Begriffe des Assoziativen Retrieval, der Assoziativen Netze und von Spreading Activation ein. Im Anschluss daran folgte ein Überblick über Systeme, welche Assoziatives Retrieval, Assoziative Netze oder Spreading Activation zur Eigenschaft haben und zusammenfassend eine direkte Gegenüberstellung der Charakteristika der angeführten Systeme.

5 Netzmodelle im Vergleich zum Vektorraummodell

5.1 Einleitung

Das Vektorraummodell erfreut sich höher Popularität unter Forschern und Anwendern im Information-Retrieval-Bereich (Baeza-Yates u. Ribeiro-Neto, 1999). Dies wird zum Anlass genommen es der Familie der Netzmodelle gegenüber zu stellen und Gemeinsamkeiten der beiden Modelle aufzuzeigen. Die hier durchgeführte Untersuchung vertieft die Literaturstudie aus Abschnitt 4.5 und bereitet die Modellentwicklung in Kapitel 6 vor. Wiederum trägt die vollzogene Untersuchung zur Strukturierung existierender Methoden und zur Erhöhung der Konstruktvalidität des durchgeführten Experiments (vgl. Abschnitt 9.3.1) bei.

5.2 Das Vektorraummodell

Im Vektorraummodell werden sowohl die Dokumente in einem Information-Retrieval-System als auch die Anfrage an das System als Vektoren repräsentiert. Die Dimensionen der Vektoren entsprechen den Termen (Wörtern) in der Anfrage und den Dokumenten. Um jene Dokumente zu finden, welche einer Anfrage am besten entsprechen, wird zwischen dem Anfragevektor $\vec{q}$ und den Dokumentvektoren $\vec{d_j}$ eine Ähnlichkeit berechnet.

Der Anfrage-Vektor $\vec{q}$ ist wie in Gleichung 5.1 definiert.

$$\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q}) \tag{5.1}$$

Wobei gilt:

- $w_{i,q}$ stellt das Gewicht des i-ten Terms der Anfrage dar
- t ... Anzahl der Terme im System
- $w_{i,q} \geq 0$ und $1 \leq i \leq t$

Der Dokument-Vektor $\vec{d_j}$ des j-ten Dokuments im System ist wie in Gleichung 5.2 definiert.

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j}) \tag{5.2}$$

Wobei gilt:

- $w_{i,j}$ stellt das Gewicht des i-ten Terms des j-ten Dokuments im System dar
- t ... Anzahl der Terme im System
- $w_{i,j} \geq 0$ und $1 \leq i \leq t$

Detaillierte Erläuterungen zum Vektorraummodell sind in (Baeza-Yates u. Ribeiro-Neto, 1999) zu finden.

5.2.1 Bestimmung der Ähnlichkeit zwischen Anfrage und Dokumenten

Zur Berechnung der Ähnlichkeit zwischen einem Dokumentvektor und dem Anfragevektor sind unterschiedliche Maße, wie beispielsweise das Innere Produkt (auch Skalarprodukt) oder der Kosinus zwischen den beiden Vektoren gebräuchlich. Eine ausführliche Erklärung zu zahlreichen Ähnlichkeitsmaßen und deren Charakteristika ist in (Jones u. Furnas, 1987) zu finden.

Gleichung 5.3 zeigt die Berechnung der Ähnlichkeit zwischen einem Dokumentvektor $\vec{d_j}$ und dem Anfragevektor $\vec{q}$ mit Hilfe des Inneren Produkts (Skalarprodukt). Lange Dokumente, welche viele mit der Anfrage übereinstimmende Termen enthalten, welche jedoch selten vorkommen, werden durch dieses Maß hoch gereiht.

$$sim(\vec{d_j}, \vec{q}) = \vec{d_j} \cdot \vec{q} = \sum_{i=1}^{t} w_{i,j} \cdot w_{i,q} \tag{5.3}$$

Gleichung 5.4 zeigt das sogenannte *Kosinusmaß*, welches dem Kosinus des Winkels zwischen einem Dokumentvektor $\vec{d_j}$ und dem Anfragevektor $\vec{q}$ entspricht. Beim Kosinusmaß handelt es sich um eine normierte Version des Skalarprodukts. Anfragevektor und Dokumentvektor werden anhand ihre Länge normiert. Somit werden lange Dokumente nicht mehr bevorteilt.

$$sim(\vec{d_j}, \vec{q}) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{i,j} \cdot w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{5.4}$$

5.2.2 Gewichtung der Terme im Dokument

Neben einem Maß zur Berechnung der Ähnlichkeit zwischen Vektoren sind die Gewichtungsfunktionen für die Terme in den Dokumentvektoren ein wichtiger Bestandteil zur Berechnung der Relevanz eines Dokuments für eine Anfrage.

Ein klassisches Werkzeug zur Gewichtung der Terme in den Dokumentvektoren ist *tf*idf*. Tf*idf (auch *tfidf*, *tf–idf*[1] oder *tf/idf*[2]) ist eine Maß für die Wichtigkeit eines Wortes (Terms) für ein Dokument in einer Dokumentensammlung. Die Wichtigkeit ist umso höher, je häufiger das Wort im Dokument vorkommt und je weniger häufig das Wort in den anderen Dokumenten der Dokumentenmenge vorkommt. Tf*idf besteht aus den beiden Faktoren *Termfrequenz* (*term frequencey*, *tf*) und *Inverse Dokumenthäufigkeit* (*inverse document frequency*, *idf*). Es gibt unterschiedliche Varianten wie Termfrequenz und Inverse Dokumenthäufigkeit berechnet werden.

Die Termfrequenz basiert auf der Frequenz eines Terms k_i in einem Dokument d_j. Gleichung 5.5 zeigt die normalisierte Termfrequenz. Die Normalisierung erfolgt durch die Division durch die Termfrequenz des Terms, welcher am häufigsten im Dokument vorkommt. Hierdurch wird die Termfrequenz auf den Wertebereich [0..1] normalisiert.

[1] - ist als Bindestrich zu verstehen und nicht als Subtraktion
[2] / ist als Schrägstrich zu verstehen und nicht als Division

$$tf_{i,j} = \frac{freq_{i,j}}{max_l(freq_{l,j})} \tag{5.5}$$

Für Gleichung 5.5 gilt:

- $tf_{i,j}$ ... (Normalisierte) Termfrequenz des Terms k_i im Dokument d_j
- $freq_{i,j}$ ... Frequenz des Terms k_i im Dokument d_j
- t ... Anzahl der Terme in Dokument d_j

Terme, welche in sehr vielen Dokumenten vorkommen eigenen sich nicht gut um relevante Dokumente von nicht relevanten Dokumente zu trennen. Um Terme, welche in sehr vielen Dokumenten der Dokumentensammlung vorkommen niedriger zu gewichten wird die inverse Dokumenthäufigkeit genutzt. Gleichung 5.6 zeigt deren Berechnung.

$$idf_i = \log \frac{N}{n_i} \tag{5.6}$$

Wobei gilt:

- idf_i ... Inverse Dokumenthäufigkeit des Terms k_i
- N ... Anzahl aller Dokumente in der Dokumentensammlung
- n_i ... Anzahl der Dokumente in denen der Terms k_i vorkommt.

Aus dem Produkt der beiden Faktoren tf und idf ergibt sich die Termfrequenz und Inverse Dokumenthäufigkeit (tf*idf). Gleichung 5.7 zeigt die Berechnung von tf*idf und die Verwendung für die Gewichtung ($w_{i,j}$) von Termen i in Dokumenten j.

$$w_{i,j} = tfidf_{i,j} = tf_{i,j} \cdot idf_i = \frac{freq_{i,j}}{max_l(freq_{l,j})} \cdot \log \frac{N}{n_i} \tag{5.7}$$

5.2.3 Gewichtung der Terme in der Anfrage

Auch für die Gewichtung ($w_{i,q}$) der Terme i in einer Anfrage q ist eine tf*idf-basierte Gewichtung gebräuchlich. Gleichung 5.8 zeigt die von Salton u. Buckley (1988b) vorgeschlagene Gewichtung der Terme in der Anfrage.

$$w_{i,q} = (0,5 + \frac{0,5 \cdot freq_{i,q}}{max_l(freq_{l,q})}) \cdot \log \frac{N}{n_i} \tag{5.8}$$

5.3 Gemeinsamkeiten von Netzmodellen mit dem Vektorraummodell

Im Folgenden werden drei Ansätze präsentiert, die Gleichheiten zwischen dem Vektorraummodell und Netzmodellen zeigen.

(Wilkinson u. Hingston, 1991): Wilkinson u. Hingston (1991, 1992) stellen ein Netzmodell mit drei Schichten vor. Das Netz besteht aus einer Schicht Q an Knoten, welche Anfrageterme repräsentieren (vgl. Gleichung 5.9), einer Schicht T an Knoten, welche Dokumentterme repräsentieren (vgl. Gleichung 5.10) und einer Schicht D an Knoten, welche für Dokumente stehen (vgl. Gleichung 5.11). Die Knoten der Anfragetermschicht sind mit den Knoten der Dokumenttermschicht verbunden. Die Verbindungen zwischen Anfragetermknoten und Dokumenttermknoten sind gewichtet ($\overline{w}_{i,q}$, siehe Gleichung 5.12). Die Knoten der Dokumenttermschicht sind mit den Knoten der Dokumentschicht verbunden, die Kanten zwischen den beiden Schichten sind gewichtet ($\overline{w}_{i,j}$, siehe Gleichung 5.13). Die Knoten innerhalb einer Schicht sind nicht miteinander verbunden. Abbildung 5.1 zeigt die Struktur des Netzes.

$$Q = \{q_1, ..., q_t\} \tag{5.9}$$

$$T = \{t_1, ..., t_t\} \tag{5.10}$$

$$D = \{d_1, ..., d_N\} \tag{5.11}$$

Wobei für Gleichung 5.9, Gleichung 5.10 und Gleichung 5.11 gilt:

- t ist gleich der Anzahl der Terme im System
- N ist gleich der Anzahl der Dokumente im System

Um eine Reihung der Dokumente in Abhängigkeit der Anfrage zu erzielen fließt Aktivierung von einer Menge an initial aktivierten Anfragetermknoten, über die Dokumenttermknoten zu den Dokumentknoten. Jene Anfragetermknoten werden aktiviert, welche Terme repräsentieren, die in der Anfrage vertreten sind. Zu diesem Zweck wird den Anfragetermknoten initial ein fixer Wert an Aktivierung zugewiesen (die Knoten werden aktiviert).

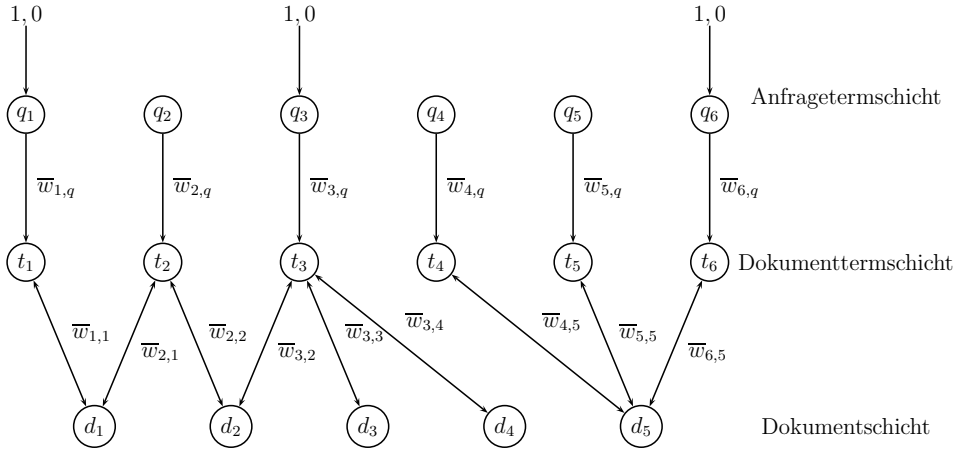

Abbildung 5.1: Netzmodell mit einer Schicht für Anfrageterme, Dokumentterme und Dokumente (nach (Wilkinson u. Hingston, 1991))

In (Wilkinson u. Hingston, 1991) beträgt der initiale Aktivierungsgrad der Anfragetermknoten $1,0$.

Die Besonderheit an dem vorgestellten Netzmodell ist, dass es derart konfiguriert ist, dass es bei der Reihung der Suchergebnisse äquivalente Ergebnisse zum Vektorraummodell mit Kosinusmaß und tf*idf liefert. Auch Baeza-Yates u. Ribeiro-Neto (1999) stellen dieses System vor. Um eine bessere Vergleichbarkeit mit dem in Abschnitt 5.2 beschriebenen Vektorraummodell zu ermöglichen wird im Folgenden die Notation aus (Baeza-Yates u. Ribeiro-Neto, 1999) verwendet.

Gleichung 5.12 zeigt das Kantengewicht $\overline{w}_{i,q}$ einer Kante von einem Anfrageknoten q_i zu einem Dokumenttermknoten t_i.

$$\overline{w}_{i,q} = \frac{w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \tag{5.12}$$

Wobei gilt:

- $w_{i,q}$ ... Gewicht des i-ten Terms in der Anfrage
- t ... Anzahl der Terme im System
- $w_{i,q} \geq 0$ und $1 \leq i \leq t$

Gleichung 5.13 zeigt das Kantengewicht $\overline{w}_{i,j}$ einer Kante von einem Dokumentermkknoten t_i zu einem Dokumentknoten d_j.

$$\overline{w}_{i,j} = \frac{w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2}} \tag{5.13}$$

Wobei gilt:

- $w_{i,j}$ ... Gewicht des i-ten Terms im j-ten Dokument im System
- t ... Anzahl der Terme im System
- $w_{i,j} \geq 0$ und $1 \leq i \leq t$

Nach der initialen Ausbreitung der Aktivierung von der Anfragetermschicht zur Dokumenttermschicht werden im nächsten Schritt die Knoten der Dokumentschicht aktiviert. Am Ende dieses Prozesse hat ein Dokumentknoten d_j den Aktivierungswert $A(d_j)$. Gleichung 5.14 zeigt die Berechnung der Aktivierung eines Dokumentknotens.

$$A(d_j) = \sum_{i=1}^{t} \overline{w}_{i,q} \cdot \overline{w}_{i,j} = \frac{\sum_{i=1}^{t} w_{i,q} \cdot w_{i,j}}{\sqrt{\sum_{i=1}^{t} w_{i,q}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{i,j}^2}} \tag{5.14}$$

Die Aktivierungswerte der Dokumentknoten, die durch diesen Ansatz erzielt werden, entsprechen exakt den durch das Vektorraummodell mit Kosinusmaß und tf*idf berechneten Ähnlichkeiten zwischen Anfrage- und Dokumentvektor. Durch die Verwendung der Notation aus (Baeza-Yates u. Ribeiro-Neto, 1999) ist ein direkter Vergleich mit dem Vektorraummodell einfach möglich (vgl. Gleichung 5.4).

(Crouch u. a., 1994b): Crouch u. a. (1994b,b) stellen ein Netzmodell mit drei Schichten vor. Das Netz besteht aus einer Schicht aus Knoten welche Anfragen repräsentieren, einer Schicht aus Knoten welche Terme repräsentieren und einer Schicht aus Knoten welche für Dokumente stehen. Die Knoten der Anfrageschicht sind mit den Knoten der Termschicht verbunden. Die Verbindungen zwischen Anfrageknoten und Termknoten sind gewichtet. Die Knoten der Termschicht sind mit den Knoten der Dokumentschicht verbunden, die Kanten zwischen den beiden Schichten sind gewichtet. Die Knoten innerhalb einer Schicht sind nicht miteinander verbunden. Abbildung 5.2 zeigt die Struktur des Netzes, die in der Abbildung verwendetet Notation entspricht der von Abbildung 5.1.

Das von Crouch u. a. (1994b) vorgestellte Modell ähnelt dem in (Wilkinson u. Hingston, 1991) (vgl. Abschnitt 5.3) beschriebenen. Der Unterschied zwischen den beiden Modellen ist, dass in (Crouch u. a., 1994b) eine Schicht für Anfragen existiert, deren Knoten Anfragen darstellen, während in (Wilkinson u. Hingston, 1991) eine Schicht von Anfragetermen existiert, deren Knoten die Terme von Anfragen an das System repräsentieren.

Um eine Reihung der Dokumente in Abhängigkeit der Anfrage zu erzielen, fließt Aktivierung vom initial aktivierten Anfrageknoten über die Termknoten zu den Dokumentknoten. Ein Anfrageknoten ist mit jenen Termknoten verbunden, welche Terme repräsentieren, die in der Anfrage vorkommen. Um den Anfrageknoten zu aktivieren wird diesem initial ein fixer Aktivierungswert zugewiesen (der Knoten wird aktiviert).

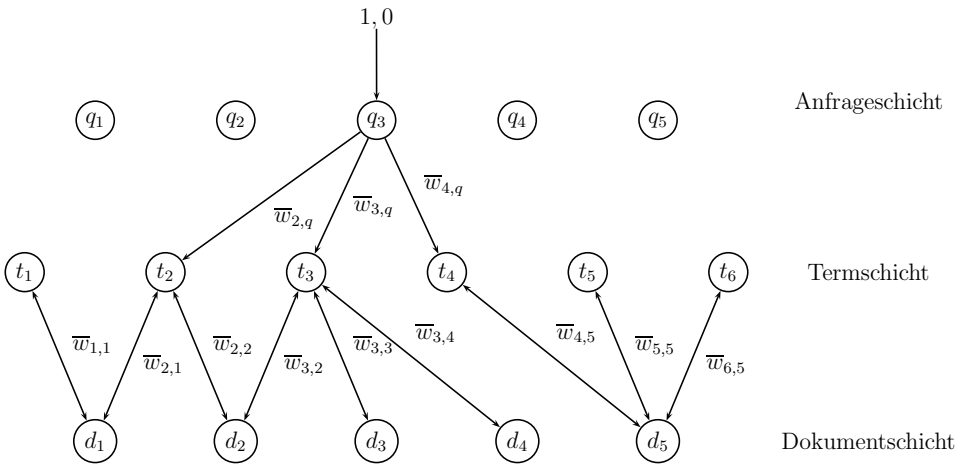

Abbildung 5.2: Netzmodell mit einer Schicht für Anfragen, Terme und Dokumente (nach (Crouch u. a., 1994b))

Crouch u. a. (1994b) führen keine Gleichungen an, die beschreiben wie ihr Netzmodell genau konfiguriert ist. Die textuelle Beschreibung lässt allerdings vermuten, dass es ident mit von Wilkinson u. Hingston (1991) (vgl. Abschnitt 5.3) vorgestellten parametriert ist.

Crouch u. a. (1994b) vergleichen die Ergebnisse ihres Systems mit denen des Vektorraummodell-basierenden Systems SMART (Buckley, 1985). In beiden Fällen werden Terme mit tf*idf gewichtet und Dokument- und Anfrage-Vektoren mit der L2-Norm normalisiert. Aus den beinahe identen

Ergebnissen[3] der beiden Systeme schließen Crouch u. a. (1994b), dass ihr konnektionistisches Modell äquivalent gute Suchergebnisse wie das Vektorraummodell liefert.

In (Crouch u. a., 1994a) fokussieren die Autoren auf Assoziatives Retrieval mit ihrem Modell. Nach der ersten Ausbreitung der Aktivierung von der Term- zur Dokumentschicht wird die Aktivierung für zwei weitere Schritte fortgesetzt. Das heißt, die Aktivierung breitet sich von den Dokumentknoten zurück zu den Termknoten und von dort erneut zu den Dokumentknoten aus[4]. Crouch u. a. (1994a) testen diesen Ansatz mit zwei Dokumentensammlungen und stellen fest, dass er bei einer Dokumentensammlung (Medlars) gute Ergebnisse liefert, bei einer anderen (CACM) allerdings die Suchergebnisse verschlechtert. Crouch u. a. (1994a) führen dies darauf zurück, dass in der CACM Collection unter den ersten fünf Dokumenten nur ein bis zwei relevante Dokumente vorkommen. Somit werden für die Suche nach assoziierten Dokumenten viele Terme aus Dokumenten verwendet, welche nicht für die Anfrage relevant sind. Crouch u. a. (1994a) stellen fest, dass die Verwendung eines Schwellwertes die Ergebnisse zusätzlich verbessert. Das bedeutet ein Knoten darf nur dann Aktivierung weitergeben, wenn seine Aktivierung einen gewissen Schwellwert überschritten hat. Dieser Schwellwert wurde empirisch ermittelt.

SYRENE (Mothe, 1994): Mothe (1994) stellt ein assoziatives Information-Retrieval-System vor, welches auf einem Netzmodell mit zwei Schichten basiert. Eine Schicht besteht aus Knoten, welche Indexterme i repräsentieren, die andere Schicht aus Knoten, welche für die Dokumente j im System stehen. Die Verbindungen zwischen den Knoten der beiden Schichten sind bidirektional und gewichtetet.

Im Gegensatz zu dem in (Wilkinson u. Hingston, 1991) präsentierten Modell, gibt es keine eigene Schicht für die Anfrageterme. Jene Knoten, welche zu den Termen in der Anfrage korrespondieren, werden bei einer Anfrage durch einen externen Stimulus aktiviert, welcher den Aktivierungsprozess im Netz initiiert. Die Höhe des externen Stimulus entspricht dem Gewicht

[3] Die maximale Differenz eines Precision-Werts der beiden System beträgt 0,0018.

[4] Daher auch die bidirektionalen Pfeile zwischen den Termknoten und den Dokumentknoten in Abbildung 5.2

des Terms in der Anfrage. Abbildung 5.3 zeigt die Struktur des Netzes.

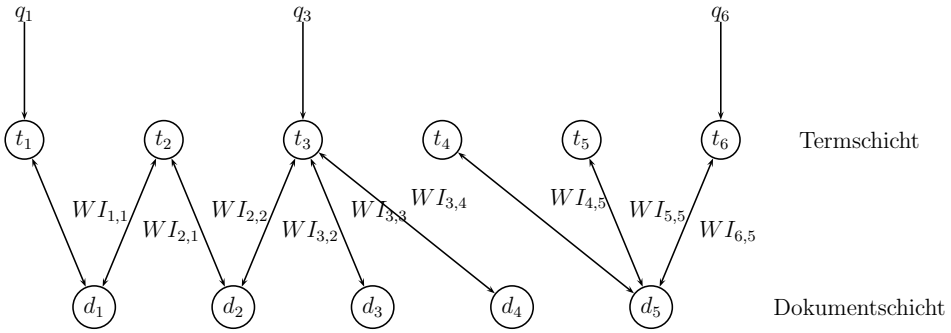

Abbildung 5.3: Netzmodell mit je einer Schicht für Terme und Dokumente (nach (Mothe, 1994))

Ein Knoten wird in (Mothe, 1994) als Zelle (cell) bezeichnet. Das Gewicht der Verbindung $WI_{i,j}$ zwischen einem Termknoten und einem Dokumentknoten ist äquivalent zur Wichtigkeit des Terms i für das Dokument j. Zur Gewichtung der Kanten zwischen einem Termknoten und einem Dokumentknoten wird die Häufigkeit eines Terms i (Termfrequenz, tf) im Dokument j verwendet, welche durch f_{ij} repräsentiert wird (siehe Erklärung zu Gleichung 5.16). Jeder Knoten der für einen Term i steht wird mit $TW(i)$, welches der inversen Dokumentfrequenz (idf) des Terms in der Dokumentsammlung entspricht, gewichtet (siehe Erklärung zu Gleichung 5.16).

Somit erfolgt in dem von Mothe (1994) vorgestellten Modell so wie in (Wilkinson u. Hingston, 1991) eine Gewichtung der Verbindungen zwischen Termen und Dokument mittels tf*idf, allerdings werden in (Mothe, 1994) die beiden Faktoren tf und idf auf Termknoten (gewichtet mit idf) und Verbindung zwischen Termknoten und Dokumentknoten (gewichtet mit tf) aufgeteilt. Im in (Wilkinson u. Hingston, 1991) vorgestellten Modell erfolgt ausschließlich eine Gewichtung der Kanten mit $w_{i,j}$.

Die Aktivierung eines (Dokument-)Knotens $E_j^D(\tau)$[5] wird in Abhängigkeit der Aktivierung $E_j^D(\tau - 1)$, die er im vorherigen Schritt hatte, der Aktivierung $E_i^t(\tau - 1)$ jener Knoten mit denen er verbunden ist und dem Gewicht der Verbindungen WI_{ij} zu diesen Knoten berechnet.

[5] in (Mothe, 1994) wird Aktivierung durch E symbolisiert

Gleichung 5.15 repräsentiert diesen Zusammenhang durch die Funktion f.

$$E_j^D(\tau = 1) = f(E_i^T(\tau = 0), WI_{ij}, E_j^D(\tau = 0)) \tag{5.15}$$

Für (Mothe, 1994) folgt daraus im Speziellen:

$$E_j^D(\tau = 1) = \sum_{i=1}^{n} E_i^T(\tau = 0) \cdot TW(i) \cdot WI_{ij} \tag{5.16}$$

Wobei gilt:

- E_j^D ... Aktivierung des Dokumentknotens zu Dokument j
- E_i^t ... Aktivierung des Termknotens zu Term i
- τ ... Die Nummer des Schrittes
- $TW(i)$ ... Gewicht des Terms i in der Dokumentensammlung, es ist dem Knoten zugeordnet, welcher den Term repräsentiert
- $TW(i) = idf_i$
- idf_i ... Inverse Dokumentfrequenz des Terms i
- WI_{ij} ... Gewicht der Verbindung zwischen einem Termknoten i und einem Dokumentknoten j
- $WI_{ij} = f_{ij}$
- f_{ij} ... Termfrequenz eines Terms i in einem Dokument j

Um eine einheitliche Benennung zu Abschnitt 5.2 zu behalten sei darauf hingewiesen, dass gilt:

$$w_{i,q} \equiv E_i^T(\tau = 0) \tag{5.17}$$

$$w_{i,j} \equiv WI_{ij} \cdot TW(i) \tag{5.18}$$

Mothe (1994) vergleicht die Aktivierungsausbreitungsfunktion ihres Modells (siehe Gleichung 5.19) mit einem Vektorraummodell welche die Ähnlichkeit zwischen Dokument und Anfrage mittels des Skalarprodukts und tf*idf berechnet (siehe Gleichung 5.20). Sie zeigt, dass beide Modelle eine idente Reihung der Ergebnisse zu einer Anfrage liefern und die Ergebnisse, bei gleichen Werten für tf und idf, gleich gewichten.

$$
\begin{aligned}
E_j^D(\tau=1) &= \sum_{i=1}^{n} E_i^T(\tau=0)\cdot TW(i)\cdot WI_{ij} \\
&= E_1^T(\tau=0)\cdot WI_{1j}\cdot TW(1)+\cdots+E_t^T(\tau=0)\cdot WI_{tj}\cdot TW(t) \\
&= f_{1j}\cdot idf_1\cdot q_1+\cdots+f_{tj}\cdot idf_t\cdot q_t \qquad (5.19)\\
sim(\vec{d_j},\vec{q}) &= \vec{d_j}\cdot\vec{q} \\
&= w_{1,j}\cdot w_{1,q}+\cdots+w_{t,j}\cdot w_{t,q} \\
&= f_{1j}\cdot idf_1\cdot q_1+\cdots+f_{tj}\cdot idf_t\cdot q_t \qquad (5.20)
\end{aligned}
$$

Darauf aufbauend beschreibt Mothe (1994) Ausbreitungsfunktionen, welche identische Ergebnisse zu anderen Ähnlichkeitsmaßen liefern. Gleichung 5.21 zeigt die Ausbreitungsfunktion, welche dem Kosinusmaß entspricht.

$$
E_j^D(\tau=1)=\frac{\sum_{i=1}^{n} E_i^T(\tau=0)\cdot TW(i)\cdot WI_{ij}}{\sqrt{\sum_{i=1}^{n}(E_i^T(\tau=0))^2}\cdot\sqrt{\sum_{i=1}^{n}(TW(i)\cdot WI_{ij})^2}} \qquad (5.21)
$$

(Salton u. Buckley, 1988a): Salton u. Buckley (1988a) vergleichen, für die Suche nach Textdokumenten, verschiedene Konfigurationen von Spreading-Activation-Modellen mit verschiedenen Konfigurationen des Vektorraummodells. Zu diesem Zweck werden die von den Modellen verwendeten Ähnlichkeitsfunktionen für die Berechnung der Ähnlichkeit zwischen Anfrage und Dokument gegenüber gestellt. Vier Ähnlichkeitsfunktionen basieren auf dem Spreading-Activation-Modell, wie es in (Jones u. Furnas, 1987) beschrieben wurde, vier Ähnlichkeitsfunktionen basieren auf dem Vektorraummodell. Während bei den Spreading-Activation-Varianten mit L1- und L2-Normalisierung und Normalisierung der Dokumentenlänge experimentiert wird, wird bei den Vektorraummodellen die Normalisierung der Dokumentlänge und die Termgewichtung mit tf*idf (vgl. Abschnitt 5.2.2) variiert. Salton u. Buckley (1988a) kommen zum Schluss, dass die getesteten Vektorraum-Konfigurationen bessere Ergebnisse als die getesteten Spreading-Activation-Modelle liefern. Crestani (1997a) merkt an, dass bei dieser Studie sehr einfache Versionen von Spreading-Activation-Modellen verwendet wurden, welche dem Vektorraummodell sehr ähnlich

sind. Zusätzlich ist festzuhalten, dass die besten Ergebnisse durch jene Ähnlichkeitsfunktionen erzielt wurden, welche idf (vgl. Abschnitt 5.2.2) enthielten und keines der Spreading-Activation-Modelle mit idf konfiguriert wurde. Den Ergebnissen von Mothe (1994) folgend kann ein Spreading-Activation-Modell so konfiguriert werden, dass die verwendetet Ähnlichkeitsfunktion der eines Vektorraummodells entspricht.

5.4 Zusammenfassung

Spreading-Activation-Netze können so konfiguriert werden, dass die gelieferten Suchergebnisse denen eines Vektorraummodells entsprechen. Folgende Punkte müssen hierzu beachtet werden:

1. Wahl der Aktivierungsfunktion
   Das Netz muss eine lineare Aktivierungsfunktion verwenden (vgl. Abschnitt 4.4.1). Nur unter Verwendung einer lineare Aktivierungsfunktion kann die Ähnlichkeitsfunktion eines Spreading-Activation-Modells gleich der eines Vektorraummodells sein.
2. Gewichtung der Kanten
   Die Kantengewichte zwischen Term- und Dokumentknoten müssen den Termgewichten der Dokumentvektoren des Vektorraummodells entsprechen. Knoten im Netz können für Terme und Dokumente stehen. Die Kanten zwischen Term- und Dokumentknoten können äquivalent zum Vektorraummodell, beispielsweise auf Basis von tf*idf (vgl. Abschnitt 5.2.2), gewichtet werden.
3. Anzahl der Iterationen
   Soll das Spreading-Activation-Modell einem typischen Vektorraummodell (vgl. Abschnitt 5.2) entsprechen, dürfen nicht mehr als zwei Iterationen der Aktivierungsausbreitung (von der Anfrageschicht zur Termschicht und von der Termschicht zur Dokumentschicht) verwendet werden.

6 Ein Modell für Assoziatives Retrieval im Semantic Web

6.1 Einleitung

In diesem Abschnitt wird das im Rahmen dieser Arbeit entwickelte Modell beschrieben. Modelle im Information Retrieval legen eine Repräsentation für Anfragen und zu suchende Ressourcen (z.B. Dokumente) fest und beschreiben wie die Relevanz eines Ergebnisses für eine Anfrage ermittelt wird (vgl. Abschnitt 2.3). Diese durch ein Information-Retrieval-Modell formal festgelegte Vorgehensweise kann in einem Information-Retrieval-System implementiert werden.

Das in dieser Arbeit entwickelte Modell ermöglicht die Suche nach Ressourcen[1] im Semantic Web. Ausgehend von einer Menge an Konzepten aus einer Ontologie wird eine, in Abhängigkeit der Anfrage gereihte, Menge an Suchergebnissen retourniert. Zusätzlich erlaubt das vorgestellte Modell auch eine assoziative Suche basierend auf semantischer und/oder inhaltsbasierter Ähnlichkeit. Beide Ansätze zur Suche werden in einem Modell miteinander vereint. Sowohl bei der assoziativen Suche als auch der Vereinigung von assoziativer Suche mit exakter Suche in einem Modell handelt es sich, basierend auf den Untersuchungen in Abschnitt 3.4, um einen neuen Ansatz im Kontext der Suche im Semantic Web.

Während in diesem Kapitel auf das entwickelte Modell eingegangen wird,

[1] Der Begriff der Ressource wird in Abschnitt 2.2.1 eingeführt. Im Kontext des Semantic Web kann alles für das ein URI vergeben werden kann ein Ressource sein.

folgt in Kapitel 7 eine Beschreibung der Umsetzung dieses Modell in einer Semantic-Desktop-Umgebung zur Suche nach Dokumenten. Da Semantic Web und Semantic Desktop auf den gleichen Technologien aufbauen ist für eine Implementierung des Modells in einem System für die Suche im Semantic Web keine Modifikation des Modells notwendig. Allerdings müssen für eine Implementierung im Semantic Web andere Wege der Datenbeschaffung beschritten werden als am Semantic Desktop (vgl. Abschnitt 9.4.2).

6.2 Motivation

Abschnitt 3 gibt einen Überblick über aktuelle Ansätze zur Suche im Semantic Web. Das Semantic Web und im Zuge dessen auch Suchansätze für das Semantic Web befinden sich in einer frühen Phase der Entwicklung. Erst wenige Methoden zur Suche im Semantic Web und am Semantic Desktop existieren, und bei der Menge an verfügbaren Modellen und Systemen handelt es sich um ein inhomogenes, experimentelles Feld. Das in dieser Arbeit entwickelte Modell soll einen Beitrag zu diesem Feld des Information Retrieval im Semantic Web leisten.

Zahlreiche der in Abschnitt 3 untersuchten Modelle und Systeme basieren auf der Verwendung von semantischen Metadaten, also Metadaten für deren Definition auf das in einer Ontologie spezifizierte Wissen zurückgegriffen wird. Durch das Fehlen von semantischen Metadaten wird die Suche nach Ressourcen im Semantic Web basierend auf dieser Information erschwert. Szenarien in denen eine limitierte Anzahl an textuellen Inhalten von Ressourcen zur Verfügung steht und somit primär Suchen basierend auf semantischen Metadaten durchgeführt werden können, wie das Auffinden von multimedialen Inhalten oder (Semantic) Web Services, werden hierdurch erschwert. Aber auch die postulierten Vorteile von semantischer Zusatzinformation für die text-basierte Suche können nicht genutzt werden (vgl. Abschnitt 3.3).

Huang u. a. (2004a) adressieren ein verwandtes Problem im Kontext von Recommender-Systemen. Sie modellieren Kunden und Produkte als Knoten in einem Graphen. Kauft eine Kunde ein Produkt wird eine Verbindungen zwischen dem Knoten, der den Kunden repräsentiert und dem Knoten, der für das Produkt steht, im Graphen erzeugt. Beim erstmaligen Starten

eines Recommender-Systems, dem Kaltstart, hat das System noch keine Rückmeldungen durch Benutzer gesammelt. In der Datenbasis, auf der das Recommender-System arbeitet, existieren noch keine, oder wenige, händisch erzeugte Verbindungen zwischen Kunden und Produkten. Hierdurch wird das Vorschlagen von Produkten an Kunden erschwert. Huang u. a. (2004a) zeigen wie dieses Problem mit Hilfe eines Assoziativen Suchansatzes erfolgreich adressiert werden kann (vgl. Abschnitt 4.5).

Ähnlich zum Kaltstart eines Recommender-Systems, wenn nur wenige Verbindungen zwischen Kunden und Produkten verfügbar sind, wird im Semantic Web die Suche erschwert, wenn nur wenige Verbindungen zwischen Ressourcen und semantischen Metadaten vorhanden sind. Ausgehend von einer Anfrage, welche auf Teilen des Graphen basiert, können nur wenige Resultate gefunden werden.

Um dieses Problem zu adressieren soll das in dieser Arbeit entwickelte Modell, ähnlich wie in (Huang u. a., 2004a), die Möglichkeit einer assoziativen Suche bieten. Diese Form der Suche wird mit nicht-assoziativer Suche für das Semantic Web in einem Modell vereint.

6.3 Das Retrieval-Modell

Im Folgenden wird das in dieser Arbeit entwickelte assoziative Retrieval-Modell vorgestellt. Die Beschreibung des Modells erfolgt in Anlehnung an die Systembeschreibungen in Abschnitt 5.

Das Modell stellt eine *grundlegende Suchfunktionalität* zur Verfügung um, ausgehend von einer Menge an Konzepten, eine Menge an Ressourcen zu finden und diese in Abhängigkeit der Anfrage zu reihen. Zusätzlich zu dieser grundlegenden Suchfunktionalität sollen mittels einer erweiterten, assoziativen Suche auch Ressource gefunden werden, welche mit ähnlichen Konzepten zu jenen in der Anfrage annotiert wurden. Des Weiteren sollen ähnliche Ressourcen zu den Ressourcen in der Ergebnismenge gesucht werden. Diese *erweiterte Suchfunktionalität* wurde mit dem Ziel entwickelt relevante Ressourcen auffindbar zu machen auch, wenn diese nicht mit Konzepten aus der Domänenontologie annotiert wurden bzw., wenn zur Annotation der Ressourcen andere Konzepte als jene in der Anfrage verwendet wurden.

Sowohl für die grundlegende Suche als auch die erweiterte, assoziative, Suche wird ein Netzmodell verwendet, welches mit einem Spreading-Activation-Ansatz durchsucht wird.

Neben dem Ansatz, der für die Suche verwendet wird (vgl. Abschnitt 6.3.6), wird in diesem Kapitel auch auf den Einsatz von semantischen Annotationen für die Suche eingegangen (vgl. Abschnitt 6.3.1) und der Aufbau jener Netzstruktur beschrieben, welche der Suche zugrunde liegt (vgl. Abschnitt 6.3.2). Des Weiteren wird dargestellt wie semantische Annotationen (vgl. Abschnitt 6.3.3) und Ähnlichkeit zwischen zwei Konzepten (vgl. Abschnitt 6.3.4) und zwischen zwei Ressourcen (vgl. Abschnitt 6.3.5) als Kanten im Netz modelliert werden.

6.3.1 Semantische Annotation von Ressourcen

Um existierende, semantische Information für eine Suche nach Ressourcen nutzbar machen zu können, muss diese mit der zu durchsuchenden Ressourcenmenge verbunden werden. Ressourcen werden mit Elementen aus einer Ontologie *semantisch annotiert*.

Handschuh u. Staab (2003) definieren semantische Annotation als den Vorgang bestehende Daten mit semantischen Metadaten zu versehen, welche diese Daten beschreiben:

> *[Semantic annotation] frequently involves the decoration of existing data, e.g. plain text, that is only understandable for the human with semantic metadata that describes, e.g., the text.*
>
> (Handschuh u. Staab, 2003)

Handschuh (2007) weißt darauf hin, dass mit *semantischer Annotation* sowohl der Prozess der semantischen Annotation als auch das Ergebnis dieses Prozesses, die semantische Annotation, bezeichnet wird. Weiters hält Handschuh (2007) fest, dass es sich bei semantischen Metadaten um Fakten handelt, welche einer Domänenontologie entstammen bzw. mit einer Domänenontologie in Verbindung stehen.

Aktuell existieren verschiedene Ansätze zur semantischen Annotation, wobei prinzipiell zwischen manueller und semi-automatischer Annotation

unterschieden werden kann. Während bei der manuellen Annotation die Vergabe von Metadaten ausschließlich durch den Benutzer erfolgt, wird bei der semi-automatischen Annotation versucht den Benutzer durch Programmfunktionalität zu unterstützen. Einführungen in das Thema semantische Annotation und ein Überblick über existierende Ansätze sind in (Handschuh u. Staab, 2003), (Reeve u. Han, 2005) und (Reif, 2006) zu finden.

Handschuh (2007) zeigt auf, dass es in der Literatur unterschiedliche Zugänge zur semantischen Annotation gibt und verweist auf Bechhofer u. Goble (2001), welche folgende Arten der semantischen Annotation unterscheiden:

- *Decoration*: Annotation der Ressource mit einem Kommentar des Benutzers.
- *Linking*: Annotation der Ressource mit weiterführenden Links.
- *Instance identification*: Annotation der Ressource mit einem Konzept. Die annotierte Ressource ist eine Instanz des Konzepts.
- *Instance reference*: Annotation der Ressource mit einem Konzept. Die annotierte Ressource referenziert ein Individuum in der Welt, welches eine Instanz des Konzepts ist.
- *Aboutness*: Annotation der Ressource mit einem Konzept. Die annotierte Ressource handelt vom Konzept.
- *Pertinence*: Annotation der Ressource mit einem Konzept. Die annotierte Ressource gibt weiterführende Information über das Konzept.

Die semantischen Annotation im vorliegenden System basieren auf der *Aboutness* der Ressourcen. Das heißt, eine Ressource wird mit einem Konzept versehen, wenn der Inhalt der Ressource vom Konzept handelt. Bei diesem Ansatz werden Ressourcen mit Konzepten, welche aus einer Ontologie stammen, ausgezeichnet. Bei einer Ressource kann es sich in diesem Zusammenhang beispielsweise um ein Dokument oder einen Teil eines Dokuments handeln. Im weitesten Sinn kann alles für das ein URI vergeben werden kann eine Ressource sein (vgl. Abschnitt 2.2.1).

In Ansätzen zur semantischen Annotation von textuellen Ressourcen, welche auf *Instance identification* oder *Instance reference* basieren, wie beispielsweise (Castells u. a., 2007) oder (Kiryakov u. a., 2004), erfolgt die semantischen Annotation auf einer feingranulareren Ebene: Einzelne Wörter

in textuellen Ressourcen werden mit Konzepten aus einer Ontologie annotiert.

Der Aboutness-basierende Ansatz wird aus folgenden drei Gründen verfolgt:

1. Ziel dieser Arbeit ist es einen Beitrag zur Realisierung des Semantic Webs in naher Zukunft zu leisten. Zu diesem Zweck wird ein pragmatischer Ansatz gewählt, um ein wenig Semantik (*litte semantics* (Hendler, 2007)) ins gegenwärtige Web zu bringen.
2. Auch wenn mit dem verwendeten Ansatz nicht die vollständige Bedeutung von Wörtern, welche in einer Ressource vorkommen, abgebildet werden kann, wird trotzdem zusätzliche Information zur Ressource hinzugefügt, welche für die Suche verwendet werden kann (Spärck Jones, 2004). Zusätzlich erfolgt dies mit einem deutlich geringeren menschlichen Aufwand als bei der semantischen Auszeichnung auf Wortebene.
3. Der Aboutness-basierende Ansatz erlaubt die Gleichbehandlung von unterschiedlichen Typen von Ressourcen. So können sowohl Bild-, Musik- oder Text-Ressourcen von etwas *handeln*. Ebenfalls können mit diesem Ansatz Dienste beschrieben werden. Der Instanz-basierte Ansatz zur semantischen Annotation, welcher bei textuellen Ressourcen Verwendungen findet, kann nicht einfach auf andere Typen von Ressourcen umgelegt werden.

6.3.2 Repräsentation als Netzmodell

Die Menge der sich im System befindenden Ressourcen und deren semantische Annotationen mit Konzepten werden in der Form eines Netzes repräsentiert. Das Netzmodell besteht aus zwei Schichten, einer Schicht C an Knoten, welche Konzepte repräsentieren (vgl. Gleichung 6.1) und einer Schicht R an Knoten, welche für Ressourcen stehen (vgl. Gleichung 6.2).

$$C = \{c_1, ..., c_N\} \tag{6.1}$$

$$R = \{r_1, ..., r_M\} \tag{6.2}$$

Wobei für Gleichung 6.1 und Gleichung 6.2 gilt:

- C ... Schicht an Knoten, welche Konzepte repräsentieren
- R ... Schicht an Knoten, welche Ressourcen repräsentieren
- $c_1, ..., c_N$ ... Konzeptknoten 1 bis N
- $r_1, ..., r_M$ ... Ressourceknoten 1 bis M
- N ist gleich der Anzahl der Konzepte im System
- M ist gleich der Anzahl der Ressourcen im System
- $\forall c_i \in C$ gilt : $\mathcal{C}(c_i) \in K$
- $\forall r_j \in R$ gilt : $\mathcal{R}(r_j) \in S$
- $\mathcal{C}(c_i)$ ... Konzept für das Konzeptknoten c_i steht
- $\mathcal{R}(r_j)$ ... Ressource für das Ressourceknoten r_j steht
- K ... Menge der Konzepte in der Ontologie
- S ... Menge der Ressourcen im System

Abbildung 6.1 zeigt eine grafische Repräsentation des Modells mit je einer Schicht aus Konzeptknoten und einer Schicht aus Ressourceknoten.

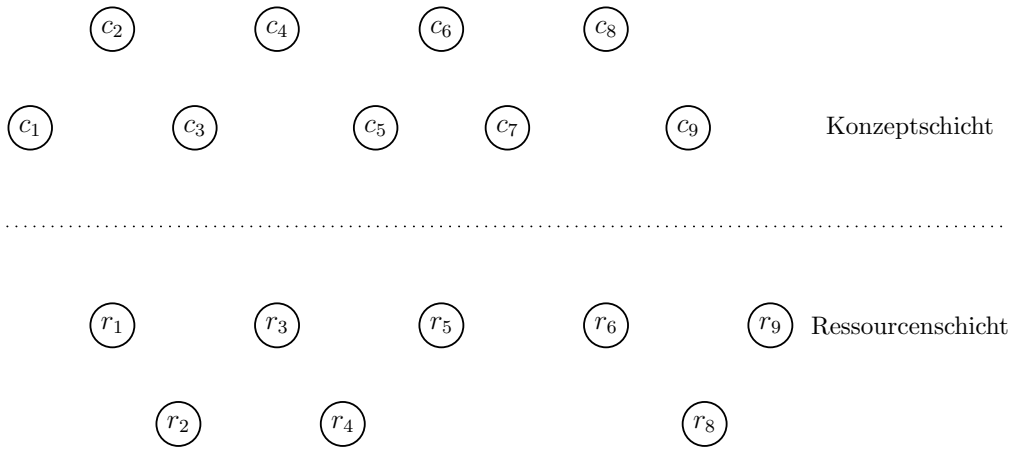

Abbildung 6.1: Das Netzmodell mit je einer Schicht für Konzepte und für Ressourcen

Für das Netz in Abbildung 6.1 ergeben sich folgende Instanzierungen von Gleichung 6.1 und Gleichung 6.2 für die Konzeptschicht und die Ressourceschicht:

- $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$
- $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_8, r_9\}$

Die Knoten der Ressourceschicht sind mit jenen Knoten der Konzeptschicht verbunden, welche die Konzepte repräsentieren mit denen die Ressourcen annotiert sind. Die Verbindungen sind von den Konzeptknoten zu den Ressourceknoten gerichtet. Dies hat den Grund, dass im hier vorgestellten Modell eine Suche nach Ressourcen ausgehend von einer Menge an Konzepten erfolgt. Um ebenfalls eine Suche nach Konzepten ausgehend von einer Menge an Ressourcen zu ermöglichen, müsste eine weitere Kante in die entgegengesetzte Richtung eingeführt werden.

Zusätzlich zur Richtung können die Verbindungen zwischen Konzeptknoten und Ressourceknoten benannt und gewichtet sein. Die Benennung der Kanten kann dazu verwendet werden, Kanten eine unterschiedliche Bedeutung zu geben. Das Gewicht kann dazu genutzt werden, die Wichtigkeit der Ressource für das Konzept widerzuspiegeln.

6.3.3 Gewichtung der Annotationen

Durch die Annotation von Ressourcen mit Konzepten aus einer Ontologie wird eine nicht gewichtete Verbindung zwischen diesen beiden Entitäten hergestellt. Auf Basis dieser Verbindungen kann durch eine Suche ausgehend von einer Menge an Konzepten ein Ressource entweder gefunden (wenn diese mit einem der Konzepte in der Anfrage verbunden ist) oder nicht gefunden (wenn die Ressource mit keinem der Konzepte aus der Suchanfrage annotiert wurde) werden. Alle Suchergebnisse zu einer Anfrage sind gleich relevant, eine Reihung der Suchergebnisse ist mit diesem Ansatz nicht möglich.

Dieser Ansatz zur Suche findet in den SQL-ähnlichen Abfragesprachen für das Semantic Web, wie RQL[2], TRIPLE[3], RDQL[4] oder der SPARQL

[2] `http://athena.ics.forth.gr:9090/RDF/RQL/index.html` (18.05.2008)
[3] `http://triple.semanticweb.org/` (18.05.2008)
[4] `http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/` (18.05.2008)

Query Language for RDF (SPARQL)[5], Verwendung. Zu einer Suchanfrage werden jene Tripel retourniert, welche der Anfrage entsprechen. Alle retournierten Ergebnisse sind relevant und vor allem gleich relevant. In Abschnitt 3.4.1 und in Castells u. a. (2007) ist eine Diskussion dieser Problematik zu finden. Auch Stojanovic u. a. (2003) identifizieren dieses Situation und argumentieren, dass auch, wenn alle Ergebnisse relevant sind, es höchst unwahrscheinlich erscheint, dass ein Computerprogramm oder die Benutzer, welche mit diesem Programm arbeiten, mit der potentiell großen Menge an Daten auf eine Anfrage an das Semanitc Web umgehen können werden. Aus diesem Grund schlagen sie einen Ansatz zur Reihung von Suchergebnissen im Semantic Web vor (vgl. Abschnitt 3.4).

Um im hier vorgestellten Modell eine Reihung der Suchergebnisse zu ermöglichen, wird den Annotationen von Ressourcen mit Konzepten ein Gewichtungsfaktor zugeordnet. Das Gewicht einer Annotation ist umso höher, je höher die Wichtigkeit einer Ressource für ein Konzept ist. Durch diesen Ansatz wird es ermöglicht, die Menge an Suchergebnissen zu reihen, während die Ergebnismenge einer Suche unverändert bleibt. Eine Suche wie in einem wissensbasierten System ohne die Gewichtung von Annotationen ist noch immer möglich, hierzu muss allen Annotationen dasselbe Gewicht zugeordnet werden.

Gleichung 6.3 zeigt das Kantengewicht w_{c_i,r_j} einer Kante von einem Konzeptknoten c_i zu einem Ressourceknoten r_i.

$$w_{c_i,r_j} = weight(c_i, r_j) \tag{6.3}$$

Wobei gilt:

- w_{c_i,r_j} ... Kantengewicht der Kante zwischen Konzeptknoten c_i und Ressourceknoten r_j
- $weight(c_i, r_j)$ ... Gewicht der Annotation, die durch die Kante zwischen Konzeptknoten c_i und Ressourceknoten r_j repräsentiert wird. Die Berechnung kann[6] wie in Gleichung 7.11 in Abschnitt 7.5.1 erfol-

[5] `http://www.w3.org/TR/rdf-sparql-query/` (18.05.2008)

[6] Während in diesem Abschnitt ein Modell zur assoziativen Suche vorgestellt wird, erfolgt in Abschnitt 7 eine Parametrisierung des Modells mit einem konkreten Gewichtungssche-

gen.

- $0 \leq weight(c_i, r_j) \leq 1$

Gleichung 6.4 zeigt die Menge aller Kanten, welche die Schicht an Konzeptknoten und die Schicht an Ressourceknoten miteinander verbinden, wobei eine Kante zwischen einem Konzeptknoten c_i zu einem Ressourceknoten r_j nur dann erzeugt wird, falls das Kantengewicht w_{c_i,r_j} dieser Kante > 0 ist. Für den Fall, dass keine Annotationen von Ressourcen mit Konzepten vorhanden sind oder dass die Gewichte aller Annotationen $= 0$ sind, besteht das Netz aus zwei nicht miteinander verbundene Schichten (vgl. Abbildung 6.1).

$$W_{C,R} = \{w_{c_i,r_j} | w_{c_i,r_j} > 0\} \tag{6.4}$$

- $W_{C,R}$ ... Menge aller Kanten zwischen C und R
- C ... Schicht an Knoten, welche Konzepte repräsentieren
- R ... Schicht an Knoten, welche Ressourcen repräsentieren
- $c_1, ..., c_N$ ... Konzeptknoten 1 bis N
- $r_1, ..., r_M$ ... Ressourceknoten 1 bis M
- N ist gleich der Anzahl der Konzepte im System
- M ist gleich der Anzahl der Ressourcen im System

Abbildung 6.2 zeigt eine grafische Repräsentation der beiden Schichten des Modells unter Berücksichtigung der Kantengewichte zwischen Konzept- und Ressourceknoten.

Für das Netz in Abbildung 6.2 ergeben sich folgende Instanzierungen von Gleichung 6.1, von Gleichung 6.2 und von Gleichung 6.4 für die Konzeptschicht, die Ressourceschicht und die Menge an Kanten zwischen den beiden Schichten:

- $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$

---

ma auf dessen Basis das entwickelte System evaluiert wird. Neben dem in Abschnitt 7 präsentierten Gewichtungsschema können zur Gewichtung der Annotationen auch andere Maße Verwendung finden, solange sie das Formalkriterium $0 \leq weight(c_i, r_j) \leq 1$ erfüllen.

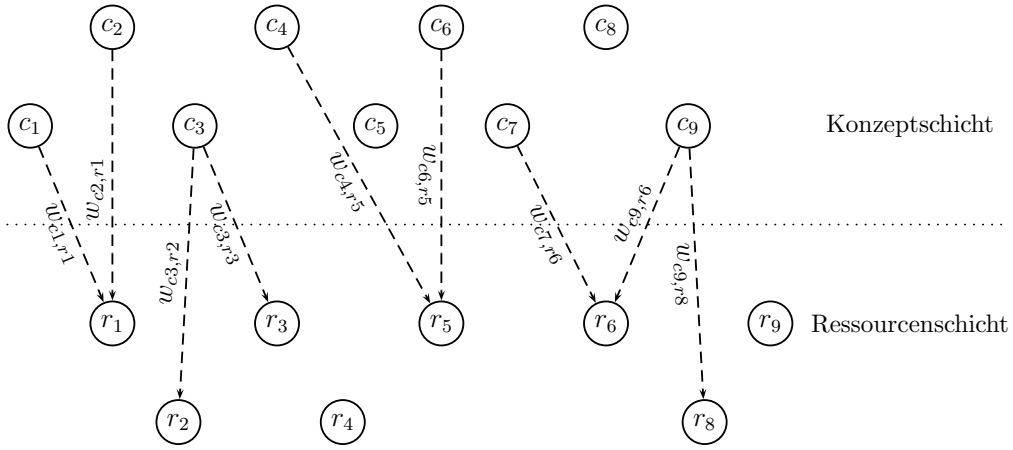

Abbildung 6.2: Das verwendete Netzmodell mit je einer Schicht für Konzepte und Ressourcen und gewichteten Verbindungen zwischen Konzept- und Ressourceknoten

- $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_8, r_9\}$
- $W_{C,R} = \{w_{c_1,r_1}, w_{c_2,r_1}, w_{c_3,r_2}, w_{c_3,r_3}, w_{c_4,r_5}, w_{c_6,r_5}, w_{c_7,r_6}, w_{c_9,r_6}, w_{c_9,r_8}\}$

6.3.4 Ähnlichkeit zwischen Konzepten

Knoten in der Konzeptschicht können untereinander verbunden sein. Diese Verbindungen sind mittels der semantischen Ähnlichkeit zwischen den beiden Konzepten, welche durch die Knoten repräsentiert werden, gewichtet (w_{c_i,c_j}, siehe Gleichung 6.5). Je höher die semantische Ähnlichkeit zwischen zwei Konzepten, desto höher das Kantengewicht. Wird ein asymmetrisches Ähnlichkeitsmaß zur Bestimmung der semantischen Ähnlichkeit verwendet, kann die Ähnlichkeit von Konzept 1 zu 2 eine andere sein kann als jene von Konzept 2 zu 1. Um diesen Fall zu unterstützen, wird die Ähnlichkeit zwischen zwei Konzepten durch zwei gerichtet Kanten repräsentiert.

Gleichung 6.5 zeigt das Kantengewicht w_{c_i,c_j} einer Kante von einem Konzeptknoten c_i zu einem Konzeptknoten c_j.

$$w_{c_i,c_j} = semsim(c_i, c_j) \tag{6.5}$$

Wobei gilt:

- w_{c_i,c_j} ... Kantengewicht der Kante zwischen Konzeptknoten c_i und Konzeptknoten c_j

- $semsim(c_i, c_j)$ ... Semantische Ähnlichkeit zwischen den Konzepten, welche durch die Konzeptknoten c_i und c_j repräsentiert werden. Die Berechnung kann[7] wie in Gleichung 7.12, Gleichung 7.13 oder Gleichung 7.15 in Abschnitt 7.5.2 erfolgen.
- $0 \leq semsim(c_i, c_j) \leq 1$

Abhängig von den Charakteristika, welche eine Ontologie aufweist, können unterschiedliche Maße zur Berechnung der Ähnlichkeit zwischen zwei Konzepten Verwendung finden (vgl. Abschnitt 7.5.2).

Gleichung 6.6 zeigt die Menge aller Kanten, welche Konzeptknoten mit Konzeptknoten verbinden, wobei eine Kante zwischen einem Konzeptknoten c_i zu einem anderen Konzeptknoten c_j nur dann erzeugt wird, falls das Kantengewicht w_{c_i,c_j} dieser Kante > 0 ist.

$$W_{C,C} = \{w_{c_i,c_j} | w_{c_i,c_j} > 0\} \tag{6.6}$$

- $W_{C,C}$ ... Menge aller Kanten zwischen Konzepten in C
- C ... Schicht an Knoten, welche Konzepte repräsentieren
- $c_1, ..., c_N$ ... Konzeptknoten 1 bis N
- N ist gleich der Anzahl der Konzepte im System

6.3.5 Ähnlichkeit zwischen Ressourcen

Auch die Knoten der Ressourceschicht können untereinander verbunden sein. In diesem Fall wird die inhaltsbasierte Ähnlichkeit als Gewichtungsfunktion für die Kanten zwischen zwei Ressourceknoten herangezogen (w_{r_i,r_j}, siehe Gleichung 6.7). Je höher die inhaltsbasierte Ähnlichkeit zwischen zwei Ressourcen, desto höher das Kantengewicht. Auch hier wird die Ähnlichkeit zwischen zwei Ressourcen über zwei Kanten repräsentiert um,

[7] Während in diesem Abschnitt ein Modell zur assoziativen Suche vorgestellt wird, erfolgt in Abschnitt 7 eine Parametrisierung des Modells mit konkreten Ähnlichkeitsmaßen auf deren Basis das entwickelte System evaluiert wird. Neben den in Abschnitt 7 präsentierten Ähnlichkeitsmaßen können zur Berechnung der semantische Ähnlichkeit auch andere Maße Verwendung finden, solange sie das Formalkriterium $0 \leq semsim(c_i, c_j) \leq 1$ erfüllen.

so wie bei der semantischen Ähnlichkeit, asymmetrische Ähnlichkeitsmaße zu unterstützen.

Gleichung 6.7 zeigt das Kantengewicht w_{r_i,r_j} einer Kante von einem Ressourceknoten r_i zu einem Ressourceknoten r_j.

$$w_{r_i,r_j} = contsim(r_i, r_j) \tag{6.7}$$

Wobei gilt:

- w_{r_i,r_j} ... Kantengewicht der Kante zwischen Ressourceknoten r_i und Ressourceknoten r_j
- $contsim(r_i, r_j)$ ... Inhaltsbasierende Ähnlichkeit zwischen den Ressourcen, welche durch die Ressourceknoten r_i und r_j repräsentiert werden. Die Berechnung kann[8] wie in Gleichung 7.16 in Abschnitt 7.5.3 erfolgen.
- $0 \leq contsim(r_i, r_j) \leq 1$

Gleichung 6.8 zeigt die Menge aller Kanten, welche Ressourceknoten mit Ressourceknoten verbinden, wobei eine Kante zwischen einem Ressourceknoten r_i zu einem anderen Ressourceknoten r_j nur dann erzeugt wird, falls das Kantengewicht w_{r_i,r_j} dieser Kante > 0 ist.

$$W_{R,R} = \{w_{r_i,r_j} | w_{r_i,r_j} > 0\} \tag{6.8}$$

- $W_{R,R}$ ... Menge aller Kanten zwischen Ressourcen in R
- R ... Schicht an Knoten, welche Ressourcen repräsentieren
- $r_1, ..., r_M$ ... Ressourceknoten 1 bis M
- M ist gleich der Anzahl der Ressourcen im System

[8] Während in diesem Abschnitt ein Modell zur assoziativen Suche vorgestellt wird, erfolgt in Abschnitt 7 eine Parametrisierung des Modells mit einem konkreten Ähnlichkeitsmaß auf dessen Basis das entwickelte System evaluiert wird. Neben dem in Abschnitt 7 präsentierten Ähnlichkeitsmaß können zur Berechnung der inhaltsbasierte Ähnlichkeit auch andere Maße Verwendung finden, solange sie das Formalkriterium $0 \leq contsim(r_i, r_j) \leq 1$ erfüllen.

Abbildung 6.3 zeigt eine grafische Repräsentation der beiden Schichten des Modells unter Berücksichtigung semantischer und inhaltsbasierter Ähnlichkeit.

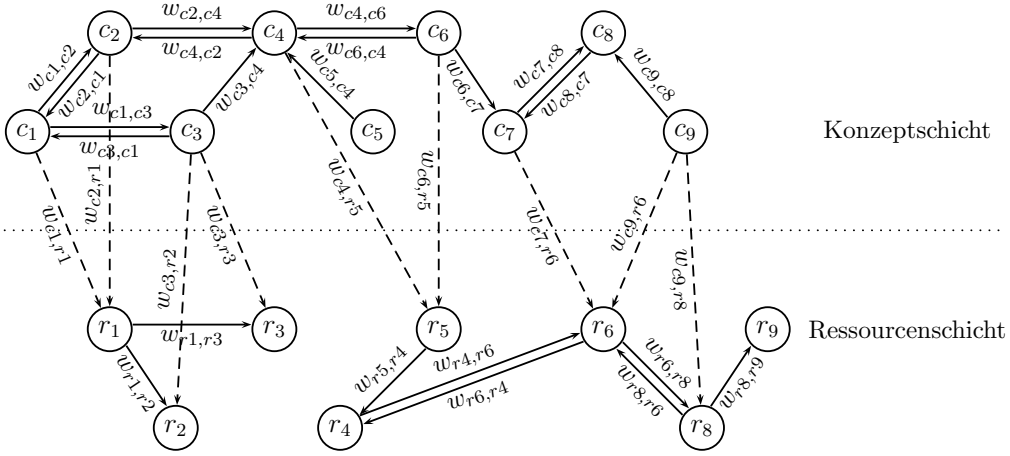

Abbildung 6.3: Das Netzmodell unter Berücksichtigung semantischer und inhaltsbasierter Ähnlichkeit

Für das Netz in Abbildung 6.3 ergeben sich folgende Instanzierungen der Gleichung 6.1, 6.2, 6.4, 6.6 und 6.8 für die Konzeptschicht, die Ressourceschicht, die Menge an Kanten zwischen den beiden Schichten, die Menge an Kanten zwischen Konzeptknoten und die Menge an Kanten zwischen Ressourceknoten:

- $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$
- $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_8, r_9\}$
- $W_{C,R} = \{w_{c_1,r_1}, w_{c_2,r_1}, w_{c_3,r_2}, w_{c_3,r_3}, w_{c_4,r_5}, w_{c_6,r_5}, w_{c_7,r_6}, w_{c_9,r_6}, w_{c_9,r_8}\}$
- $W_{C,C} = \{w_{c1,c2}, w_{c1,c3}, w_{c2,c1}, w_{c2,c4}, w_{c3,c1}, w_{c4,c3}, w_{c4,c2}, w_{c4,c6},$
  $w_{c5,c4}, w_{c6,c4}, w_{c6,c7}, w_{c7,c8}, w_{c8,c7}, w_{c9,c8}\}$
- $W_{R,R} = \{w_{r1,r2}, w_{r1,r3}, w_{r4,r6}, w_{r5,r4}, w_{r6,r4}, w_{r6,r8}, w_{r8,r6}, w_{r8,r9}\}$

6.3.6 Suche im Netz

Die Netzstruktur, die dem hier vorgestellten Modell zu Grunde liegt wird mittels eines Spreading-Activation-Ansatzes durchsucht. Ausgehend von einer Menge an initial aktivierten Knoten im Netz, breitet sich Aktivierung

über die Kanten im Netz aus und aktiviert Knoten, die mit den initial aktivierten Knoten verbunden sind.

Spreading Activation nahm seinen Ursprung im Gebiet der Wahrnehmungspsychologie, wo es als Modell für die Ausbreitung von Gedanken im menschlichen Gehirn diente (Anderson, 1983). Über Anwendungen in Neuronalen und Semantischen Netzen (vgl. Abschnitt 4.4.5) fand es schließlich Einzug in das Feld des Information Retrieval (Crestani, 1997a). In ihrer Leistungsfähigkeit sind Spreading-Activation-Modelle mit anderen Ansätzen wie dem Vektorraummodell vergleichbar (Mandl, 2001). Eine detaillierte Einführung in Spreading Activation ist in den Kapiteln 4.4 und 5 zu finden.

Neben Systemen, welche Spreading Activation zur Identifikation von Ähnlichkeiten zwischen Textdokumenten oder zwischen Suchtermen und Textdokumenten einsetzen (vgl. (Wilkinson u. Hingston, 1991) oder (Mothe, 1994) in Abschnitt 5), existieren auch Systeme, welche Spreading Activation dazu verwenden ähnliche Konzepte in Wissensrepräsentationen zu finden (vgl. (Cohen u. Kjeldsen, 1987), (Alani u. a., 2003) oder (Rocha u. a., 2004a) in Abschnitt 4.5). Der hier vorgestellte Ansatz kombiniert Spreading-Activation-Suche in einer Ressourcensammlung mit Spreading-Activation-Suche in einer Wissensrepräsentation.

Gleichung 6.9 zeigt wie die Aktivierungsausbreitung im Netz berechnet wird. Die Aktivierung eines Knotens n_j wird aus der Summe der Aktivierungen jener Knoten n_i berechnet von denen er Aktivierung erhält. Falls ein Knoten von keinem anderen Knoten Aktivierung erhält, ist seine Aktivierung = 0. Der vorgestellte Ansatz beruht auf Aktivierungserhaltung, das heißt jeder Knoten kann nur soviel Aktivierung abgeben wie er aufnimmt. Hierdurch wird ein Ansteigen der Aktivierung im Netz nach mehreren Schritten der Aktivierungsausbreitung verhindert (vgl. Abschnitt 5.4). Um Aktivierungserhaltung zu erzielen, muss die Summe aller Kantengewichte der Kanten über die ein Knoten Aktivierung abgibt $= 1,0$ sein. Um dies zu erreichen, wird das Kantengewicht jener Kanten über die ein Knoten n_i Aktivierung abgibt, durch die Summe der Kantengewichte dieser Kanten dividiert. Die Ausgangsaktivierung jedes Knoten n_i wird somit in Summe mit $1,0$ multipliziert, der Knoten n_i gibt exakt soviel Aktivierung ab wie er besitzt.

$$\mathcal{A}(n_j) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(n_i) \cdot w_{i,j}}{\sum_{k=1}^{s} w_{i,k}} & \text{für } t > 0 \\ 0 & \text{für } t = 0 \end{cases} \tag{6.9}$$

Wobei gilt:

- $\mathcal{A}(n_j)$ ... Aktivierung von Knoten n_j
- $\mathcal{A}(n_i)$ ... Aktivierung von Knoten n_i
- t ... Anzahl der Knoten von denen Knoten n_j Aktivierung erhält
- $w_{i,j}$ ... Gewicht der Kanten zwischen Knoten n_i und Knoten n_j
- s ... Anzahl der Knoten an die Knoten n_i Aktivierung abgibt [9]
- $w_{i,k}$ ... Gewicht der Kanten zwischen Knoten n_i und Knoten n_k

Beispiel für die Ausbreitung von Aktivierung

Abbildung 6.4 zeigt eine Netzstruktur anhand derer im Folgenden ein Beispiel für die Ausbreitung von Aktivierung nach Gleichung 6.9 gegeben wird.

Es wird nun die Aktivierung von Knoten n_5 berechnet. Für Knoten n_5 ist $t = 4$ da dieser Knoten von 4 anderen Knoten (n_1, n_2, n_3, n_4) Aktivierung erhält. Somit gilt für die Berechnung der Aktivierung von Knoten n_5 der obere Fall in Gleichung 6.9. Dieser wird in Gleichung 6.10 angeführt.

$$\mathcal{A}(n_5) = \frac{\mathcal{A}(n_1) \cdot w_{1,5}}{w_{1,5} + w_{1,6} + w_{1,7}} + \frac{\mathcal{A}(n_2) \cdot w_{2,5}}{w_{2,5} + w_{2,7}} + \frac{\mathcal{A}(n_3) \cdot w_{3,5}}{w_{3,5} + w_{3,8}} + \frac{\mathcal{A}(n_4) \cdot w_{4,5}}{w_{4,5}} \tag{6.10}$$

Zur weiteren Berechnung werden nun Kantengewichte festgelegt:

- $w_{1,5} = 0,4$
- $w_{1,6} = 0,2$
- $w_{1,7} = 0,3$
- $w_{2,5} = 0,4$
- $w_{2,7} = 0,5$

[9] $s > 0$ wenn $t > 0$ da n_i mit n_j verbunden ist. Somit gibt n_i zumindest an einen Knoten Aktivierung ab, wenn n_j von einem Knoten Aktivierung erhält

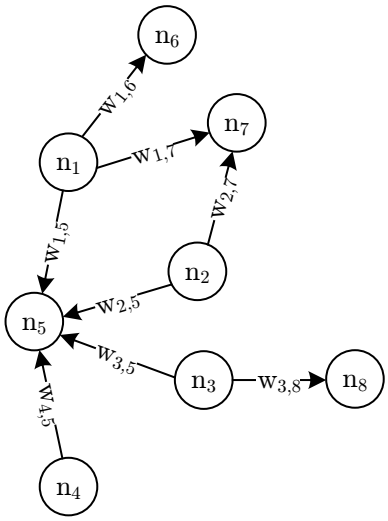

Abbildung 6.4: Beispielhafte Netzstruktur für Berechnung der Aktivierungsausbreitung

- $w_{3,5} = 0,7$
- $w_{3,8} = 0,1$
- $w_{4,5} = 0,3$

Die Aktivierungen der Knoten betragen derzeit:

- $\mathcal{A}(n_1) = 0,7$
- $\mathcal{A}(n_2) = 0,5$
- $\mathcal{A}(n_3) = 0,6$
- $\mathcal{A}(n_4) = 0,2$

Unter verwenden der oben angeführten Kantengewichte und Aktivierungen der Knoten wird aus Gleichung 6.10 Gleichung 6.11.

$$\begin{aligned} \mathcal{A}(n_5) &= \frac{0,7 \cdot 0,4}{0,4+0,2+0,3} + \frac{0,5 \cdot 0,4}{0,4+0,5} + \frac{0,6 \cdot 0,7}{0,7+0,1} + \frac{0,2 \cdot 0,3}{0,3} \\ &= \frac{0,28}{0,9} + \frac{0,2}{0,9} + \frac{0,42}{0,8} + \frac{0,06}{0,3} \approx 1,26 \end{aligned} \tag{6.11}$$

Die Aktivierung von Knoten n_5 beträgt nach der Aktivierungsausbreitung $\approx 1,26$.

Grundlegende Suchfunktionalität

Ziel der Suche im Netz ist es eine Menge an Ressourcen zu einer Anfrage, welche aus einer Menge an Konzepten besteht, zu finden und in weiterer Folge die Ergebnismenge nach Relevanz für die Anfrage zu reihen. Zu diesem Zweck breitet sich Aktivierung von einer Menge an initial aktivierten Konzeptknoten zu jenen Ressourceknoten aus, welche mit diesen Konzeptknoten verbunden sind.

Eine Suche im Netz läuft grundsätzlich wie folgt ab:

1. Die Suche startet von einer Menge an Konzepten, welche das Informationsbedürfnis des Suchenden repräsentieren. Jene Knoten, welche diese Konzepte repräsentieren, werden aktiviert.

2. Aktivierung breitet sich von der Menge an gegenwärtig aktivierten Konzeptknoten zu Ressourceknoten aus. Hierzu wird die Aktivierungsfunktion aus Gleichung 6.14 verwendet. Die Aktivierungsausbreitung erfolgt über jene Kanten, welche durch die semantische Annotation von Ressourcen (vgl. Abschnitt 6.3.3) mit Konzepten erzeugt wurden. Somit werden Ressourceknoten aktiviert, welche von jenen Konzepten handeln, die das Informationsbedürfnis des Suchenden repräsentieren.

3. Jene Ressourcen, welche zur Menge an schließlich aktivierten Ressourceknoten korrespondieren, werden als Suchergebnis an den Benutzer zurückgeliefert. Aufgrund des Aktivierungswertes der Knoten wird die Ergebnismenge der Ressourcen gereiht.

Abbildung 6.5 zeigt den Ablauf der grundlegenden Suche als Aktivitätsdiagramm.

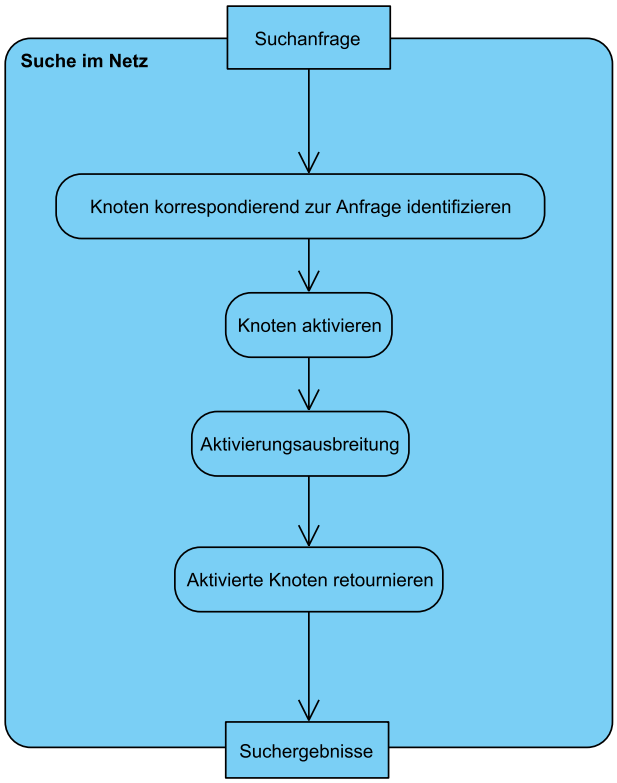

Abbildung 6.5: Grundlegender Ablauf der Suche im Assoziativen Netz

Eine Suche im Assoziativen Netz kann als eine Funktion σ beschrieben werden, welche die Menge $\overline{A}$ aller Anfragen an das System, in die Menge $\overline{E}$ aller Ergebnisse des Systems transformiert. $\overline{A}$ ist eine Menge an Anfragen, eine Anfrage A ist eine Menge an Konzepten aus K. $\overline{E}$ ist eine Menge an Ergebnissen, ein Ergebnis E ist eine Menge an Ressourcen aus S. Formel 6.12 zeigt diesen Zusammenhang.

$$\sigma : \overline{A} \rightarrow \overline{E} \tag{6.12}$$

Wobei gilt:

- $\overline{A}$ ... Menge aller Anfragen an das System
- $\overline{A} = \{\ A \mid A$ ist eine Anfrage an das System und $A \subseteq K\ \}$
- A ... Anfrage an das System
- K ... Menge der Konzepte in der Ontologie

- $\overline{E}$ ... Menge aller Ergebnisse des Systems
- $\overline{E} = \{\ E \mid E$ ist ein Ergebnis einer Suche und $E \subseteq S\ \}$
- E ... Ergebnis einer Suche
- S ... Menge der Ressourcen im System

Initiale Aktivierung

Um die Suche zu initiieren, werden jene Konzeptknoten aktiviert, welche Konzepte repräsentieren, die in der Anfrage vertreten sind. Zu diesem Zweck wird diesen Konzeptknoten initial ein fixer Aktivierungswert zugewiesen. Dieser initiale Aktivierungsgrad der Konzeptknoten beträgt $1,0$ durch die Anzahl der Konzepte in der Anfrage. Die Höhe der initialen Aktivierung eines Konzeptknotens c_i wird nach Gleichung 6.13 berechnet. Konzeptknoten zu Konzepten, die nicht in der Anfrage vorkommen, wird eine Aktivierung von 0 zugewiesen. Somit wird bei einer leeren Anfrage allen Konzeptknoten eine Aktivierung von 0 zugewiesen.

$$\forall c_i \in C \text{ gilt} : \mathcal{A}(c_i) = \begin{cases} \frac{1,0}{|A|} & \text{für } \mathcal{C}(c_i) \in |A| \\ 0 & \text{für } \mathcal{C}(c_i) \notin |A| \end{cases} \tag{6.13}$$

Wobei gilt:

- $\mathcal{A}(c_i)$ ... Aktivierung von Konzeptknoten c_i
- $|A|$ ... Anzahl der Konzepte in der Anfrage
- $\mathcal{C}(c_i)$ ... Konzept für das Konzeptknoten c_i steht
- C ... Schicht an Knoten, welche Konzepte repräsentieren

Abbildung 6.6 zeigt eine grafische Repräsentation des Modells mit einer initialen Aktivierung jener Knoten im Netz, welche zu den Konzepten in der Anfrage korrespondieren.

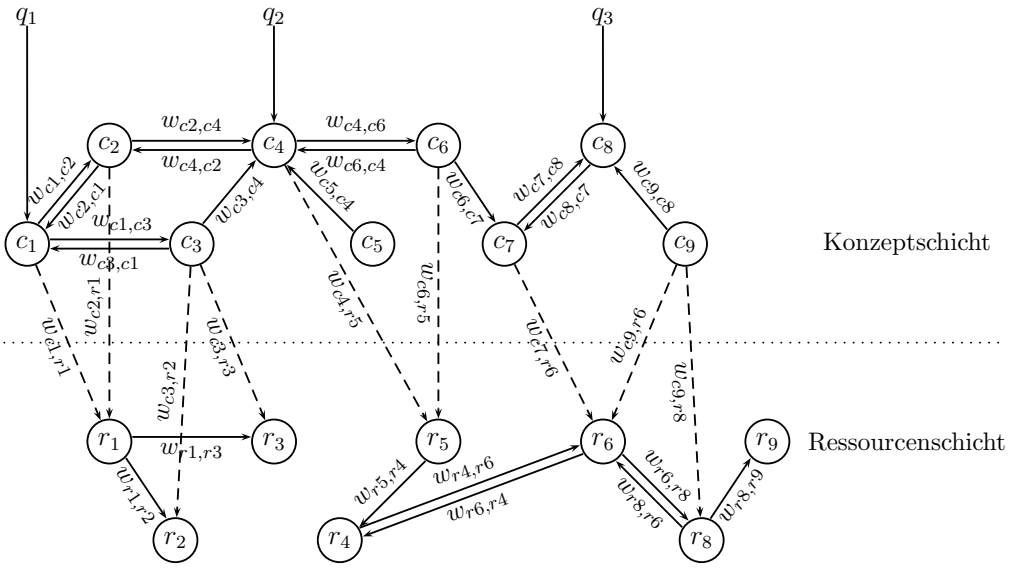

Abbildung 6.6: Initiale Aktivierung jener Knoten im Netz, welche zu den Konzepten in der Anfrage korrespondieren

Beispiel für die Berechnung der initialen Aktivierung

Die initiale Aktivierung der Konzeptknoten soll anhand eines Beispiels verdeutlicht werden. Korrespondierend zu Abbildung 6.6 ist folgendes gegeben:

- eine Anfrage $A = \{q_1, q_2, q_3\}$
- und eine Konzeptschicht $C = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$

Für die Knoten der Konzeptschicht gilt:

- $\mathcal{C}(c_1) = q_1$
- $\mathcal{C}(c_4) = q_2$
- $\mathcal{C}(c_8) = q_3$

Nach Gleichung 6.13 erhalten die Knoten der Konzeptschicht die folgenden initialen Aktivierungen:

- $\mathcal{A}(c_1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_2) = 0$
- $\mathcal{A}(c_3) = 0$

- $\mathcal{A}(c_4) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_5) = 0$
- $\mathcal{A}(c_6) = 0$
- $\mathcal{A}(c_7) = 0$
- $\mathcal{A}(c_8) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_9) = 0$

Ausbreitung der Aktivierung

Nach der initialen Aktivierung der Knoten der Konzeptschicht werden im nächsten Schritt die Knoten der Ressourceschicht aktiviert. Zu diesem Zweck breitet sich Aktivierung von Konzeptknoten zu Ressourceknoten aus. Der allgemeine Fall der Aktivierungsausbreitung wird in Abschnitt 6.3.6 beschrieben. Gleichung 6.14 zeigt die Weitergabe von Aktivierung von einem Konzeptknoten zu einem Ressourceknoten. Hierbei handelt es sich um einen Spezialfall von Gleichung 6.9 in Abschnitt 6.3.6.

Die Aktivierung eines Ressourceknoten r_j wird aus der Summe der Aktivierungen jener Knoten c_i berechnet, von denen er Aktivierung erhält. Falls der Ressourceknoten von keinem anderen Knoten Aktivierung erhält, ist die Aktivierung des Ressourceknoten = 0. Für den Fall, dass nach Gleichung 6.3 keine Kanten mit einem Gewicht $w_{c_i,r_j} > 0$ existieren, werden die Konzeptknoten zu einer Anfrage zwar aktiviert, allerdings breitet sich keine Aktivierung von den Konzept- zu den Ressourceknoten aus, da nach Gleichung 6.4 keine Kanten in das Netz eingefügt wurden.

$$\forall r_j \in R \text{ gilt} : \mathcal{A}(r_j) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(c_i) \cdot w_{c_i,r_j}}{\sum_{k=1}^{s} w_{c_i,r_k}} & \text{für } t > 0 \\ 0 & \text{für } t = 0 \end{cases} \tag{6.14}$$

Wobei gilt:

- $\mathcal{A}(r_j)$ ... Aktivierung von Ressourceknoten r_j
- $\mathcal{A}(c_i)$ ... Aktivierung von Konzeptknoten c_i
- t ... Anzahl der Konzeptknoten von denen Knoten r_j Aktivierung erhält
- w_{c_i,r_j} ... Gewicht der Kanten zwischen Knoten c_i und Knoten r_j

- s ... Anzahl der Ressourceknoten an die Knoten c_i Aktivierung abgibt[10]
- w_{c_i,r_k} ... Gewicht der Kanten zwischen Knoten c_i und Knoten r_k
- R ... Schicht an Knoten, welche Ressourcen repräsentieren

Beispiel für die Berechnung der Aktivierungsausbreitung

Die Ausbreitung der Aktivierung nach Gleichung 6.14 soll nun anhand eines Beispiels verdeutlicht werden. Hierzu wird die Aktivierung von Knoten r_5 aus Abbildung 6.6 berechnet.

Aus dem zuvor angeführten Beispiel werden die folgenden initialen Aktivierungen der Konzeptschicht übernommen:

- $\mathcal{A}(c_1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_2) = 0$
- $\mathcal{A}(c_3) = 0$
- $\mathcal{A}(c_4) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_5) = 0$
- $\mathcal{A}(c_6) = 0$
- $\mathcal{A}(c_7) = 0$
- $\mathcal{A}(c_8) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_9) = 0$

Für den Knoten r_5 ist $t = 2$ da dieser Knoten von 2 anderen Knoten (c_4, c_6) Aktivierung erhält. Somit gilt für die Berechnung der Aktivierung von Knoten r_5 der obere Fall in Gleichung 6.14. Dieser wird in Gleichung 6.15 angeführt.

$$\mathcal{A}(r_5) = \frac{\mathcal{A}(c_4) \cdot w_{c_4,r_5}}{w_{c_4,r_5}} + \frac{\mathcal{A}(c_6) \cdot w_{c_6,r_5}}{w_{c_6,r_5}} = 0,33 + 0 = 0,33 \tag{6.15}$$

Die Aktivierung von Knoten r_5 beträgt nach der Aktivierungsausbreitung $0,33$.

[10] $s > 0$ wenn $t > 0$ da c_i mit r_j verbunden ist. Somit gibt c_i zumindest an einen Knoten Aktivierung ab, wenn r_j von einem Knoten Aktivierung erhält

Zeitlicher Verlauf der Aktivierungsausbreitung

Bisher wurde die Ausbreitung von Aktivierung immer als eine Momentaufnahme betrachtet. Es wurde beschrieben wir die Aktivierung eines Knotens berechnet werden kann, allerdings nicht zu welchem Zeitpunkt das der Fall ist. Im Folgenden wird der zeitliche Verlauf der Aktivierungsausbreitung als Schritte im Prozess der Aktivierungsausbreitung in Betracht gezogen. Hierzu wird (korrespondierend zu (Mothe, 1994), vgl. Abschnitt 5.3) der Parameter τ eingeführt. Dieser steht für den aktuellen Schritt der Aktivierungsausbreitung.

Die Berechnung der Aktivierung für eine Menge an Ressourceknoten in Abhängigkeit von einer Menge an Konzeptknoten bleibt weitgehend ident zu der in Gleichung 6.14 angeführten Vorgehensweise, wird allerdings um eine Variable für den aktuellen Schritt der Aktivierungsausbreitung erweitert, um den zeitlicher Verlauf mit in Betracht ziehen zu können.

Die Aktivierung eines Ressourceknoten $\mathcal{A}(r_j, \tau+1)$ im Schritt $\tau+1$ wird in Abhängigkeit der Aktivierung $\mathcal{A}(c_i, \tau)$ jener Knoten von denen er Aktivierung erhält im Schritt τ und dem Gewicht der Verbindungen w_{c_i,r_j} zu diesen Knoten berechnet. Gleichung 6.16 repräsentiert diesen Zusammenhang durch die Funktion α.

$$\mathcal{A}(r_j, \tau+1) = \alpha(\mathcal{A}(c_i, \tau), w_{c_i,r_j}) \tag{6.16}$$

Unter Einbeziehung von Gleichung 6.9 für die allgemeine Aktivierungsausbreitung folgt daraus im Speziellen Gleichung 6.17:

$$\forall r_j \in R \text{ gilt} : \mathcal{A}(r_j, \tau+1) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(c_i,\tau) \cdot w_{c_i,r_j}}{\sum_{k=1}^{s} w_{c_i,r_k}} & \text{für } t > 0 \\ 0 & \text{für } t = 0 \end{cases} \tag{6.17}$$

Wobei gilt:

- $\mathcal{A}(r_j, \tau)$ ... Aktivierung des Ressourceknotens zu Ressource j in Schritt τ
- $\mathcal{A}(c_i, \tau)$ ... Aktivierung des Konzeptknotens zu Konzept i in Schritt τ
- τ ... Schritt der Aktivierungsausbreitung, $\tau \in \mathbb{N}_0$

- t ... Anzahl der Konzeptknoten von denen Knoten r_j Aktivierung erhält
- w_{c_i,r_j} ... Gewicht der Verbindung zwischen einem Konzeptknoten c_i und einem Ressourceknoten r_j
- $w_{c_i,r_j} = weight(c_i, r_j)$ (vgl. Gleichung 6.3)
- s ... Anzahl der Ressourceknoten an die Knoten c_i Aktivierung abgibt[11]
- w_{c_i,r_k} ... Gewicht der Verbindung zwischen einem Konzeptknoten c_i und einem Ressourceknoten r_k
- $w_{c_i,r_k} = weight(c_i, r_k)$ (vgl. Gleichung 6.3)
- R ... Schicht an Knoten, welche Ressourcen repräsentieren

Wie in Gleichung 6.14 wird die Aktivierung eines Ressourceknoten r_j aus der Summe der Aktivierungen jener Knoten c_i berechnet, von denen er Aktivierung erhält. Falls der Ressourceknoten von keinem anderen Knoten Aktivierung erhält, ist die Aktivierung des Ressourceknoten = 0. Für den Fall, dass nach Gleichung 6.3 keine Kanten mit einem Gewicht $w_{c_i,r_j} > 0$ existieren, werden die Konzeptknoten zu einer Anfrage zwar aktiviert, allerdings breitet sich keine Aktivierung von den Konzept- zu den Ressourceknoten aus, da nach Gleichung 6.4 keine Kanten in das Netz eingefügt wurden.

Beispiel zur Berechnung der Aktivierungsausbreitung unter Berücksichtigung des zeitlicher Verlaufs

Die Ausbreitung von Aktivierung nach Gleichung 6.17 soll nun anhand eines Beispiels verdeutlicht werden. Hierzu wird wie im vorherigen Beispiel die Aktivierung von Knoten r_5 aus Abbildung 6.6 berechnet.

Die Knoten der Konzeptschicht besitzen die folgenden initialen Aktivierungen:

- $\mathcal{A}(c_1, 1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_2, 1) = 0$

[11] $s > 0$ wenn $t > 0$ da c_i mit r_j verbunden ist. Somit gibt c_i zumindest an einen Knoten Aktivierung ab, wenn r_j von einem Knoten Aktivierung erhält

- $\mathcal{A}(c_3, 1) = 0$
- $\mathcal{A}(c_4, 1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_5, 1) = 0$
- $\mathcal{A}(c_6, 1) = 0$
- $\mathcal{A}(c_7, 1) = 0$
- $\mathcal{A}(c_8, 1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_9, 1) = 0$

Für den Knoten r_5 ist $t = 2$ da dieser Knoten von 2 anderen Knoten (c_4, c_6) Aktivierung erhält. Somit gilt für die Berechnung der Aktivierung von Knoten r_5 der obere Fall in Gleichung 6.17. Dieser wird in Gleichung 6.18 angeführt.

$$\mathcal{A}(r_5, 2) = \frac{\mathcal{A}(c_4, 1) \cdot w_{c_4,r_5}}{w_{c_4,r_5}} + \frac{\mathcal{A}(c_6, 1) \cdot w_{c_6,r_5}}{w_{c_6,r_5}} = 0,33 + 0 = 0,33 \quad (6.18)$$

Die Aktivierung von Knoten r_5 beträgt für Schritt $\tau = 2$ der Aktivierungsausbreitung $0,33$.

Erweiterte Suchfunktionalität (Berücksichtigung von semantischer und inhaltsbasierter Ähnlichkeit)

Zusätzlich zur zuvor beschriebenen grundlegenden Suche besteht die Möglichkeit Assoziationen zwischen Konzepten und Konzepten oder Assoziationen zwischen Ressourcen und Ressourcen in den Suchprozess mit einzubeziehen. Auch beide Fälle sind möglich. Somit ergeben sich vier verschiedene Arten der Suche im Assoziativen Netz:

1. Grundlegenden Suche auf Basis der Annotationen
2. Suche auf Basis der Annotationen und unter Einbeziehung der Assoziationen zwischen Konzepten
3. Suche auf Basis der Annotationen und unter Einbeziehung der Assoziationen zwischen Ressourcen
4. Suche auf Basis der Annotationen, unter Einbeziehung der Assoziationen zwischen Konzepten und der Assoziationen zwischen Ressourcen

Eine Suche unter Berücksichtigung der oben beschriebenen zusätzlichen Suchpfade läuft wie folgt ab:

1. Die Suche startet von einer Menge an Konzepten, welche das Informationsbedürfnis des Suchenden repräsentieren. Jene Knoten, welche diese Konzepte repräsentieren, werden aktiviert.

[2. *Optional*, breitet sich Aktivierung von der Menge an initial aktivierten Konzeptknoten, über jene Kanten, welche aufgrund semantischer Ähnlichkeit (vgl. Abschnitt 6.3.4) erzeugt wurden, zu anderen Konzeptknoten aus.]

3. Aktivierung breitet sich von der Menge an gegenwärtig aktivierten Konzeptknoten zu Ressourceknoten aus. Hierzu wird die Aktivierungsfunktion aus Gleichung 6.14 verwendet. Die Aktivierungsausbreitung erfolgt über jene Kanten, welche durch die semantische Annotation von Ressourcen (vgl. Abschnitt 6.3.3) mit Konzepten erzeugt wurden. Somit werden Ressourceknoten aktiviert, welche von jenen Konzepten handeln, die das Informationsbedürfnis des Suchenden repräsentieren.

[4. *Optional*, breitet sich Aktivierung von den gegenwärtig aktivierten Ressourceknoten zu Ressourceknoten aus, welche ähnliche Ressourcen repräsentieren. Dies geschieht über jene Kanten, welche aufgrund inhaltsbasierter Ähnlichkeit (vgl. Abschnitt 6.3.5) erzeugt wurden.]

5. Jene Ressourcen, welche zur Menge an schließlich aktivierten Ressourceknoten korrespondieren, werden als Suchergebnis an den Benutzer zurückgeliefert. Aufgrund des Aktivierungswertes der Knoten wird die Ergebnismenge der Ressourcen gereiht.

Abbildung 6.7 zeigt den Ablauf der erweiterten Suche als Aktivitätsdiagramm.

Das Kantengewicht w_{c_i,r_j}, welches in den im vorherigen Abschnitt angeführten Gleichungen Verwendung findet, gilt für die Ausbreitung der Aktivierung von Konzeptknoten zu Ressourceknoten. Für die Ausbreitung der Aktivierung von Konzeptknoten zu Konzeptknoten bzw. von Ressourceknoten zu Ressourceknoten finden die in den Abschnitten 6.3.4 und 6.3.5 angeführten Kantengewichte w_{c_i,c_j} und w_{r_i,r_j} Verwendung.

Wie bei der Ausbreitung von Aktivierung von Konzeptknoten zu Ressourceknoten ist bei der Ausbreitung der Aktivierung von Konzeptknoten

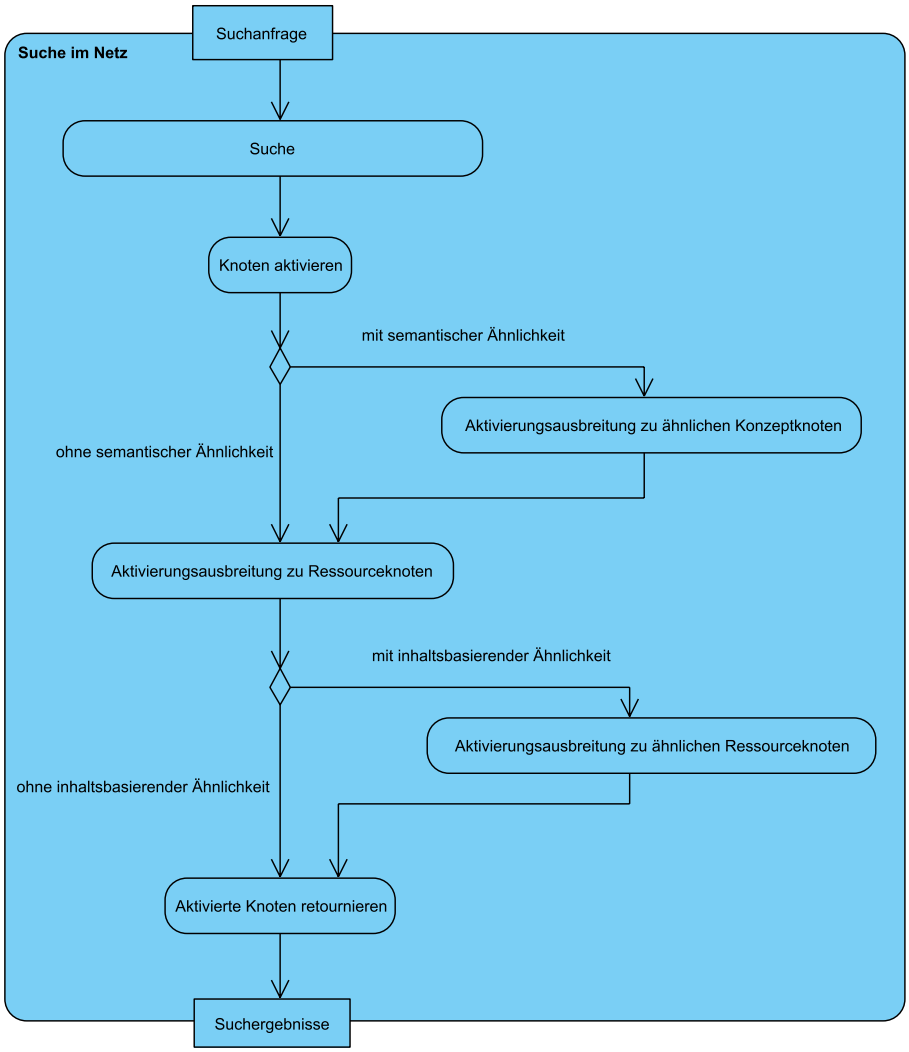

Abbildung 6.7: Ablauf der Suche im Assoziativen Netz mit optionalen Schritten

zu Konzeptknoten bzw. von Ressourceknoten zu Ressourceknoten die Aktivierung $\mathcal{A}(c_j, \tau+1)$ bzw. $\mathcal{A}(r_j, \tau+1)$ eines Knoten im Schritt $\tau+1$ von der Aktivierung $\mathcal{A}(c_i, \tau)$ bzw. $\mathcal{A}(r_i, \tau)$ jener Knoten von denen er Aktivierung erhält im Schritt τ und dem Gewicht der Verbindungen w_{c_i,c_j} bzw. w_{r_i,r_j} zu diesen Knoten abhängig. Zusätzlich wird allerdings noch die Aktivierung $\mathcal{A}(c_j, \tau)$ bzw. $\mathcal{A}(r_j, \tau)$, die der Knoten im Schritt τ hatte, in Betracht gezogen. Dies bedeutet, dass bei dieser Art der Aktivierungsausbreitung die Aktivierung, die bereits in Knoten vorhanden ist, nicht durch die Sum-

me der Aktivierungen, die von den Nachbarknoten empfangen wird, ersetzt wird. Es werden somit zusätzliche Knoten in der Konzeptschicht und in der Ressourceschicht aktiviert, während die bereits aktivierten Knoten aktiviert bleiben bzw. zusätzliche Aktivierung von ihren Nachbarn erhalten.

Dieser Zusammenhang wird in Gleichung 6.19 und Gleichung 6.20 durch die Funktion β präsentiert.

$$\mathcal{A}(c_j, \tau + 1) = \beta(\mathcal{A}(c_j, \tau), \mathcal{A}(c_i, \tau), w_{c_i,c_j}) \tag{6.19}$$

$$\mathcal{A}(r_j, \tau + 1) = \beta(\mathcal{A}(r_j, \tau), \mathcal{A}(r_i, \tau), w_{r_i,r_j}) \tag{6.20}$$

Im Speziellen ergeben sich Gleichung 6.21 für die Aktivierungsausbreitung zwischen Konzeptknoten und Gleichung 6.22 für die Aktivierungsausbreitung zwischen Ressourceknoten.

$$\forall c_j \in C \text{ gilt} : \mathcal{A}(c_j, \tau + 1) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(c_i, \tau) \cdot w_{c_i,c_j}}{\sum_{k=1}^{s} w_{c_i,c_k}} + \mathcal{A}(c_j, \tau) & \text{für } t > 0 \\ \mathcal{A}(c_j, \tau) & \text{für } t = 0 \end{cases} \tag{6.21}$$

Wobei gilt:

- $\mathcal{A}(c_j, \tau)$ ... Aktivierung des Konzeptknotens zu Konzept j in Schritt τ
- $\mathcal{A}(c_i, \tau)$ ... Aktivierung des Konzeptknotens zu Konzept i in Schritt τ
- τ ... Schritt der Aktivierungsausbreitung, $\tau \in \mathbb{N}_0$
- t ... Anzahl der Konzeptknoten von denen Knoten c_j Aktivierung erhält
- w_{c_i,r_j} ... Gewicht der Verbindung zwischen einem Konzeptknoten c_i und einem Konzeptknoten c_j
- $w_{c_i,c_j} = semsim(c_i, c_j)$ (vgl. Gleichung 6.5)
- s ... Anzahl der Konzeptknoten an die Knoten c_i Aktivierung abgibt[12]

[12] $s > 0$ wenn $t > 0$ da c_i mit c_j verbunden ist. Somit gibt c_i zumindest an einen Knoten Aktivierung ab, wenn c_j von einem Knoten Aktivierung erhält

- w_{c_i,c_k} ... Gewicht der Verbindung zwischen einem Konzeptknoten c_i und einem Konzeptknoten c_k
- $w_{c_i,c_k} = semsim(c_i, c_k)$ (vgl. Gleichung 6.5)
- C ... Schicht an Knoten, welche Konzepte repräsentieren

$$\forall r_j \in R \text{ gilt} : \mathcal{A}(r_j, \tau+1) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(r_i,\tau) \cdot w_{r_i,r_j}}{\sum_{k=1}^{s} w_{r_i,r_k}} + \mathcal{A}(r_j,\tau) & \text{für } t > 0 \\ \mathcal{A}(r_j,\tau) & \text{für } t = 0 \end{cases} \tag{6.22}$$

Wobei gilt:

- $\mathcal{A}(r_j, \tau)$ ... Aktivierung des Ressourceknotens zu Ressource j in Schritt τ
- $\mathcal{A}(r_i, \tau)$ ... Aktivierung des Ressourceknotens zu Ressource i in Schritt τ
- τ ... Schritt der Aktivierungsausbreitung, $\tau \in \mathbb{N}_0$
- t ... Anzahl der Ressourceknoten von denen Knoten r_j Aktivierung erhält
- w_{r_i,r_j} ... Gewicht der Verbindung zwischen einem Ressourceknoten r_i und einem Ressourceknoten r_j
- $w_{r_i,r_j} = contsim(r_i, r_j)$ (vgl. Gleichung 6.7)
- s ... Anzahl der Ressourceknoten an die Knoten r_i Aktivierung abgibt[13]
- w_{r_i,r_k} ... Gewicht der Verbindung zwischen einem Ressourceknoten r_i und einem Ressourceknoten r_k
- $w_{r_i,r_k} = contsim(r_i, r_k)$ (vgl. Gleichung 6.7)
- R ... Schicht an Knoten, welche Ressourcen repräsentieren

[13] $s > 0$ wenn $t > 0$ da r_i mit r_j verbunden ist. Somit gibt r_i zumindest an einen Knoten Aktivierung ab, wenn r_j von einem Knoten Aktivierung erhält

Beispiel für die Berechnung der Aktivierungsausbreitung unter Berücksichtung von semantischer und inhaltsbasierter Ähnlichkeit

Die Ausbreitung von Aktivierung unter Berücksichtung von semantischer und inhaltsbasierter Ähnlichkeit nach den Gleichung 6.21, 6.17 and 6.22 soll nun anhand eines Beispiels verdeutlicht werden. Hierzu wird die Aktivierungsausbreitung im Netz aus Abbildung 6.6 berechnet.

Die Knoten der Konzeptschicht besitzen die folgenden initialen Aktivierungen:

- $\mathcal{A}(c_1, 1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_2, 1) = 0$
- $\mathcal{A}(c_3, 1) = 0$
- $\mathcal{A}(c_4, 1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_5, 1) = 0$
- $\mathcal{A}(c_6, 1) = 0$
- $\mathcal{A}(c_7, 1) = 0$
- $\mathcal{A}(c_8, 1) = \frac{1,0}{3} \approx 0,33$
- $\mathcal{A}(c_9, 1) = 0$

Folgende Kantengewichte werden für die Kanten der Konzeptschicht festgelegt:

- $w_{c_1,c_2} = 0,5$
- $w_{c_1,c_3} = 0,2$
- $w_{c_4,c_2} = 0,3$
- $w_{c_4,c_6} = 0,4$

Die Ausbreitung der Aktivierung von Konzept- zu Konzeptknoten wird nach Gleichung 6.21 berechnet. Für die Knoten c_1, c_2, c_3, c_4, c_6, c_7 und c_8 gilt der obere Fall ($t > 0$) aus Gleichung 6.21. Für die Knoten c_5 und c_9 gilt der untere Fall ($t = 0$) aus Gleichung 6.21.

- $\mathcal{A}(c_1, 2) = \frac{\mathcal{A}(c_2,1)\cdot w_{c_2,c_1}}{w_{c_2,c_1}+w_{c_2,c_4}} + \frac{\mathcal{A}(c_3,1)\cdot w_{c_3,c_1}}{w_{c_3,c_1}+w_{c_3,c_4}} + \mathcal{A}(c_1, 1) = 0+0+0,33 = 0,33$
- $\mathcal{A}(c_2, 2) = \frac{\mathcal{A}(c_1,1)\cdot w_{c_1,c_2}}{w_{c_1,c_2}+w_{c_1,c_3}} + \frac{\mathcal{A}(c_4,1)\cdot w_{c_4,c_2}}{w_{c_4,c_2}+w_{c_4,c_6}} + \mathcal{A}(c_2, 1) = 0,24 + 0,14 + 0 = 0,38$

- $\mathcal{A}(c_3,2) = \frac{\mathcal{A}(c_1,1)\cdot w_{c_1,c_3}}{w_{c_1,c_2}+w_{c_1,c_3}} + \mathcal{A}(c_3,1) = 0,09 + 0 = 0,09$
- $\mathcal{A}(c_4,2) = \frac{\mathcal{A}(c_2,1)\cdot w_{c_2,c_4}}{w_{c_2,c_4}+w_{c_2,c_1}} + \frac{\mathcal{A}(c_3,1)\cdot w_{c_3,c_4}}{w_{c_3,c_4}+w_{c_3,c_1}} + \frac{\mathcal{A}(c_5,1)\cdot w_{c_5,c_4}}{w_{c_5,c_4}} + \frac{\mathcal{A}(c_6,1)\cdot w_{c_6,c_4}}{w_{c_6,c_4}+w_{c_6,c_7}} + \mathcal{A}(c_4,1) = 0 + 0 + 0 + 0 + 0,33 = 0,33$
- $\mathcal{A}(c_5,2) = \mathcal{A}(c_5,1) = 0$
- $\mathcal{A}(c_6,2) = \frac{\mathcal{A}(c_4,1)\cdot w_{c_4,c_6}}{w_{c_4,c_2}+w_{c_4,c_6}} + \mathcal{A}(c_6,1) = 0,19 + 0 = 0,19$
- $\mathcal{A}(c_7,2) = \frac{\mathcal{A}(c_6,1)\cdot w_{c_6,c_7}}{w_{c_6,c_4}+w_{c_6,c_7}} + \frac{\mathcal{A}(c_8,1)\cdot w_{c_8,c_7}}{w_{c_8,c_7}} + \mathcal{A}(c_7,1) = 0 + 0,33 + 0 = 0,33$
- $\mathcal{A}(c_8,2) = \frac{\mathcal{A}(c_7,1)\cdot w_{c_7,c_8}}{w_{c_7,c_8}} + \frac{\mathcal{A}(c_9,1)\cdot w_{c_9,c_8}}{w_{c_9,c_8}} + \mathcal{A}(c_8,1) = 0 + 0 + 0,33 = 0,33$
- $\mathcal{A}(c_9,2) = \mathcal{A}(c_9,1) = 0$

Folgende Kantengewichte werden für die Kanten zwischen Konzept- und Ressourceschicht festgelegt:

- $w_{c_3,r_2} = 0,8$
- $w_{c_3,r_3} = 0,6$

Die Ausbreitung der Aktivierung von Konzept- zu Ressourceknoten wird nach Gleichung 6.17 berechnet. Für die Knoten r_1, r_2, r_3, r_5, r_6 und r_8 gilt der obere Fall ($t > 0$) aus Gleichung 6.17. Für die Knoten r_4 und r_9 gilt der untere Fall ($t = 0$) aus Gleichung 6.17.

- $\mathcal{A}(r_1,3) = \frac{\mathcal{A}(c_1,2)\cdot w_{c_1,r_1}}{w_{c_1,r_1}} + \frac{\mathcal{A}(c_2,2)\cdot w_{c_2,r_1}}{w_{c_2,r_1}} = 0,33 + 0,38 = 0,71$
- $\mathcal{A}(r_2,3) = \frac{\mathcal{A}(c_3,2)\cdot w_{c_3,r_2}}{w_{c_3,r_2}+w_{c_3,r_3}} = 0,05$
- $\mathcal{A}(r_3,3) = \frac{\mathcal{A}(c_3,2)\cdot w_{c_3,r_3}}{w_{c_3,r_2}+w_{c_3,r_3}} = 0,04$
- $\mathcal{A}(r_4,3) = 0$
- $\mathcal{A}(r_5,3) = \frac{\mathcal{A}(c_4,2)\cdot w_{c_4,r_5}}{w_{c_4,r_5}} + \frac{\mathcal{A}(c_6,2)\cdot w_{c_6,r_5}}{w_{c_6,r_5}} = 0,33 + 0,19 = 0,52$
- $\mathcal{A}(r_6,3) = \frac{\mathcal{A}(c_7,2)\cdot w_{c_7,r_6}}{w_{c_7,r_6}} + \frac{\mathcal{A}(c_9,2)\cdot w_{c_9,r_6}}{w_{c_9,r_6}+w_{c_9,r_8}} = 0,33 + 0 = 0,33$
- $\mathcal{A}(r_8,3) = \frac{\mathcal{A}(c_9,2)\cdot w_{c_9,r_8}}{w_{c_9,r_6}+w_{c_9,r_8}} = 0$
- $\mathcal{A}(r_9,3) = 0$

Folgende Kantengewichte werden für die Kanten der Ressourceschicht festgelegt:

- $w_{r_1,r_2} = 0,4$
- $w_{r_1,r_3} = 0,7$

- $w_{r_6,r_4} = 0,2$
- $w_{r_6,r_8} = 0,5$

Die Ausbreitung der Aktivierung von Ressource- zu Ressourceknoten wird nach Gleichung 6.22 berechnet. Für die Knoten r_2, r_3, r_4, r_6, r_8 und r_9 gilt der obere Fall ($t > 0$) aus Gleichung 6.22. Für die Knoten r_1 und r_5 gilt der untere Fall ($t = 0$) aus Gleichung 6.22.

- $\mathcal{A}(r_1, 4) = \mathcal{A}(r_1, 3) = 0,71$
- $\mathcal{A}(r_2, 4) = \frac{\mathcal{A}(r_1,3)\cdot w_{r_1,r_2}}{w_{r_1,r_2}+w_{r_1,r_3}} + \mathcal{A}(r_2, 3) = 0,26 + 0,05 = 0,31$
- $\mathcal{A}(r_3, 4) = \frac{\mathcal{A}(r_1,3)\cdot w_{r_1,r_3}}{w_{r_1,r_2}+w_{r_1,r_3}} + \mathcal{A}(r_3, 3) = 0,45 + 0,04 = 0,49$
- $\mathcal{A}(r_4, 4) = \frac{\mathcal{A}(r_5,3)\cdot w_{r_5,r_4}}{w_{r_5,r_4}} + \frac{\mathcal{A}(r_6,3)\cdot w_{r_6,r_4}}{w_{r_6,r_4}+w_{r_6,r_8}} + \mathcal{A}(r_4, 3) = 0,52 + 0,09 + 0 = 0,61$
- $\mathcal{A}(r_5, 4) = \mathcal{A}(r_5, 3) = 0,52$
- $\mathcal{A}(r_6, 4) = \frac{\mathcal{A}(r_4,3)\cdot w_{r_4,r_6}}{w_{r_4,r_6}} + \frac{\mathcal{A}(r_8,3)\cdot w_{r_8,r_6}}{w_{r_8,r_6}+w_{r_8,r_9}} + \mathcal{A}(r_6, 3) = 0+0+0,33 = 0,33$
- $\mathcal{A}(r_8, 4) = \frac{\mathcal{A}(r_6,3)\cdot w_{r_6,r_8}}{w_{r_6,r_4}+w_{r_6,r_8}} + \mathcal{A}(r_8, 3) = 0,24 + 0 = 0,24$
- $\mathcal{A}(r_9, 4) = \frac{\mathcal{A}(r_8,3)\cdot w_{r_8,r_9}}{w_{r_8,r_6}+w_{r_8,r_9}} + \mathcal{A}(r_9, 3) = 0 + 0 = 0$

Das Ergebnis, welches an den Benutzer retourniert wird, beträgt nach initialer Aktivierung und 3 Schritten der Aktivierungsausbreitung: $E = \{r_1, r_2, r_3, r_4, r_5, r_6, r_8\}$. Die Aktivierung der Ressourcen in der Ergebnismenge ist gleich ihrer Aktivierung im vierten Schritt der Aktivierungsausbreitung (siehe oben).

6.4 Vergleich mit anderen Modellen

Im vorgestellten Netzmodell erfolgt (wie auch beispielsweise in (Wilkinson u. Hingston, 1991) oder (Crouch u. a., 1994b), in Abschnitt 5.3) zur Kontrolle der Ausbreitung der Aktivierung ausschließlich eine Gewichtung der Kanten. Knoten werden nicht gewichtet (wie es beispielsweise in (Mothe, 1994), in Abschnitt 5.3, der Fall ist).

Im Gegensatz zu dem in (Wilkinson u. Hingston, 1991) (vgl. Abschnitt 5.3) präsentierten Modell, gibt es keine eigene Schicht für Anfragen. Jene Knoten, welche zu den Konzepten in der Anfrage korrespondieren, werden

bei einer Anfrage durch einen externen Stimulus aktiviert, welcher den Aktivierungsprozess im Netz initiiert. Die Höhe des externen Stimulus entspricht dem Gewicht des Konzepts in der Anfrage. Dies ist äquivalent zu dem von (Mothe, 1994) (vgl. Abschnitt 5.3) vorgestellten Modell.

Die Aktivierungsausbreitung im vorgestellten Netzmodell findet unter dem Gesichtspunkt der Aktivierungserhaltung statt. Jeder Knoten kann nur soviel Aktivierung abgeben wie er aufgenommen hat. Dies drückt sich durch die Verwendung der L1-Norm in Gleichung 6.9 aus.

Das hier vorgestellte Netzmodell weist Charakteristika von klassischen Netzmodellen im Information Retrieval auf (vgl. Abschnitte 4.5 und 5) und wendet diese zur assoziativen Suche an. Die Verwendung von Kantengewichten auf der einen Seite und keinen Gewichten für Knoten auf der anderen Seite, sowie das Fehlen einer Schicht für Anfragen stellen Freiheiten in der Modellierung dar. Das hier vorgestellte Modell könnte auch mit diesen Eigenschaften realisiert werden. Die Verwendung von Aktivierungserhaltung erfolgt allerdings mit dem konkreten Zweck Überaktivierung (vgl. Abschnitt 5.4) zu vermeiden.

6.5 Zusammenfassung

In diesem Abschnitt wurde das im Rahmen dieser Arbeit entwickelte Retrieval-Modell vorgestellt. Nach einer Einleitung in das Thema der semantischen Annotation von Ressourcen wurde der Aufbau des Netzmodells beschrieben. Danach folgte die Darstellung der grundlegenden und erweiterten, assoziativen Suchfunktionalität des Modells. Zum Abschluss wurde das Modell den in Abschnitt 5 vorgestellten Modellen gegenübergestellt.

7 Implementierung des Modells

7.1 Einleitung

In diesem Kapitel wird das im Rahmen dieser Arbeit entwickelte System beschrieben. Der entwickelte Suchansatz wird in einer Semantic-Desktop-Umgebung implementiert und erprobt. Bei den zu suchenden Ressourcen handelt es sich im konkreten Anwendungsfall um Dokumente im APOSDLE-System, einem System zur Unterstützung von Wissensarbeitern am Arbeitsplatz (vgl. Abschnitt 7.2).

In diesem Abschnitt werden der Anwendungsfall des Systems (vgl. Abschnitt 7.2) und die Erzeugung und Speicherung der vom System verwendeten semantischen Annotationen beschrieben (vgl. Abschnitt 7.3). Des Weiteren wird die Spezialisierung des Modells aus Kapitel 6 auf die Suche nach Dokumenten veranschaulicht (vgl. Abschnitt 7.4). Ebenfalls werden die verwendeten Maße zur Gewichtung von Kanten, mit denen das Netzmodell aus Abschnitt 6 für den vorgestellten Anwendungsfall parametrisiert wird (vgl. Abschnitt 7.5), angeführt. In Folge wird auf die dem entwickelten System zugrunde liegende Architektur eingegangen und die Funktionalität der einzelnen Komponenten erläutert (vgl. Abschnitt 7.6). Neben der Beschreibung von Details zur Umsetzung, wie beispielsweise die verwendeten Software (vgl. Abschnitt 7.7.1), wird die Implementierung des verwendeten Spreading-Activation-Ansatzes hinsichtlich ihrer Komplexität untersucht (vgl. Abschnitt 7.7.3).

7.2 Kontext der Anwendung und verwendete Datenbasis

Das vorgestellte System wurde im Rahmen des APOSDLE-Projekts[1] entwickelt. APOSDLE ist ein von der Europäischen Union gefördertes Integriertes Projekt im 6. Rahmenprogramm im Bereich Technologiegestütztes Lernen. 12 Partner aus 7 Ländern der EU arbeiten gemeinsam daran die Produktivität von Wissensarbeitern durch Lernen am Arbeitsplatz zu erhöhen. APOSDLE zielt darauf ab Wissensarbeitern während ihrer Arbeit Hilfestellungen für diese zu geben und ihnen in weiterer Folge den Aufbau von Kompetenzen für die Bewältigung von ähnlichen Situationen zu ermöglichen. Diese Hilfestellungen sollen in einem generischen System implementiert werden, welches nicht auf eine Anwendungsdomäne zugeschnitten ist (Lindstaedt u. a., 2008; Lindstaedt u. Mayer, 2006).

Aus diesem Grund setz APOSDLE auf die Verwendung von Ontologien für die Trennung von Domänenwissen und betriebsbedingtem Wissen (vgl. Abschnitt 2.2.3). Für den Umgang mit Ontologien baut das APOSDLE-System auf Technologien für das Semantic Web auf. APOSDLE kann also nach der Definition aus Abschnitt 2.2.2 als ein System für den Semantic Desktop gesehen werden, da es Technologien für das Semantic Web in einer Desktop-Umgebung anwendet. Somit bietet das APOSDLE-System eine ideale Umgebung um ein Modell, welches für das Semantic Web entwickelt wurde, zu implementieren und zu erproben.

Die dem hier vorgestellten System zugrunde liegende Datenbasis basiert auf jener, welche für die erste Version des APOSDLE-Systems (APOSDLE consortium, 2007) entwickelt wurde. Die Datenbasis setzt sich aus einer Wissensbasis in Form einer Ontologie und einer Menge an Dokumenten zusammen. Elemente aus der Wissensbasis werden dazu verwendet diese Dokumente zu beschreiben. Diese Information wird in der Wissensbasis in Form von einer Teil-Ontologie gekapselt.

In der ersten Version des APOSDLE-Systems wurde *Requirements Engineering* als Anwendungsdomäne für das System ausgewählt. Dieses Wissens-

[1] `http://www.aposdle.org/` (18.05.2008)

gebiet wurde als Ontologie modelliert. Dokumente, welche unterstützendes Material zum Thema Requirements Engineering enthalten, wie beispielsweise Definitionen, Beispiele oder Kurse, wurden zum Teil mit Konzepten aus dieser Ontologie annotiert.

Die verwendete Wissensbasis wurde von Mitarbeitern von *EADS Innovation Works*[2], einem Partner im APOSDLE-Projekt, mit Hilfe des Ontologie-Editors Protégé[3] erstellt. Die Dokumentenmenge wurde von Mitarbeitern des *Centre for HCI Design*[4] der *City University London*, einem weiteren Partner im APOSDLE-Projekt, zur Verfügung gestellt, welcher über Expertise im Themengebiet Requirements Engineering verfügt. Die Annotation von Dokumenten mit Konzepten aus der Ontologie erfolgte von den beiden Partnern, welche die Ontologie modelliert bzw. die Dokumente zur Verfügungen gestellt hatten.

Die verwendete Ontologie besteht aus 79 Konzepten von denen 21 für die Annotation von Dokumenten verwenden werden. Die Dokumentenbasis besteht aus 1016 Dokumenten. 496 Dokumente sind mit einem oder mehreren Konzepten aus der Wissensbasis annotiert. Somit werden nur Teile der Ontologie für die Annotation von Dokumenten verwendet und nur eine Teilmenge der Dokumente wird mit Konzepten aus der Ontologie annotiert. Die Verbindungen zwischen den vorhanden Informationsobjekten (den Dokumenten) und der semantischen Zusatzinformation (der Ontologie) sind spärlich ausgeprägt.

Hierdurch ergibt sich folgende Problematik für die Suche nach Dokumenten: Enthält eine Suchanfrage Konzepte aus der Ontologie, welche nicht für die Annotation von Dokumenten genutzt wurden oder für die nur ein Teil an Dokumenten annotiert wurde, werden durch eine typischen Suche in wissensbasierten Systemen, wie beispielsweise mit SPARQL (vgl. Abschnitt 2.2.1), relevante Dokumente nicht gefunden. Assoziatives Retrieval wird im vorgestellten System für den Semantic Desktop dafür genutzt relevante Dokumente aufzufinden, auch wenn diese nicht mit Konzepten aus der Domänenontologie annotiert wurden bzw., wenn zur Annotation der

[2] `http://www.eads.net/` (18.05.2008)
[3] `http://protege.stanford.edu/` (18.05.2008)
[4] `http://www-hcid.soi.city.ac.uk/` (18.05.2008)

Dokumente andere Konzepte als jene in der Anfrage verwendet wurden.

7.3 Semantische Annotation von Dokumenten

Um (Konzepte aus) Ontologien für eine Suche nach Dokumenten nutzbar zu machen, müssen diese mit der zu durchsuchenden Dokumentenmenge verbunden werden. Dokumente werden mit Elementen aus einer Ontologie *semantisch annotiert*.

Die semantische Annotation im vorliegenden System basiert auf der *Aboutness* der Dokumente. Ein Dokument wird mit einem Konzept versehen, wenn der Inhalt des Dokuments vom Konzept handelt. Dieser Zusammenhang wird formal über eine Eigenschaft *deals with* beschrieben, welche in einem gekapselten Teil der Wissensbasis, der zur Ablage von Annotationen dient, modelliert wird. Abschnitt 6.3.1 diskutiert den Hintergrund des gewählten Ansatzes zur semantischen Annotation.

7.3.1 Erzeugung der Annotationen

Der Prozess der Annotation wird durch ein Plug-in für den Ontologie-Editor Protégé[5] unterstützt. Die ursprüngliche Version des Plug-ins namens *OntologyMapper* wird in (Scheir u. a., 2005a) und im Detail in (Scheir u. a., 2006a) beschrieben. Auf der existierenden Funktionalität aufbauend wurde das Plug-in überarbeitet und neue Funktionalität aus dem Umfeld von Ontologie Lernen hinzugefügt (Scheir u. a., 2006b). In seiner aktuellsten Version ist es als *Annotation Tab* Teil des *Domain Modelling Tool* (Pammer u. a., 2007), welches im APOSDLE-Projekt zur Modellierung der Domänenontologie und zur Annotation von Dokumenten mit Konzepten aus der Ontologie dient. Abbildung 7.1 zeigt einen Screenshot des Annotation Tabs.

Zusätzlich zur Bereitstellung einer grafischen Benutzeroberfläche für die Auszeichnung von Dokumenten, wird auch die Funktionalität eines Klassifikationsalgorithmus angeboten. Dieser schlägt für ein neu zu annotierendes Dokument eine Menge an Konzepten vor. Dieser Vorschlag basiert auf den bereits vorhandenen Annotationen von Dokumenten mit Konzepten. Beim

[5] `http://protege.stanford.edu/` (18.05.2008)

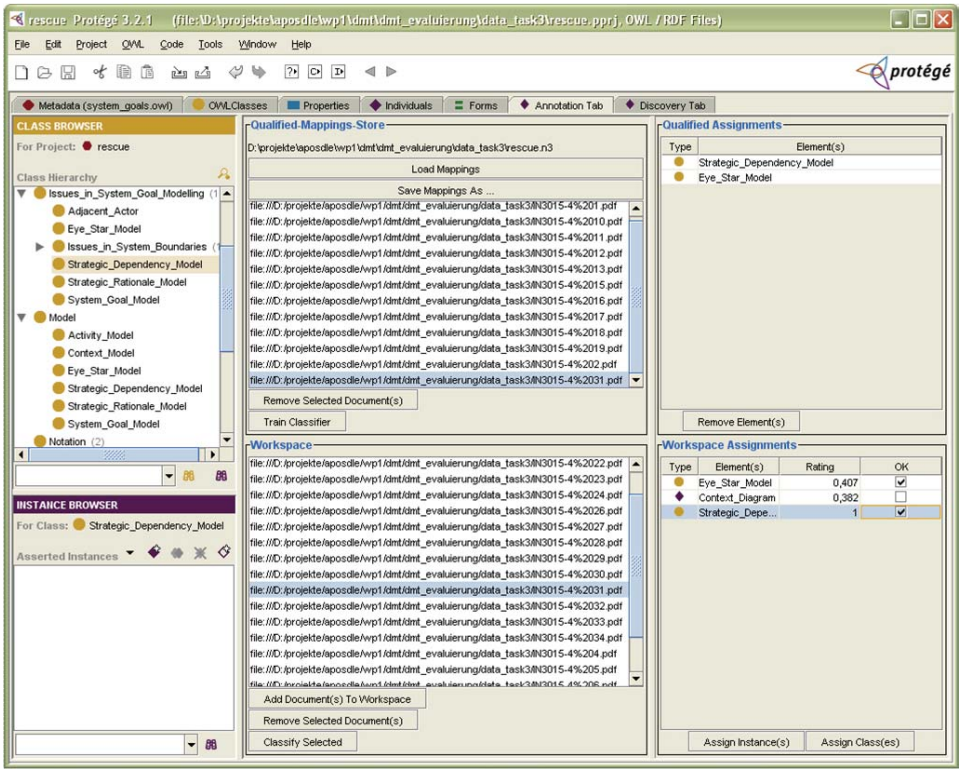

Abbildung 7.1: Screenshot des Annotation Tabs

verwendeten Klassifikationsalgorithmus handelt es sich um einen *k-Nearest-Neighbor-Algorithmus* auf Basis der Suchmaschine Lucene.

7.3.2 Speicherung der semantischen Annotationen

Uren u. a. (2006) stellen fest, dass es zwei grundlegend unterschiedliche Ansätze für die Speicherung von Annotationen gibt und bezeichnen diese als das *Semantic-Web-Modell* (*Semantic Web model*) und als das *Textverarbeitungsprogramm-Modell* (*word processor model*). Im Semantic-Web-Modell erfolgt die Speicherung der Annotationen getrennt von den Dokumenten. Ein Beispiel für diese Vorgehensweise stellt das Resource Description Framework (RDF) dar, Ressourcen werden aus einem externen Modell referenziert. Im Textverarbeitungsprogramm-Modell erfolgt die Speicherung der Annotationen als Bestandteil des Dokuments. Beispiele für diesen Ansatz sind Annotationen in HTML-Seiten oder Eigenschaften in Microsoft-

Office-Dokumenten. Das Textverarbeitungsprogramm-Modell ist der klassische Weg im Umgang mit Annotationen in Informationsmanagement-Systemen, während im Semantic-Web-Umfeld vornehmlich das Semantic-Web-Modell vertreten ist (Uren u. a., 2006).

Im hier vorgestellten System wird das Semantic-Web-Modell verwendet, um auch die Annotation von Dokumenten, in die keine Annotationen eingebettet werden können zu ermöglichen. Die Annotationen von Dokumenten mit Konzepten aus der Ontologie werden in einem speziellen Teil der Wissensbasis abgelegt. Die Trennung in zwei Teil-Ontologien, eine zur Modellierung des Domänenmodells und eine zur Annotation von Dokumenten mit Konzepten, erfolgt zur Kapselung der beiden Modelle. Dies ermöglicht es, dass die Domänenontologie mit Hilfe von OWL DL (vgl. Abschnitt 2.2.1) repräsentiert wird, während die Annotationsontologie in OWL Full oder RDF verbleiben kann. Somit bleibt die Möglichkeit zum Ziehen von Schlüssen in der Domänenontologie erhalten und geht nicht durch die Verwendung eines schwächeren Formalismus verloren.

Die Ontologie welche die Annotationen enthält wird in OWL Full formalisiert, da OWL DL es nicht erlaubt eine Klasse als Eigenschaftswert zu verwenden[6]. Der Weg Dokumente direkt als Instanzen jener Klassen zu speichern, mit denen sie annotiert werden sollen wird bewusst vermieden, da diese Art der Relation eine andere Semantik mit sich bringen würde. Durch den gewählten Ansatz besteht die Möglichkeit zu modellieren, dass ein Dokument von einer Menge an Konzepten handelt, aber auch, dass es sich bei einem Dokument um die Instanz eines Konzepts handelt (wie es beispielsweise für das Konzept *Anforderungsdokument* der Fall sein könnte).

Zur Formalisierung der Annotationen wird die Klasse `AnnotatedDocument` modelliert, welche Ausgangspunkt der Eigenschaft `dealsWith` sein kann[7]. Instanzen der Klasse `AnnotatedDocument` repräsentieren annotierte Dokumente im System und verweisen auf jene Konzepte von denen das Dokument *handelt*. Um Dokumente mit Kon-

[6] OWL DL erlaubt ausschließlich Instanzen und Literale als Eigenschaftswerte

[7] Im verwendeten Formalismus OWL sind Eigenschaften nicht, wie in der objekt-orientierten Programmierung, Teile von Klassen. Stattdessen wird eine Klasse als möglicher Ausgangspunkt (domain) bzw. Endpunkt (range) einer Eigenschaft festgelegt.

zepten aus der Ontologie zu annotieren wird eine Instanz der Klasse `AnnotatedDocument` erzeugt. Von dieser Instanz wird auf das zu annotierende Dokument und jene Konzepte, welche zur Annotation des Dokuments verwendet werden sollen, verwiesen. Für den Verweis auf das zu annotierende Dokument wird die Eigenschaft `hasResource` verwendet. Für den Verweis auf die zur Annotation verwendeten Konzepte findet die Eigenschaft `dealsWith` Verwendung. Abbildung 7.2 zeigt diesen Zusammenhang.

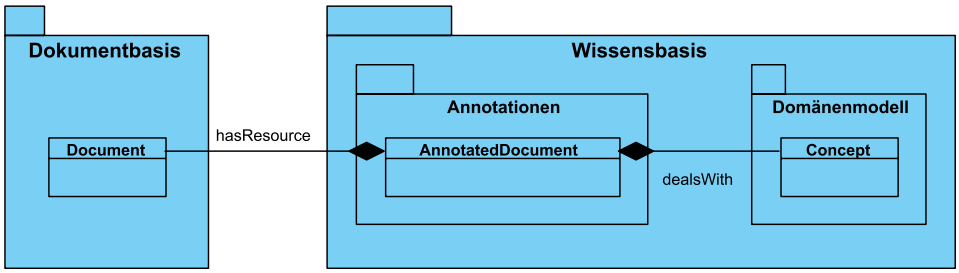

Abbildung 7.2: Zusammenhang zwischen Dokumenten, Annotationen und Konzepten in Dokumentbasis und Wissensbasis

7.4 Spezialisierung des Retrieval-Modells

Das hier vorgestellte System ermöglicht ausgehend von einer Menge an Konzepten die Suche nach einer Menge an Dokumenten, welche von diesen Konzepten handeln. Zusätzlich zur grundlegenden Suchfunktionalität nach den Dokumenten zu den Konzepten, welche die Anfrage bilden, stellt das System eine assoziative Suchfunktionalität zur Verfügung, um auch relevante Dokumente zu finden, welche nicht mit den Konzepten in der Anfrage annotiert wurden.

Für die Implementierung im hier vorgestellten System erfolgt eine Spezialisierung das Modells aus Kapitel 6. Die Einschränkung beim hier vorgestellten System ist, dass es sich bei den zu suchenden Ressourcen ausschließlich um Textdokumente handelt. Die inhaltsbasierte Ähnlichkeit zwischen zwei Ressourcen wird somit mittels eines Maßes für die textuelle Ähnlichkeit berechnet (vgl. Abschnitt 7.5.3).

Im Folgenden werden die durch diese Einschränkung betroffenen Gleichungen aus Kapitel 6 angeführt. Um die Lesbarkeit dieses Kapitels zu

erhöhen werden bewusst Elemente aus Kapitel 6 wiederholt. Änderungen zum Modell in Kapitel 6 sind durch ▶ in den Erläuterungen zu den Gleichungen hervorgehoben.

7.4.1 Schichten des Netzes

Das dem vorgestellten System zugrunde liegende Netzmodell besteht aus zwei Schichten, einer Schicht C an Knoten, welche Konzepte repräsentieren (vgl. Gleichung 7.1) und einer Schicht D[8] an Knoten, welche für Textdokumente stehen (vgl. Gleichung 7.2).

$$C = \{c_1, ..., c_N\} \tag{7.1}$$

$$D = \{d_1, ..., d_M\} \tag{7.2}$$

Wobei für Gleichung 7.1 und Gleichung 7.2 gilt:

- C .. Schicht an Knoten, welche Konzepte repräsentieren
- ▶ D .. Schicht an Knoten, welche Dokumente repräsentieren
- $c_1, ..., c_N$ Konzeptknoten 1 bis N
- ▶ $d_1, ..., d_M$ Dokumentknoten 1 bis M
- N ist gleich der Anzahl der Konzepte im System
- ▶ M ist gleich der Anzahl der Dokumente im System
- $\forall c_i \in C$ gilt : $\mathcal{C}(c_i) \in K$
- ▶ $\forall d_j \in D$ gilt : $\mathcal{D}(d_j) \in U$
- $\mathcal{C}(c_i)$ ... Konzept für das Konzeptknoten c_i steht
- ▶ $\mathcal{D}(d_j)$ ... Dokument für das Dokumentknoten d_j steht
- K ... Menge der Konzepte in der Ontologie
- ▶ U ... Menge der Dokumente im System

Die Knoten der Konzeptschicht sind mit den Knoten der Dokumentschicht verbunden. Die Verbindungen sind von den Konzeptknoten zu den

[8] Hierbei handelt es sich um einen Spezialfall der Schicht an Ressourcen R, welche in Abschnitt 6.3.2 beschrieben wurde

Dokumentknoten gerichtet. Des Weiteren sind die Verbindungen zwischen Konzeptknoten und Dokumentknoten gewichtet (w_{c_i,d_j}, siehe Gleichung 7.3). Das Gewicht spiegelt die Wichtigkeit des Dokuments für das Konzept wider.

Gleichung 7.3 zeigt das Kantengewicht w_{c_i,d_j} einer Kante von einem Konzeptknoten c_i zu einem Dokumentknoten d_i.

$$w_{c_i,d_j} = weight(c_i, d_j) \tag{7.3}$$

Wobei gilt:

- w_{c_i,d_j} ... Kantengewicht der Kante zwischen Konzeptknoten c_i und Dokumentknoten d_j
- $weight(c_i, d_j)$ ... Gewicht der Annotation, die durch die Kante zwischen Konzeptknoten c_i und Dokumentknoten d_j repräsentiert wird. Berechnet nach Gleichung 7.11.
- $0 \leq weight(c_i, d_j) \leq 1$

Gleichung 7.4 zeigt die Menge aller Kanten, welche die Schicht an Konzeptknoten und die Schicht an Dokumentknoten miteinander verbinden, wobei eine Kante zwischen einem Konzeptknoten c_i zu einem Dokumentknoten d_j nur dann erzeugt wird, falls das Kantengewicht w_{c_i,d_j} dieser Kante > 0 ist. Für den Fall, dass keine Annotationen von Dokumenten mit Konzepten vorhanden sind oder, dass die Gewichte aller Annotationen $= 0$ sind, besteht das Netz aus zwei nicht miteinander verbundene Schichten.

$$W_{C,D} = \{w_{c_i,d_j} | w_{c_i,d_j} > 0\} \tag{7.4}$$

- $W_{C,D}$ ... Menge aller Kanten zwischen C und D
- C ... Schicht an Knoten, welche Konzepte repräsentieren
- D ... Schicht an Knoten, welche Dokumente repräsentieren
- $c_1, ..., c_N$ ... Konzeptknoten 1 bis N
- $d_1, ..., d_M$ ... Dokumentknoten 1 bis M
- N ist gleich der Anzahl der Konzepte im System
- M ist gleich der Anzahl der Dokumente im System

7.4.2 Modellierung der text-basierten Ähnlichkeit zwischen Dokumenten

Die Knoten der Dokumentschicht können untereinander verbunden sein. In diesem Fall wir die textuelle Ähnlichkeit als Gewichtungsfunktion für die Kanten zwischen zwei Dokumentknoten herangezogen (w_{d_i,d_j}, siehe Gleichung 7.5). Auch hier wird die Ähnlichkeit zwischen zwei Dokumenten über zwei Kanten repräsentiert um, so wie bei der semantischen Ähnlichkeit, asymmetrische Ähnlichkeitsmaße zu unterstützen.

Gleichung 7.5 zeigt das Kantengewicht w_{d_i,d_j} einer Kante von einem Dokumentknoten d_i zu einem Dokumentknoten d_j.

$$w_{d_i,d_j} = sim(d_i, d_j) \tag{7.5}$$

Wobei gilt:

- w_{d_i,d_j} ... Kantengewicht der Kante zwischen Dokumentknoten d_i und Dokumentknoten d_j
- $sim(d_i, d_j)$ ... Textuelle Ähnlichkeit zwischen den Dokumenten, welche durch die Dokumentknoten d_i und d_j repräsentiert werden. Berechnet nach Gleichung 7.16.
- $0 \leq sim(d_i, d_j) \leq 1$

Gleichung 7.6 zeigt die Menge aller Kanten, welche Dokumentknoten mit Dokumentknoten verbinden, wobei eine Kante zwischen einem Dokumentknoten d_i zu einem anderen Dokumentknoten d_j nur dann erzeugt wird, falls das Kantengewicht w_{d_i,d_j} dieser Kante > 0 ist.

$$W_{D,D} = \{w_{d_i,d_j} | w_{d_i,d_j} > 0\} \tag{7.6}$$

- $W_{D,D}$ ... Menge aller Kanten zwischen Dokumenten in D
- D ... Schicht an Knoten, welche Dokumente repräsentieren
- $d_1, ..., d_M$ ... Dokumentknoten 1 bis M
- M ist gleich der Anzahl der Dokumente im System

7.4.3 Suche im System

Ziel einer Suche im System ist es zu einer Anfrage, welche aus einer Menge an Konzepten besteht, eine Menge an Dokumenten zu finden und in weiterer Folge die Ergebnismenge nach Relevanz für die Anfrage zu reihen. Zu diesem Zweck wird die Netzstruktur, die dem hier vorgestellten System zu Grunde liegt, mittels eines Spreading-Activation-Ansatzes durchsucht. Ausgehend von einer Menge an initial aktivierten Konzeptknoten, breitet sich Aktivierung über die Kanten im Netz zu jenen Dokumentknoten aus, welche mit diesen Konzeptknoten verbunden sind.

Während im vorgestellten System die Gewichtung der Kanten im Vergleich zum Modell in Abschnitt 6 parametrisiert werden (vgl. Abschnitte 7.5.1, 7.5.2 und 7.5.3), bleibt der im Modell beschriebene Ansatz zur Suche unverändert. Die zentrale Spezialisierung des Systems im Vergleich zum Modell ist, dass es sich beim Typ der zu findenden Ressourcen um den Spezialfall Dokumente handelt.

Im Folgenden wird darauf verzichtet, die in Abschnitt 6.3 beschriebene Suchfunktionalität im Detail zu wiederholen. Stattdessen werden jene Gleichungen aus Abschnitt 6.3, welche die Ausbreitung der Aktivierung beschreiben, adaptiert an den Spezialfall der Suche nach Textdokumenten, angeführt.

Eine Suche im Assoziativen Netz kann wie in Abschnitt 6.3 beschrieben als Funktion σ formuliert werden, welche die Menge $\overline{A}$ aller Anfragen an das System, in die Menge $\overline{E}$ aller Ergebnisse des Systems transformiert (siehe Formel 7.7).

$$\sigma : \overline{A} \rightarrow \overline{E} \tag{7.7}$$

Im Gegensatz zu Formel 6.12 in Abschnitt 6.3 gilt allerdings für den Spezialfall der Suche nach Textdokumenten:

- $\overline{A}$ ... Menge aller Anfragen an das System
- $\overline{A} = \{\ A \mid A$ ist eine Anfrage an das System und $A \subseteq K\ \}$
- A ... Anfrage an das System
- K ... Menge der Konzepte in der Ontologie
- $\overline{E}$ ... Menge aller Ergebnisse des Systems

- $\overline{E} = \{\ E \mid E$ ist ein Ergebnis einer Suche und $E \subseteq U\ \}$
- E ... Ergebnis einer Suche
- U ... Menge der Dokumente im System

Zeitlicher Verlauf der Aktivierungsausbreitung

Auch bei der Beschreibung des zeitlichen Verlaufs der Aktivierungsausbreitung findet statt einem Ressourceknoten ein Dokumentknoten Verwendung. Die Aktivierung eines Dokumentknoten $\mathcal{A}(d_j, \tau + 1)$ im Schritt $\tau + 1$ wird aus der Aktivierung $\mathcal{A}(c_i, \tau)$ jener Knoten, von denen er Aktivierung erhält im Schritt τ und dem Gewicht der Verbindungen w_{c_i,d_j} zu diesen Knoten berechnet. Gleichung 7.8 repräsentiert diesen Zusammenhang durch die Funktion α.

$$\mathcal{A}(d_j, \tau + 1) = \alpha(\mathcal{A}(c_i, \tau), w_{c_i,d_j}) \tag{7.8}$$

Unten Einbeziehung von Gleichung 6.9 für die allgemeine Aktivierungsausbreitung folgt daraus im Speziellen Gleichung 7.9:

$$\forall d_j \in D \text{ gilt} : \mathcal{A}(d_j, \tau + 1) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(c_i, \tau) \cdot w_{c_i,d_j}}{\sum_{k=1}^{s} w_{c_i,d_k}} & \text{für } t > 0 \\ 0 & \text{für } t = 0 \end{cases} \tag{7.9}$$

Wobei gilt:

- $\mathcal{A}(d_j, \tau)$ ... Aktivierung des Dokumentknotens zu Dokument j in Schritt τ
- $\mathcal{A}(c_i, \tau)$ ... Aktivierung des Konzeptknotens zu Konzept i in Schritt τ
- τ ... Schritt der Aktivierungsausbreitung, $\tau \in \mathbb{N}_0$
- t ... Anzahl der Konzeptknoten von denen Knoten d_j Aktivierung erhält
- w_{c_i,d_j} ... Gewicht der Verbindung zwischen einem Konzeptknoten c_i und einem Dokumentknoten d_j
- $w_{c_i,d_j} = weight(c_i, d_j)$, berechnet nach Gleichung 7.3

- s ... Anzahl der Dokumentknoten an die Knoten c_i Aktivierung abgibt[9]
- w_{c_i,d_k} ... Gewicht der Verbindung zwischen einem Konzeptknoten c_i und einem Dokumentknoten d_k
- $w_{c_i,d_k} = weight(c_i, d_k)$, berechnet nach Gleichung 7.3
- D ... Schicht an Knoten, welche Dokumente repräsentieren

Suche unter Berücksichtigung von textueller Ähnlichkeit

Für die Ausbreitung der Aktivierung unter Berücksichtigung von textueller Ähnlichkeit findet ebenfalls ein Dokumentknoten statt einem Ressourceknoten Verwendung. Hierbei ergibt sich für Ausbreitung der Aktivierung von Dokumentknoten zu Dokumentknoten unter Berücksichtigung des in Abschnitt 7.5.3 angeführten Kantengewichtes w_{d_i,d_j} Gleichung 7.10 für die Aktivierungsausbreitung zwischen Dokumentknoten.

$$\forall d_j \in D \text{ gilt} : \mathcal{A}(d_j, \tau+1) = \begin{cases} \sum_{i=1}^{t} \frac{\mathcal{A}(d_i,\tau) \cdot w_{d_i,d_j}}{\sum_{k=1}^{s} w_{d_i,d_k}} + \mathcal{A}(d_j,\tau) & \text{für } t > 0 \\ \mathcal{A}(d_j,\tau) & \text{für } t = 0 \end{cases} \tag{7.10}$$

Wobei gilt:

- $\mathcal{A}(d_j, \tau)$ ... Aktivierung des Dokumentknotens zu Dokument j in Schritt τ
- $\mathcal{A}(d_i, \tau)$ ... Aktivierung des Dokumentknotens zu Dokument i in Schritt τ
- τ ... Schritt der Aktivierungsausbreitung, $\tau \in \mathbb{N}_0$
- t ... Anzahl der Dokumentknoten von denen Knoten d_j Aktivierung erhält
- w_{d_i,d_j} ... Gewicht der Verbindung zwischen einem Dokumentknoten d_i und einem Dokumentknoten d_j
- $w_{d_i,d_j} = sim(d_i, d_j)$, berechnet nach Gleichung 7.5

[9] $s > 0$ wenn $t > 0$ da c_i mit d_j verbunden ist. Somit gibt c_i zumindest an einen Knoten Aktivierung ab, wenn d_j von einem Knoten Aktivierung erhält

- s ... Anzahl der Dokumentknoten an die Knoten d_i Aktivierung abgibt[10]
- w_{d_i,d_k} ... Gewicht der Verbindung zwischen einem Dokumentknoten d_i und einem Dokumentknoten d_k
- $w_{d_i,d_k} = sim(d_i, d_k)$, berechnet nach Gleichung 7.5
- D ... Schicht an Knoten, welche Dokumente repräsentieren

7.5 Parametrisierung des Retrieval-Modells

Neben der Spezialisierung auf die Suche nach Dokumenten, welche im vorherigen Abschnitt eingeführt wurde, erfolgt für die Implementierung im hier vorgestellten System auch eine Parametrisierung des Modells: Das in Kapitel 6 beschriebenen Modell legt nicht fest welche Maße für die Berechnung der Kantengewichte des Netzes zu verwenden sind. Im folgenden Abschnitt werden jene Maße angeführt, die im umgesetzten System zur Berechnung der Kantengewichte verwendet werden. Hierbei handelt es sich um ein Maß zur Gewichtung der semantischen Annotationen von Dokumenten mit Konzepten (Abschnitt 7.5.1), drei Maße zur Berechnung der semantischen Ähnlichkeit zwischen zwei Konzepten (Abschnitt 7.5.2) und ein Maß zur Berechnung der textuellen Ähnlichkeit zwischen zwei Dokumenten (Abschnitt 7.5.3).

7.5.1 Bestimmung des Gewichts der Annotationen

Im hier vorgestellten System werden die Annotationen von Dokumenten mit Konzepten gewichtet, um eine Reihung der Suchergebnisse zu ermöglichen. Abschnitt 6.3.3 beleuchtet den Hintergrund dieser Vorgehensweise.

Die Gewichtung im hier vorgestellten System basiert auf einem tf*idf-ähnlichem Maß. Das Gewicht einer Annotation ist umso höher, je höher die Wichtigkeit eines Dokuments für ein Konzept ist. Tf*idf (vgl. Abschnitt 5.2.2) ist ein Standardinstrument im Information Retrieval um Verbindungen zwischen Dokumenten und den Termen, welche in ihnen vorkommen, zu

[10] $s > 0$ wenn $t > 0$ da d_i mit d_j verbunden ist. Somit gibt d_i zumindest an einen Knoten Aktivierung ab, wenn d_j von einem Knoten Aktivierung erhält

gewichten (Robertson u. Spärck Jones, 1994). Das hier verwendete Gewichtungsschema ist zu dem von Castells u. a. (2007) verwendeten verwandt, welches auch auf tf*idf basiert. Gleichung 7.11 zeigt die Gewichtung der Annotationen.

$$weight(c,d) = \begin{cases} tf(c,d) \cdot idf(c) = tf(c,d) \cdot \log \frac{D}{a(c)} & \text{für } a(c) > 0 \\ 0 & \text{für } a(c) = 0 \end{cases} \quad (7.11)$$

Wobei gilt:

- c ... ein Konzept
- d ... ein Dokument
- $tf(c,d)$ ... 1 wenn d mit c annotiert ist, 0 sonst
- $idf(c)$ ... inverse Dokumenthäufigkeit von Konzept c
- D ... Anzahl aller Dokumente im System
- $a(c)$ ... Anzahl der Dokumente im System, welche mit Konzept c annotiert sind[11]

7.5.2 Bestimmung der semantischen Ähnlichkeit zwischen Konzepten

Im Rahmen dieser Arbeit werden drei Ähnlichkeitsmaße dazu genutzt die semantische Ähnlichkeit zwischen Konzepten in einer Ontologie zu bestimmen und somit die Kanten der Konzeptschicht des Assoziativen Netzes zu gewichten. Die Konfigurationen des implementierten Systems auf Basis dieser drei Maße werden in die Evaluierung des entwickelten Systems mit einbezogen (vgl. Abschnitt 8). Bei den drei verwendeten Ähnlichkeitsmaßen handelt es sich um ein Maß basierend auf dem kürzesten Pfad (Shortest-Path) zwischen zwei Konzepten in einer gemeinsamen Klassenhierarchie, das Maß von Resnik (1999) und ein auf den Eigenschaften, welche Konzepte in der Ontologie miteinander verbinden, basierenden Maß.

[11] Wenn kein Dokument im System mit Konzept c annotiert ist, dann ist auch das Dokument d nicht mit Konzept c annotiert, hieraus ergibt sich der Fall für a(c) = 0

Alle drei Maße haben die Gemeinsamkeit, dass um die Ähnlichkeit zwischen zwei Konzepten berechnen zu können diese in derselben Ontologie vorkommen müssen. Für das hier vorgestellte System ist dies der Fall, alle Konzepte stammen aus derselben Ontologie zum Thema Requirements Engineering (vgl. Abschnitt 7.2). Soll mit einem dieser Maße die Ähnlichkeit zwischen zwei Konzepten aus unterschiedlichen Ontologien berechnet werden, kann wie in (Ziegler u. a., 2006) vorgegangen werden. Dort wird eine zusätzliche Klasse `Super_Thing` eingeführt, welche die direkte Superklasse aller `owl:Thing`-Klassen bildet. Bei `owl:Thing` handelt es sich um die Superklasse aller Klassen in einer Ontologie welche mittels OWL formalisiert wurde. Durch `Super_Thing` werden somit vormals getrennte Klassenhierarchien miteinander verbunden.

Alternativ dazu können für Anwendungsfälle, in denen mehrere Ontologien Verwendung finden sollen, andere Maße zur Berechnung der Ähnlichkeit zwischen Konzepten genutzt werden. In diesem Fall eignen sich Maße, wie sie derzeit für das Alignment bzw. Mapping (Euzenat u. Shvaiko, 2007) von Ontologien eingesetzt werden und somit dafür entworfen wurden die Ähnlichkeit von Konzepten aus unterschiedlichen Ontologien zu ermitteln.

Das Shortest-Path-Maß

Eines der verwendeten Maße um die Ähnlichkeit zwischen zwei Konzepten aus einer Ontologie zu berechnen basiert auf der Anzahl der Kanten im kürzesten Pfad zwischen zwei Konzepten in einer gemeinsamen Klassenhierarchie. Hierbei handelt es sich um ein symmetrisches Ähnlichkeitsmaß. Zur Berechnung des Ähnlichkeitsmaßes wird das SimPack-Framework (Bernstein u. a., 2005) (Ziegler u. a., 2006) verwendet.

Die Ähnlichkeit zwischen zwei Konzepten wird wie in Gleichung 7.12 beschrieben berechnet. Das verwendete Ähnlichkeitsmaß basiert auf der Anzahl der Kanten im kürzesten Pfad zwischen zwei Konzepten. Die so berechnete Pfadlänge wird um 1 erhöht und invertiert. Handelt es sich bei den beiden Konzepten um dasselbe Konzept, hat der Pfad die Länge 0. In diesem Fall ist die berechnete semantische Ähnlichkeit = 1.

$$sim(c_1, c_2) = \frac{1}{sp(c_1, c_2) + 1} \tag{7.12}$$

Wobei gilt:

- c_1 ... erstes Konzept
- c_2 ... zweites Konzept
- sp ... Anzahl der Kanten im kürzesten Pfad zwischen c_1 und c_2

Abhängig von den Charakteristika, welche eine Ontologie aufweist, können unterschiedliche Maße zur Berechnung der Ähnlichkeit zwischen zwei Konzepten Verwendung finden. Das Shortest-Path-Maß wurde gewählt, da eine charakteristische Eigenschaft der verwendeten Ontologie die taxonomische Hierarchie aus Konzepten darstellt.

Das Maß von Resnik (1999)

Ein weiteres verwendetes Maß zur Berechnung der Ähnlichkeit zwischen zwei Konzepten aus einer Ontologie ist eine von Resnik (1999) vorgeschlagenen Variante des von Wu u. Palmer (1994) vorgestellten Maßes. Hierbei handelt es sich ebenfalls um ein symmetrisches Ähnlichkeitsmaß. Die Implementierung des Maßes erfolgte durch Mitarbeiter der *Fondazione Bruno Kessler*[12], einem Partner im APOSDLE-Projekt.

Die Ähnlichkeit zwischen zwei Konzepten wird wie in Gleichung 7.13 beschrieben berechnet. Das verwendete Ähnlichkeitsmaß basiert auf der Anzahl der Knoten im Pfad vom Wurzelkonzept (hier `owl:Thing`) der gemeinsamen Klassenhierarchie zum *kleinsten gemeinsamen Vorfahren* (*least common subsumer - lcs*) zweier Konzepte, also dem spezifischten Konzept, welches sich die beiden Klassen als Superklasse teilen. Die so berechnete Pfadlänge wird durch die Summe der Anzahl der Knoten der Pfade von beiden Konzepten zum Wurzelkonzept normiert. Bei der Ermittlung der Pfadlänge wird der Start- und der Endknoten mitgezählt. Ein Pfad der nur aus einem Knoten besteht, beinhaltet einen Knoten, somit gilt $depth(\text{Wurzelkonzept}) = 1$.

$$sim(c_1, c_2) = \frac{2 \cdot depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} \tag{7.13}$$

[12] `http://www.fbk.eu/irst/` (18.05.2008)

Wobei gilt:

- c_1 ... erstes Konzept
- c_2 ... zweites Konzept
- *lcs* ... kleinster gemeinsamer Vorfahre beider Konzepte
- *depth* ... Tiefe eines Konzepts in der Klassenhierarchie (Anzahl der Knoten im Pfad vom Wurzelkonzept zum Konzept)

Wie das Shortest-Path-Maß wird das von Resnik (1999) vorgestellte Maß gewählt, da eine charakteristische Eigenschaft der verwendeten Ontologie die taxonomische Hierarchie der Konzepte darstellt. Ansätze zur Berechnung der semantischen Ähnlichkeit auf Basis einer Klassenhierarchie teilen sich die Problematik, dass es sich bei dem Abstand zwischen zwei Klassen in der Hierarchie nicht immer um dieselbe (semantische) Distanz handelt. Dies wird durch den von Resnik (1999) vorgestellten Ansatz adressiert, indem der kleinste gemeinsamer Vorfahre in das Maß einfließest.

Beispiele für die Berechnung der semantischen Ähnlichkeit mit dem Shortest-Path-Maß und dem Maß von Resnik (1999)

Im Folgenden werden Beispiele für die Berechnung der semantischen Ähnlichkeit basierend auf dem Shortest-Path-Maß und dem Maß von Resnik (1999) angeführt. Beide Maße berechnen die Ähnlichkeit zwischen Konzepten basierend auf deren Position in der Klassenhierarchie. Zur Berechnung der folgenden Beispiele wird die Klassenhierarchie aus Abbildung 7.3 verwendet. Diese stellt einen Auszug aus der im APOSDLE-System verwendeten Ontologie dar.

Beispiele für die Berechnung der Ähnlichkeit mit dem Shortest-Path-Maß

Beispiel 1: Berechnung der Ähnlichkeit zwischen dem Konzept *Use Case Model* und dem Konzept *Context Model*: Der kürzeste Pfad vom Konzept *Use Case Model* zum Konzept *Context Model* ist jener über das Konzept *Model*. In diesem Pfad befinden sich 2 Kanten. Hieraus folgt nach Gleichung 7.12 eine semantische Ähnlichkeit von $\frac{1}{2+1} \approx 0,33$.

Beispiel 2: Berechnung der Ähnlichkeit zwischen dem Konzept *Use Case Model* und dem Konzept *Use Case Diagram*: Der kürzeste Pfad vom Konzept *Use Case Model* zum Konzept *Use Case Diagram* ist jener über die

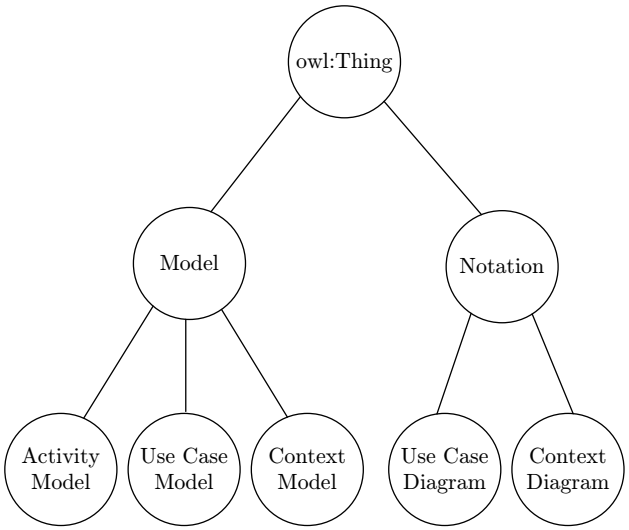

Abbildung 7.3: Auszug aus der Klassenhierarchie der in der ersten Version des APOSDLE-Systems verwendeten Ontologie

Konzepte *Model*, *owl:Thing* und *Notation*. In diesem Pfad befinden sich 4 Kanten. Hieraus folgt nach Gleichung 7.12 eine semantische Ähnlichkeit von $\frac{1}{4+1} = 0,2$.

Beispiel 3: Berechnung der Ähnlichkeit zwischen dem Konzept *Model* und dem Konzept *Notation*: Der kürzeste Pfad vom Konzept *Model* zum Konzept *Notation* ist jener über das Konzept *owl:Thing*. In diesem Pfad befinden sich 2 Kanten. Hieraus folgt nach Gleichung 7.12 eine semantische Ähnlichkeit von $\frac{1}{2+1} = \approx 0,33$.

Beispiele für die Berechnung der Ähnlichkeit mit dem Maß von Resnik (1999)

Beispiel 1: Berechnung der Ähnlichkeit zwischen dem Konzept *Use Case Model* und dem Konzept *Context Model*: Der kleinste gemeinsame Vorfahre des Konzepts *Use Case Model* und des Konzepts *Context Model* ist das Konzept *Model*. Das Konzept *Use Case Model* befindet sich in einer Tiefe von 3 in der Hierarchie, das Konzept *Context Model* befindet sich in einer Tiefe von 3 in der Hierarchie und das Konzept *Model* befindet sich in einer Tiefe von 2 in der Hierarchie. Hieraus folgt nach Gleichung 7.13 eine semantische Ähnlichkeit von $\frac{2 \cdot 2}{3+3} = 0,67$.

Beispiel 2: Berechnung der Ähnlichkeit zwischen dem Konzept *Use Case*

Model und dem Konzept *Use Case Diagram*: Der kleinste gemeinsame Vorfahre des Konzepts *Use Case Model* und des Konzepts *Use Case Diagram* ist das Konzept *owl:Thing*. Das Konzept *Use Case Model* befindet sich in einer Tiefe von 3 in der Hierarchie, das Konzept *Use Case Diagram* befindet sich in einer Tiefe von 3 in der Hierarchie und das Konzept *owl:Thing* befindet sich in einer Tiefe von 1 in der Hierarchie. Hieraus folgt nach Gleichung 7.13 eine semantische Ähnlichkeit von $\frac{2 \cdot 1}{3+3} = 0,33$.

Beispiel 3: Berechnung der Ähnlichkeit zwischen dem Konzept *Model* und dem Konzept *Notation*: Der kleinste gemeinsame Vorfahre des Konzepts *Model* und des Konzepts *Notation* ist das Konzept *owl:Thing*. Das Konzept *Model* befindet sich in einer Tiefe von 2 in der Hierarchie, das Konzept *Notation* befindet sich in einer Tiefe von 2 in der Hierarchie und das Konzept *owl:Thing* befindet sich in einer Tiefe von 1 in der Hierarchie. Hieraus folgt nach Gleichung 7.13 eine semantische Ähnlichkeit von $\frac{2 \cdot 1}{2+2} = 0,5$.

An den zuvor angeführten Beispielen wird das zentrale Charakteristikum des Maßes von Resnik (1999) deutlich. Dieses liefert eine höhere semantische Ähnlichkeit zwischen zwei Konzepten, je tiefer deren kleinster gemeinsame Vorfahre in der Hierarchie vorkommt. Dies wird am Vergleich der Beispiele 1 und 3 für die beiden Ähnlichkeitsmaße deutlich.

Das Trento-Ähnlichkeitsmaß

Das Trento-Ähnlichkeitsmaß[13] ermittelt die Ähnlichkeit zwischen Konzepten basierend auf den Relationen in der Ontologie, welche Konzepte miteinander verbinden. Zwei Konzepte welche durch eine Eigenschaft miteinander verbunden sind, werden als ähnlich angesehen. Je mehr Eigenschaften zwei Konzepte miteinander verbinden, je ähnlicher werden die beiden Konzepte aufgefasst.

Um die Ähnlichkeit zwischen Konzepten, basierend auf den Eigenschaften, welche sie verbinden zu berechnen, findet ein Vektor-basiertes Ähnlichkeitsmaß Verwendung. Konzepte werden hierzu als Vektoren repräsentiert und Eigenschaften werden Merkmale der Konzeptvektoren. Je-

[13] Dieses Maß ist im Rahmen eines Forschungsaufenthalts am FBK-irst in Trento entstanden und wurde vom Autor mitentwickelt und implementiert

de Relation in der Ontologie wird durch eine Dimension im Vektorraum repräsentiert, welcher durch die Konzeptvektoren aufgespannt wird. Die Ähnlichkeit zwischen zwei Vektoren wird auf Basis des Kosinusmaß (vgl. Gleichung 5.4 in Abschnitt 5.2) berechnet. Je mehr Merkmale sich zwei Vektoren teilen, desto kleiner ist der Winkel zwischen den beiden Vektoren und desto ähnlicher werden diese erachtet. Somit sind zwei Konzepte umso ähnlicher, je mehr Eigenschaften sie sich teilen.

Auch hier baut die Implementierung des Maßes auf dem SimPack-Toolkit (Bernstein u. a., 2005) auf, welches Klassen für die Repräsentation von Vektoren zur Verfügung stellt und das Kosinusmaß implementiert. Das Befüllen der Vektoren basierend auf dem entwickelten Verfahren wurde selbst implementiert.

Gleichung 7.14 zeigt die Repräsentation eines Konzeptes als Vektor. Gleichung 7.15 zeigt die Berechnung der semantischen Ähnlichkeit auf Basis des Kosinusmaßes.

$$\vec{c} = (d_1, d_2, \ldots, d_n) \tag{7.14}$$

Wobei gilt:

- $\vec{c}$ ... Vektor der das Konzept c repräsentiert
- d_n ... n-te Dimension des Vektors der das Konzept c repräsentiert
- n ... Anzahl der Eigenschaften in der Ontologie[14]
- $d_n = 0$ wenn Konzept c weder Domain noch Range der Eigenschaft n ist
- $d_n = 1$ wenn Konzept c Domain oder Range (oder beides) der Eigenschaft n ist

$$sim(c_1, c_2) = sim_{cos}(\vec{c_1}, \vec{c_2}) = \frac{\vec{c_1} \cdot \vec{c_2}}{|\vec{c_1}| \cdot |\vec{c_2}|} \tag{7.15}$$

[14] Im verwendeten Formalismus OWL sind Eigenschaften nicht, wie in der objekt-orientierten Programmierung, Teile von Klassen. Stattdessen wird eine Klasse als möglicher Ausgangspunkt (domain) bzw. Endpunkt (range) einer Eigenschaft festgelegt.

Wobei gilt:

- c_1 ... erstes Konzept
- c_2 ... zweites Konzept
- $\vec{c_1}$ ... Vektor der das erstes Konzept repräsentiert
- $\vec{c_2}$ ... Vektor der das zweite Konzept repräsentiert
- $sim_{cos}(\vec{c_1}, \vec{c_2})$ ... Ähnlichkeit zwischen den beiden Vektoren $\vec{c_1}$ und $\vec{c_2}$ basierend auf dem Kosinusmaß (vgl. Gleichung 5.4 in Abschnitt 5.2)

Das Trento-Maß wurde als Alternative zu den beiden Maßen gewählt, welche die semantische Ähnlichkeit basierend auf der Position zweier Konzepte in einer gemeinsamen Klassenhierarchie berechnen (Shortest-Path, Resnik). Das Trento-Maß baut auf den Eigenschaften einer Ontologie auf und nutzt somit zur Bestimmung der semantischen Ähnlichkeit ein anderes Charakteristikum der Wissensrepräsentation als die beiden zuvor vorgestellten Maße.

Beispiele für die Berechnung der semantischen Ähnlichkeit mit dem Trento-Maß

Anhand der Beispielontologie in Abbildung 7.4 wird im Folgenden das Trento-Maß beschrieben. Diese stellt einen Auszug aus der im APOSDLE-System verwendeten Ontologie dar.

Für die Berechnung der Ähnlichkeit zwischen Konzepten werden diese als Vektoren repräsentiert. Bei den Dimensionen der Vektoren handelt es sich um die Eigenschaften, welche in der Ontologie verwendet werden. In der Ontologie in Abbildung 7.4 sind fünf verschiedene Eigenschaften vorhanden. Hierbei handelt es sich um: *usesNotation*, *isAnElementOf*, *isUsedForCreation*, *isUsedForDescribing* und *isPartOf*. Somit wird zur Berechnung der semantischen Ähnlichkeit zwischen den Konzepten in dieser Ontologie ein Vektorraum mit fünf Dimensionen aufgespannt. Die Elemente eines Vektors können, wie in Gleichung 7.14 beschrieben, entweder den Wert 0 oder 1 beinhalten. Ein Element eines Konzeptvektors wird dann 1, wenn die zu diesem Element korrespondierende Eigenschaft das Konzept als Domain oder als Range hat. Anderenfalls wird das Element mit dem Wert 0 gesetzt.

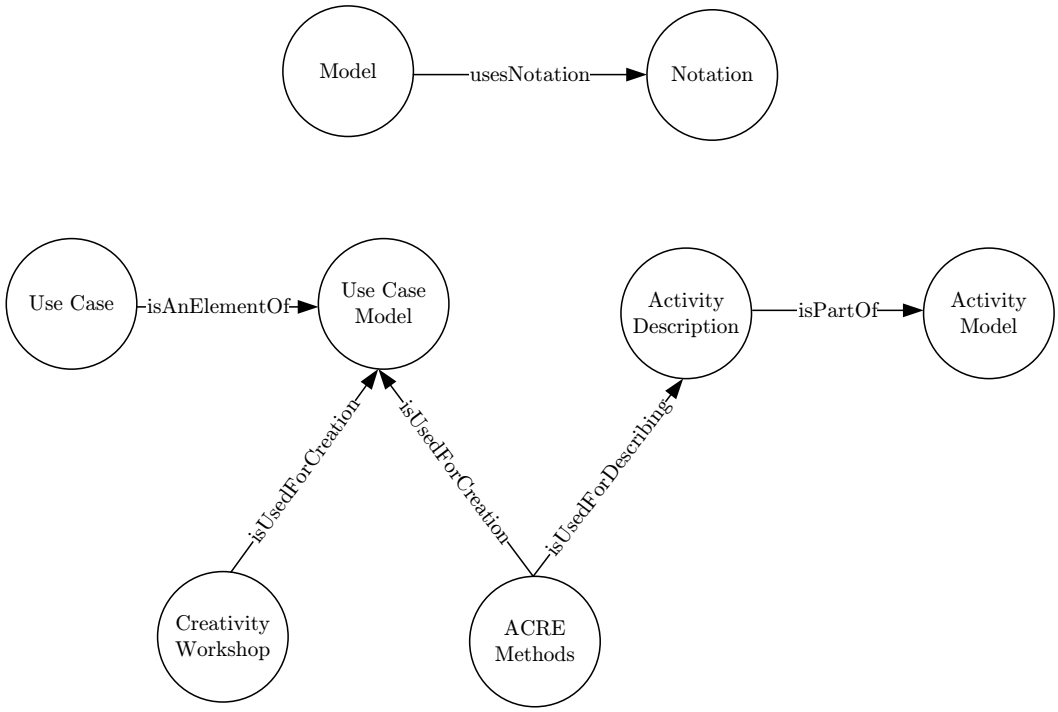

Abbildung 7.4: Auszug aus der Menge an Eigenschaften der in der ersten Version des APOSDLE-Systems verwendeten Ontologie

Für die in Abbildung 7.4 angeführten Konzepte ergeben sich unter Verwendung der Eigenschaften aus Abbildung 7.4 die folgenden Vektorrepräsentationen:

- *Model*: $[1, 0, 0, 0, 0]$
- *Notation*: $[1, 0, 0, 0, 0]$
- *Use Case*: $[0, 1, 0, 0, 0]$
- *Use Case Model*: $[0, 1, 1, 0, 0]$
- *Creativity Workshop*: $[0, 0, 1, 0, 0]$
- *ACRE Methods*: $[0, 0, 1, 1, 0]$
- *Activity Description*: $[0, 0, 0, 1, 1]$
- *Activity Model*: $[0, 0, 0, 0, 1]$

Wobei gilt:

- Dimension 1: *usesNotation*

- Dimension 2: *isAnElementOf*
- Dimension 3: *isUsedForCreation*
- Dimension 4: *isUsedForDescribing*
- Dimension 5: *isPartOf*

Es folgen drei Beispiele für die Berechnung der semantische Ähnlichkeit anhand des Trento-Maßes. Diese wird wie in Gleichung 7.15 beschreiben berechnet.

Beispiel 1: Berechnung der Ähnlichkeit zwischen dem Konzept *Model* und dem Konzept *Notation*: Für die Konzepte *Model* und *Notation* folgt nach Gleichung 7.15 eine semantische Ähnlichkeit von $\frac{[1,0,0,0,0]\cdot[1,0,0,0,0]}{|[1,0,0,0,0]|\cdot|[1,0,0,0,0]|} = 1$.

Beispiel 2: Berechnung der Ähnlichkeit zwischen dem Konzept *Model* und dem Konzept *Use Case Model*: Für die Konzepte *Model* und *Use Case Model* folgt nach Gleichung 7.15 eine semantische Ähnlichkeit von $\frac{[1,0,0,0,0]\cdot[0,1,0,0,0]}{|[1,0,0,0,0]|\cdot|[0,1,0,0,0]|} = 0$.

Beispiel 3: Berechnung der Ähnlichkeit zwischen dem Konzept *Use Case* und dem Konzept *Use Case Model*: Für die Konzepte *Use Case* und *Use Case Model* folgt nach Gleichung 7.15 eine semantische Ähnlichkeit von $\frac{[0,1,0,0,0]\cdot[0,1,1,0,0]}{|[0,1,0,0,0]|\cdot|[0,1,1,0,0]|} \approx 0,71$.

7.5.3 Bestimmung der text-basierten Ähnlichkeit zwischen Dokumenten

Als Ähnlichkeitsmaß für Textdokumente wird im vorgestellten System ein asymmetrisches Maß verwendet. Dieses basiert auf dem Vektorraummodell (vgl. Abschnitt 5.2) und wurde anhand der open-source Suchmaschine Lucene[15] umgesetzt. Die Funktionalität der Ähnlichkeitsbestimmung zwischen textuellen Inhalten wird in der Komponente KnowMiner gekapselt (vgl. Abschnitt 7.6). Die Ähnlichkeit zwischen zwei Textdokumenten wird wie in Gleichung 7.16 beschrieben berechnet.

$$sim(d1, d2) = score(d1_{25}, d2) \tag{7.16}$$

[15] `http://lucene.apache.org/` (18.05.2008)

Wobei gilt:

- $d1$ ... Dokumentvektor des ersten Dokuments
- $d2$ ... Dokumentvektor des zweiten Dokuments
- $d1_{25}$ ... Dokumentvektor des ersten Dokuments aus dem alle Termgewichte entfernt wurden, außer die 25 größten Termgewichte (die 25 Terme die am häufigsten im Dokument vorkommen)
- $score(d1_{25}, d2)$ ... Ähnlichkeitsfunktion von Lucene, siehe Gleichung 7.17

Um die 25 Terme mit den höchsten Gewichten zu extrahieren werden sowohl Terme im Titel als auch im Dokumentinhalt verwendet. Der modifizierte Dokumentvektor $d1_{25}$ wird als Anfragevektor für eine Suche mit Lucene verwendet. Die Suche mit $d1_{25}$ retourniert die ähnlichsten Dokumente zum Anfragevektor, also zum aktuellen Dokument. Für jedes Suchergebnis wird ein Ähnlichkeitswert (*score*) mittels Lucene berechnet. Dieser Wert gibt Aufschluss über die Relevanz eines Suchergebnisses zur Anfrage und ermöglicht eine Reihung der Suchergebnisse.

Die Berechnung des *score*-Wertes durch Lucene wird in Gleichung 7.17 beschreiben. Eine detaillierte Erläuterung der verschiedenen Parameter, welche zur Adaptierung der Berechnungen des *score*-Wertes von Lucene verwendete werden können, kann in der Javadoc-Beschreibung der Klasse `org.apache.lucene.search.Similarity`[16] gefunden werden.

$$\begin{aligned} score(q,d) &= coord(q,d) \cdot queryNorm(q) \\ &\quad \cdot \sum_{t\_in\_q} \Big[tf(t\_in\_d) \cdot idf(t)^2 \cdot t.getBoost() \cdot norm(t,d) \Big] \end{aligned} \tag{7.17}$$

Wobei gilt:

- q ... Anfragevektor
- d ... Dokumentvektor

[16] `http://lucene.apache.org/java/2_2_0/api/org/apache/lucene/search/Similarity.html` (18.05.2008)

- $coord(q,d) = \frac{numberOfMatchingTerms}{numberOfQueryTerms}$
- $numberOfMatchingTerms$ ... Anzahl der Terme im Dokument welche mit der Anfrage übereinstimmen
- $numberOfQueryTerms$ ... Anzahl der Terme in der Anfrage
- $queryNorm(q)$ ... Normalisierung des Anfragevektors; Voreinstellung von Lucene werden verwendet
- $tf(t\_in\_d)$ ... Termfrequenz des aktuellen Terms im Dokument; Voreinstellung von Lucene werden verwendet
- $idf(t)$ ... Inverse Dokumenthäufigkeit des aktuellen Terms in der Dokumentenmenge; Voreinstellung von Lucene werden verwendet
- $t.getBoost() = tf(t\_in\_q) \cdot idf(t)$
- $tf(t\_in\_q)$ ... Termfrequenz des aktuellen Terms in der Anfrage
- $norm(t,d) = \frac{1}{\sqrt{(numberOfDocumentTerms)}}$
- $numberOfDocumentTerms$ ... Anzahl der Terme im aktuellen Dokument

Für die Bestimmung der textuellen Ähnlichkeit wird mit einem Maß basierend auf dem Vektorraummodell und der Suchmaschine Lucene auf erprobte Verfahren zur Bestimmung von textueller Ähnlichkeit zurückgegriffen.

7.6 Softwarearchitektur

Im Zuge der vorliegenden Arbeit wird eine Suchsystem für den Semantic Desktop entwickelt, welches auf Basis von Assoziativem Retrieval operiert. Im Folgenden werden die Komponenten des entwickelten Systems beschrieben. Im ersten Schritt wird ein Überblick über die Basiskomponenten des entwickelten Systems gegeben (Abschnitt 7.6.1). Darauf aufbauend wird speziell auf jene Komponenten eingegangen, welche für den Aufbau des Assoziativen Netzes Verwendung finden (Abschnitt 7.6.2).

7.6.1 Basiskomponenten

Abbildung 7.5 zeigt die Basiskomponenten des entwickelten Systems, diese dienen zum Aufbau, und zur Verwaltung des Netzes, zur Suche im Netz und

zur Kontrolle des Systems.

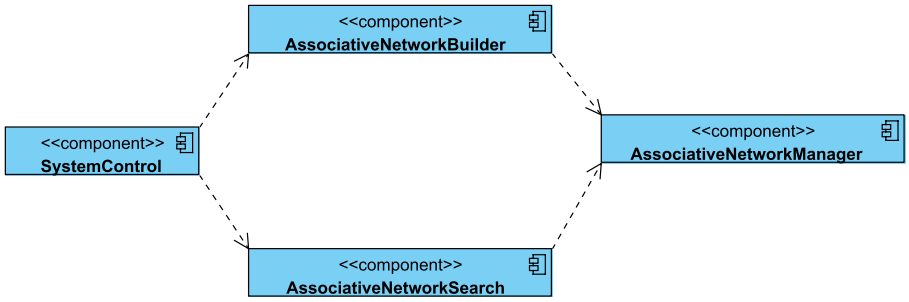

Abbildung 7.5: Basiskomponenten des entwickelten Systems

Im Folgenden werden die Komponenten aus Abbildung 7.5 im Detail beschrieben:

`SystemControl:` Diese Komponente dient zur Kontrolle der Komponenten im Suchsystem. Mit Hilfe dieser Komponente wird der Aufbau des Assoziativen Netzes angestoßen (`AssociativeNetworkBuilder`) und eine Suche initiiert (`AssociativeNetworkSearch`).

`AssociativeNetworkSearch:` Diese Komponente dient zur Suche im Assoziativen Netz. Ausgehend von einer Anfrage werden jene Knoten im Netz aktiviert, welche zur Anfrage korrespondieren. In weiterer Folge breitet sich Aktivierung, ausgehend von den initial aktivierten Knoten, im Netz aus. Schließlich werden Ergebnisse zur Anfrage retourniert, hierbei handelt es sich um die Menge der nach der Aktivierungsausbreitung aktivierten Knoten. Abbildung 6.5 und Abbildung 6.7 zeigen den Ablauf einer Suche im Assoziativen Netz als Aktivitätsdiagramm. Eine detaillierte Beschreibung der Suche ist in Abschnitt 7.4.3 zu finden.

`AssociativeNetworkManager:` Diese Komponente dient zur Verwaltung der Datenstruktur, mit welcher das Assoziative Netz repräsentiert wird. Mit Hilfe dieser Komponente ist es möglich auf die Topologie des Netzes zuzugreifen bzw. diese zu modifizieren. Wird das Assoziative Netz erstellt, werden mit Hilfe dieser Komponente neue Knoten und Kanten im Netz erstellt. Durch die Kapselung der Funktionalität zur Verwaltung der Netzstruktur

ist es möglich das Netz unterschiedlich abzulegen. So ist alternativ zur Verwaltung in einer Datenbank, wie es derzeit der Fall ist, auch die Verwaltung direkt im Arbeitsspeicher, in Textdateien, oder in einem Indexdienst denkbar.

`AssociativeNetworkBuilder:` Diese Komponente stellt Funktionalität zur Verfügung um das Assoziative Netz zu erzeugen. Dieser Schritt ist mit dem Erstellen eines Suchindexes in klassischen Retrieval-Systemen vergleichbar und wird technisch ähnlich realisiert. Das Assoziative Netz wird in der Form von Arrays of Edges (Sedgewick u. Schidlowsky, 2003) gespeichert, welche in einer Datenbank abgelegt sind. Jene Spalten in der Datenbank, in denen die Knoten der Kanten gespeichert sind, werden indiziert. Somit handelt es sich beim verwendeten Ansatz um eine Variante der Speicherung in der von Form Adjazenzlisten, dem empfohlenen Ansatz für die Repräsentation von dünnen Graphen (Sedgewick u. Schidlowsky, 2003). Eine Diskussion dieses Ansatzes ist in (Scheir u. Lindstaedt, 2006)) zu finden.

Auf die Datenstruktur zur Verwaltung des Netzes wird in den Abschnitten 7.6.1 und 7.7.1 eingegangen. Auf die Komplexität des verwendeten Ansatzes wird in Abschnitt 7.7.3 eingegangen.

7.6.2 Komponenten für den Aufbau des Assoziativen Netzes

Für den Aufbau des Netzes wird auf weitere Komponente zurückgegriffen. Abbildung 7.6 zeigt die Zusammenhänge zwischen den Komponenten, welche für den Aufbau des Assoziativen Netzes verwendet werden, im Detail. Abbildung 7.7 zeigt den Aufbau des Assoziativen Netzes als Aktivitätsdiagramm.

Im Folgenden werden die Komponenten aus Abbildung 7.6 im Detail beschrieben:

`DocumentManager:` Diese Komponente stellt eine Liste mit allen sich im System befindenden Dokumenten zur Verfügung. Über diese Liste wird iteriert und für jedes dieser Dokumente wird ein Knoten im Assoziativen Netz erzeugt. In weiterer Folge erfolgt eine Ähnlichkeitsberechnung zwischen jedem Dokument und jedem anderen Dokument auf Basis des Inhaltes. Dieses Funktionalität wird von der Komponente `TextbasedSimiliarity` zur Verfügung gestellt.

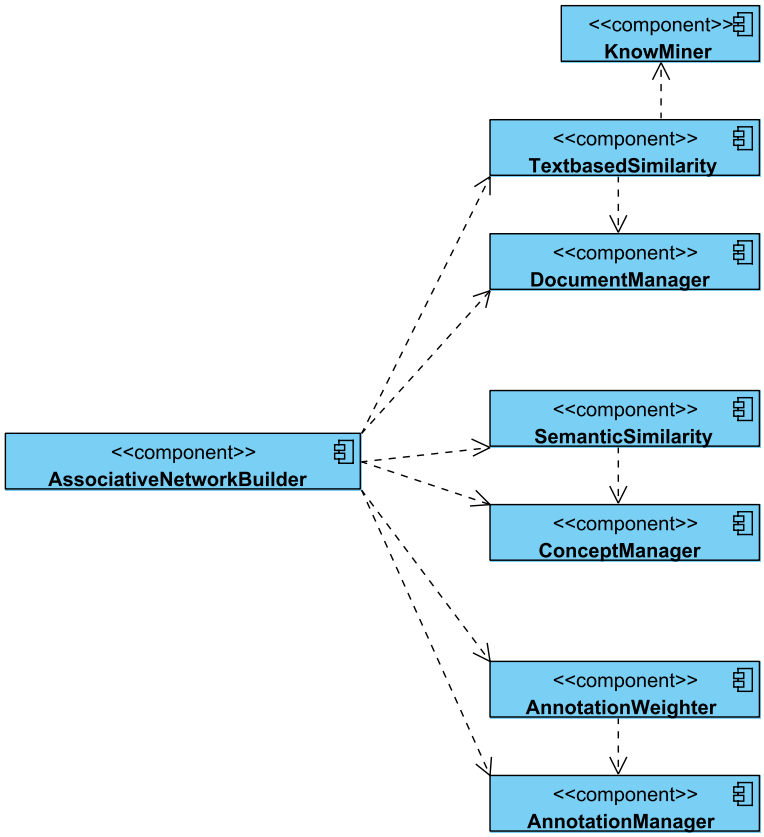

Abbildung 7.6: Komponenten welche für den Aufbau des Assoziativen Netzes Verwendung finden

`TextbasedSimilarity:` Dieser Komponente dient zur Bestimmung der text-basierten Ähnlichkeit zwischen zwei Dokumenten und basiert auf Funktionalität, die von der Komponente `KnowMiner` zur Verfügung gestellt wird. In Abschnitt 7.5.3 wird auf den verwendeten Ansatz zur text-basierten Ähnlichkeitsberechnung im Detail eingegangen.

`KnowMiner:` Auf der Basis dieser Komponente erfolgt die Ähnlichkeitsberechnung zwischen zwei Dokumenten. Dazu werden im ersten Schritt jene Textdokumente, zwischen denen die Ähnlichkeit berechnet werden soll, indiziert. In weitere Folge wird die Ähnlichkeit auf Basis des Vektorraummodells (vgl. Abschnitt 5.2) bestimmt. Eine detaillierte Erklärung des verwendeten Ansatzes zur Ähnlichkeitsbestimmung ist in

Abschnitt 7.5.3 zu finden. Das verwendete *KnowMiner*-Framework ist eine generisches Framework zur Wissenserschließung, es wird im Detail in Granitzer (2006) beschrieben.

`ConceptManager:` Diese Komponente verwaltet die Konzepte der aktuell im System verwendeten Ontologie. Der `ConceptManager` ermöglicht eine Auflistung der verfügbaren Konzepte und das Laden einer oder mehrerer Ontologien. Ähnlich zu den Dokumenten im System wird für jedes Konzept ein Knoten im Assoziativen Netz erzeugt. Auch wird zwischen jedem Konzept und jedem anderen Konzept die Ähnlichkeit bestimmt und diese Information für die Erzeugung von Kanten im Assoziativen Netz verwendet. Die Funktionalität der Ähnlichkeitsbestimmung wird von der Komponente `SemanticSimilarity` bereit gestellt.

`SemanticSimilarity:` Diese Komponente dient zur Bestimmung der semantischen Ähnlichkeit zwischen zwei Konzepten aus einer Ontologie. In Abschnitt 7.5.2 wird auf den verwendeten Ansatz zur semantischen Ähnlichkeitsberechnung im Detail eingegangen.

`AnnotationManager:` Um schließlich Dokumentknoten und Konzeptknoten miteinander verbinden zu können wird die Information benötigt welche Dokumente mit welchen Konzepten annotiert sind. Diese Information wird von der Komponente `AnnotationManager` bereitgestellt. Diese verwaltet die Annotationen von Dokumenten mit Konzepten und ermöglicht das Hinzufügen und Löschen von Annotationen zu Dokumenten.

Annotationen von Dokumenten mit Konzepten werden gewichtet. Hierzu wird die Funktionalität der Komponente `AnnotationWeighter` verwendet.

`AnnotationWeighter:` Diese Komponente gewichtet die Annotationen von Dokumenten mit Konzepten. Hierfür wird ein auf tf*idf basierendes (vgl. Abschnitt 5.2.2) Maß verwendet. Jene Konzepte, mit denen sehr viele unterschiedliche Dokumente annotiert sind, werden niedriger gewichtet, da sie schlechtere Diskriminatoren zwischen den Dokumenten sind. Der Ansatz zur Gewichtung der Annotationen wird in Abschnitt 7.5.1 beschrieben.

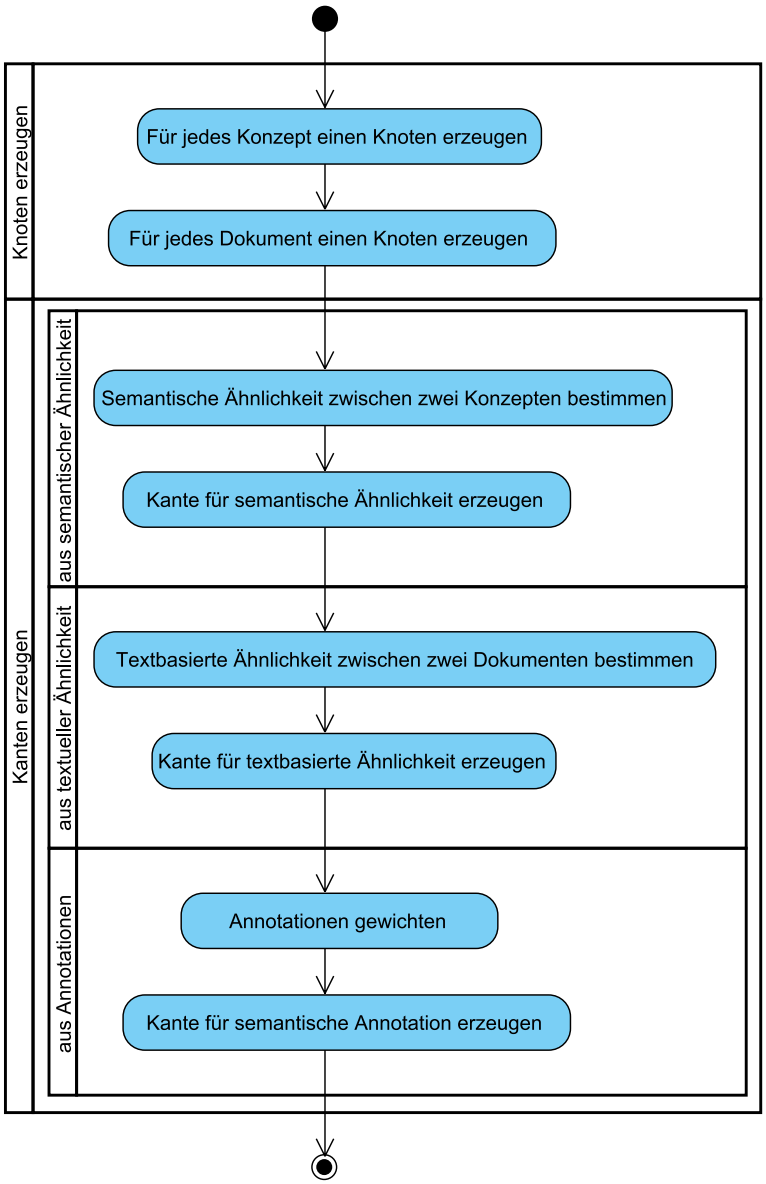

Abbildung 7.7: Aufbau des Assoziativen Netzes

7.7 Details zur Realisierung des Prototypen

Im Folgenden wird auf Details zur Realisierung des hier vorgestellten Systems eingegangen. In Abschnitt 7.7.1 wird die für die Umsetzung verwendete Software angeführt. In Abschnitt 7.7.2 wird die Indizierung der Netzstruktur

beschrieben.

7.7.1 Für die Umsetzung des Systems verwendete Software

Für die Umsetzung des hier vorgestellten Systems wird Java[17] in der Version 1.6 verwendet. Das Spring Framework[18] diente dazu die Komponenten, welche in Abschnitt 7.6 beschrieben wurden, miteinander zu verbinden und gleichzeitig deren funktionale Kapselung sicher zu stellen. Hierfür fand das Dependency Injection Pattern (Fowler, 2004) Verwendung.

Die Datenstruktur des Assoziativen Netzes wird in einer MySQL Datenbank in der Version 5.0.27 gespeichert. Zur Ablage des Netzes in der Datenbank werden 3 Tabellen verwendet, welche jeweils die Kanten zwischen den zwei unterschiedlichen Knotentypen beinhalten:

- c2c ... Eine Kante zwischen einem Konzept- und einem Konzeptknoten
- c2d ... Eine Kante zwischen einem Konzept- und einem Dokumentknoten
- d2d ... Eine Kante zwischen einem Dokument- und einem Dokumentknoten

Für die Extraktion von Termen aus Dokumenten und die Verwaltung von Term-Dokumentpaaren in einem Suchindex wird das KnowMiner-Framework (Granitzer, 2006) verwendet. Dieses nutzt intern die Suchmaschine Lucene für den Aufbau und die Verwaltung des Suchindexes.

7.7.2 Im System verwendete Indizes

Um die Zugriffszeit auf die zu suchenden Dokumente zu optimieren, nutzen Suchmaschinen üblicherweise einen Suchindex (Baeza-Yates u. Ribeiro-Neto, 1999). Hierbei werden bei Systemen zur schlüsselwortbasierten Suche Dokumente, die Terme welche in ihnen vorkommen und deren Frequenz in der Form eines invertierten Indexes gespeichert: Zu jedem Term im System wird die Menge aller Dokumente in denen der Term vorkommt und dessen Häufigkeit im Dokument gespeichert. Die Liste an Termen wird indiziert

[17] `http://java.sun.com/` (18.05.2008)
[18] `http://www.springframework.org/` (18.05.2008)

um den Zugriff auf die Dokumente zu beschleunigen, in denen der jeweilige Term vorkommt.

Das hier vorgestellte System verfolgt einen ähnlichen Ansatz zur Beschleunigung der Suche. Eine Anfrage im hier vorgestellten System enthält eine Menge an Konzepten und retourniert jene Dokumente, welche von diesen Konzepten handeln. Die Konzept-Dokument-Paare bzw. die Kanten zwischen Konzept- und Dokumentknoten, werden in einer Tabelle in einer Datenbank abgespeichert (vgl. Abschnitt 7.7.1). Jene Spalte welche den Ausgangspunkt der Kante beinhaltet, also die Konzeptknoten enthält, wird indiziert. Für Verbindungen zwischen Konzepten und Konzepten wird derselbe Ansatz verfolgt. Auch hier wird die Spalte mit den Konzepten von denen eine Kante ausgeht indiziert. Selbiges erfolgt mit Kanten zwischen Dokumenten und Dokumenten. Dieser Ansatz zur Optimierung der Suchgeschwindigkeit wird im Detail in Abschnitt 7.7.3 beschrieben.

Des Weiteren erfolgt eine Extraktionen und Indizierung der Terme, welche in den sich im System befindenden Dokumenten vorkommen. Diese Terme werden für die Berechnung der Ähnlichkeit von Dokumenten im System verwendet (siehe Abschnitt 7.5.3). Für diese Funktionalität wird auf die Suchmaschine Lucene zurückgegriffen. Diese wird durch das KnowMiner-Framework gekapselt, welches zusätzlich für die Extraktion der Terme aus den Dokumenten verantwortlich zeichnet und das Speichern der Term-Dokumentpaare im Lucene-Index übernimmt.

7.7.3 Komplexität des Suchansatzes

Kosten von Algorithmen werden in Form von *Komplexität* (vgl. (Cormen u. a., 2004)) beschrieben. Im Folgenden wird auf die *Platzkomplexität* (*space complexity*) und die *Zeitkomplexität* (*time complexity*) der Implementierung des hier verwendeten Verfahrens zur Aktivierungsausbreitung eingegangen. Hierfür wird auf die *O-Notation* (vgl. (Cormen u. a., 2004)) zurückgegriffen. Diese dient zur Angabe der oberen Schranke des Laufzeitverhaltens bzw. des Speicherverbrauchs von Algorithmen. Der verwendete Spreading-Activation-Ansatz kann somit weniger Ressourcen brauchen als hier angeführt, allerdings nicht mehr.

Abhängig von den jeweiligen Eigenschaften des verwendeten Spreading-Activation-Ansatzes und dessen Implementierung können unterschiedliche

Kosten entstehen. Hier sei auf die Ausführungen in (Preece, 1981) hingewiesen, der die Kosten seines Ansatzes beschreibt. Ebenfalls finden sich Angaben zur Komplexität der verwendeten Spreading-Activation-Ansätze in (Rocha u. a., 2004a), (Tsatsaronis u. a., 2007), (Liu, 2007) und (Gelgi u. a., 2005).

Die Datenstruktur des Assoziativen Netzes wird in einer Datenbank gehalten (vgl. Abschnitt 7.7.1). Für die Aktivierungsausbreitung werden Teile dieser Datenstruktur aus der Datenbank in den Hauptspeicher geladen und dort verarbeitet. Dieser Ansatz bietet den Vorteil, dass nicht die komplette Datenstruktur des Netzes im Hauptspeicher gehalten und beim Start des Systems geladen werden muss. In dieser Hinsicht ist das hier vorgestellt System äquivalent zu klassischen Retrieval-Systemen welchen auf einem invertieren Index aufbauen, welcher als Datei(en) im File-System abgelegt ist und im Hauptspeicher verarbeitet wird.

Die Platzkomplexität des vorgestellten Ansatzes muss somit für zwei Orte, die Datenbank, in der die Netzstruktur gehalten wird, und den Hauptspeicher, in dem die Verarbeitung des Netzes erfolgt, beschrieben werden. Auch die Zeitkomplexität des vorgestellten Ansatzes setzt sich aus elementaren Operationen auf diesen beiden Datenstrukturen zusammen.

Kosten für den Aufbau der Netzstruktur in der Datenbank

Die Netzstruktur wird in der Form von mehren Tabellen in einer MySQL Datenbank gespeichert (vgl. Abschnitt 7.7.1). Für jeden Kantentyp existiert eine eigene Tabelle, pro Kante im Netz wird eine Zeile in der Datenbank angelegt. Es handelt sich hierbei um einen Spezialfall der Speicherung als *array of edges* (Sedgewick u. Schidlowsky, 2003) bzw. um drei Arrays von Kanten (eine Tabelle pro Kantentyp).

Sedgewick u. Schidlowsky (2003) gehen auf die Worst-Case-Kosten von Operationen auf dieser Datenstruktur ein, Tabelle 7.1 gibt einen Auszug.

Für Tabelle 7.1 gilt:

- E ... Anzahl der Kanten

Um die Zugriffszeit auf die Netzstruktur in der Datenbank zu verringern, werden die verwendeten Tabellen indiziert (vgl. Abschnitt 7.7.2). Dieser Index optimiert den Zugriff auf Kanten im Netz, um ausgehend von einem

Speicherverbrauch	$O(E)$
Initialisierung der Datenstruktur	$O(1)$
Löschen der Datenstruktur	$O(1)$
Kante hinzufügen	$O(1)$
Kante finden	$O(E)$
Kante löschen	$O(E)$

Tabelle 7.1: Kosten der Speicherung als array of edges (nach (Sedgewick u. Schidlowsky, 2003))

Knoten über dessen Kanten dessen Nachbarknoten identifizieren zu können, wie es für die Weitergabe von Aktivierung notwendig ist. Die Indizes werden von MySQL als B-Baum (MySQL AB, 2008) realisiert. Der B-Baum zu einer Tabelle hält jene Knoten des Netzes, auf welche aus dieser Tabelle referenziert wird. Aus einem Knoten im Baum wird auf jene Zeilen in der Datenbank (Kanten im Netz) verwiesen, in welchen der korrespondierende Knoten im Netz als Ausgangspunkt vorkommt. Aus der Verwendung eines B-Baums leiten sich die Größe des Index und die Zeit für die Suche im Index ab.

Die Speicherung des Assoziativen Netzes in der Form eines B-Baums, welcher auf Zeilen in einer Tabelle verweist, ist zur Speicherung des Netzes als Adjazenzliste deren Knoten mittels eines B-Baums indiziert sind äquivalent (vgl. (Sedgewick u. Schidlowsky, 2003)).

Tabelle 7.2 führt die Worst-Case-Kosten an, die sich durch diesen Ansatz der Speicherung ergeben. Die Kosten für den B-Baum sind aus (Cormen u. a., 2004) übernommen[19].

[19]Herbei handelt es sich um eine Abschätzung, da von Seitens MySQL keine Aussagen über die Komplexität ihrer Implementierung verfügbar sind

	B-Baum	Tabelle	Gesamt
Speicherverbrauch	$O(n)$	$+ O(E)$	$= O(V + E)$
Initialisierung der Datenstruktur	$O(1)$	$+ O(1)$	$= O(1)$
Löschen der Datenstruktur	$O(1)$	$+ O(1)$	$= O(1)$
Kante hinzufügen	$O(t \log_t n)$	$+ O(1)$	$= O(t \log_t V)$
Kante finden	$O(t \log_t n)$	$+ O(1)$	$= O(t \log_t V)$
Kante löschen	$O(t \log_t n)$	$+ O(1)$	$= O(t \log_t V)$

Tabelle 7.2: Kosten der Speicherung als array of edges und der Indizierung (nach (Sedgewick u. Schidlowsky, 2003) und (Cormen u. a., 2004))

Für Tabelle 7.2 gilt:

- n ... Anzahl der Schlüssel des Baumes
- $n = V$
- V ... Anzahl der Knoten in der Tabelle
- E ... Anzahl der Kanten in der Tabelle
- t ... Tiefe des Baumes

Funktion 1 zeigt den Pseudocode für den Aufbau der Netzstruktur in der Datenbank. Hierbei wird für alle Kanten vom jeweiligen Kantentyp (Konzept-Konzept, Konzept-Dokument, Dokument-Dokument) eine Zeile in der korrespondierenden Tabelle in der Datenbank angelegt. Basierend auf den elementaren Kosten für die verwendeten Operationen, welche in Tabelle 7.2 angeführt werden, ergibt sich eine Platzkomplexität von $O(V + E)$ und eine Zeitkomplexität von $O(E \cdot t \log_t V)$.

```
1: function NetzstrukturAufbauen
2:     Tabelle anlegen
                          ▷ Für jede Kante im Netz vom jeweiligen Kantentyp
3:     while Kanten verfügbar do
4:         Kante in Tabelle einfügen
5:     end while
6: end function
```

Funktion 1: Netzstruktur in der Datenbank aufbauen

Kosten für Aktivierungsausbreitung im Hauptspeicher

Für die Verwaltung von Knoten des Assoziativen Netzes im Hauptspeicher wird die Java-Klasse `java.util.Hashtable` verwendet. Die Komplexität von Operationen mit dieser Datenstruktur ist in Tabelle 7.3 zu finden. Die Kosten für die Operationen mit der Hash-Tabelle sind aus (Cormen u. a., 2004) übernommen[20]. Hierbei ist darauf hinzuweisen, dass für die Operationen *Knoten hinzufügen*, *Knoten finden* und *Knoten löschen* im Durchschnitt $O(1)$ gilt und $O(n)$ nur bei einer Hash-Kollision der Fall ist.

Speicherverbrauch	$O(n)$
Initialisierung der Datenstruktur	$O(1)$
Löschen der Datenstruktur	$O(n)$
Knoten hinzufügen	$O(n)$
Knoten finden	$O(n)$
Knoten löschen	$O(n)$

Tabelle 7.3: Kosten der Speicherung als Java Hashtable (nach (Cormen u. a., 2004))

Für Tabelle 7.3 gilt:

- n ... Anzahl der Schlüssel in der Hash-Tabelle
- $n = V$
- V ... Anzahl der Knoten in der Hash-Tabelle

Funktion 2 zeigt die Initiierung der Suche und deren Ablauf basierend auf dem Prozess der Aktivierungsausbreitung (vgl. Funktion 3) im Pseudocode. Für jedes Konzept in der Anfrage wird ein Knoten im Hauptspeicher erzeugt und initial aktiviert (vgl. Abschnitt 6.3.6 für die exakte Berechnung der initialen Aktivierung). In weiterer Folge wird die Aktivierungsausbreitung von den initial aktivierten Knoten zu ihren Nachbarn durch den Aufruf der Funktion `Aktivierungsausbreitung` initiiert.

[20] Herbei handelt es sich um eine Abschätzung, da von Seitens Java keine Aussagen über die Komplexität der Implementierung verfügbar sind

```
1: function SUCHE(anfrage)
2:     Knoten zur Anfrage im Hauptspeicher erzeugen
3:     Knoten zur Anfrage initial aktivieren
4:     Knoten zur Anfrage zu Liste der aktivierten Knoten hinzufügen
5:     return AKTIVIERUNGSAUSBREITUNG(Liste der aktivierten Knoten)
6: end function
```

Funktion 2: Suche

Funktion 3 zeigt den Pseudocode für die Ausbreitung von Aktivierung im Hauptspeicher. Vor der Aktivierungsausbreitung werden jene Knoten aus der Datenbank im Hauptspeicher angelegt zu denen sich die Aktivierung ausbreiten soll. Auf dieser Datenstruktur findet schließlich die Ausbreitung von Aktivierung statt (vgl. Abschnitt 6.3.6 für die exakte Berechnung der Aktivierungsausbreitung). Dieser Vorgang wird so lange wiederholt wie Schritte der Aktivierungsausbreitung durchgeführt werden sollen.

```
1:  function AKTIVIERUNGSAUSBREITUNG(Liste der aktivierten Knoten)
2:      Eingabeliste = Liste der aktivierten Knoten
3:      for Anzahl der Schritte do
4:          Hash-Tabelle leeren
5:          for Anzahl der Knoten in Eingabeliste do
6:              Nachbarknoten zu aktiviertem Knoten aus Datenbank holen      ▷ Suche in Datenbank
7:              for Anzahl von Nachbarknoten aus Datenbank do
8:                  if Nachbarknoten existiert noch nicht im Hauptspeicher then   ▷ Suche in Hashtable
9:                      Nachbarknoten im Hauptspeicher erzeugen
10:                     Nachbarknoten zu Hash-Tabelle hinzufügen          ▷ Einfügen in Hashtable
11:                 end if
12:                 Aktivierung an Nachbarknoten weiter geben
13:             end for
14:         end for
15:         Eingabeliste = Knoten in der Hash-Tabelle
16:     end for
17:     return Eingabeliste
18: end function
```

Funktion 3: Aktivierungsausbreitung

Basierend auf den elementaren Kosten für die verwendeten Operationen welche in Tabelle 7.2 und in Tabelle 7.3 angeführt werden, ergibt sich für die Suche basierend auf Aktivierungsausbreitung eine Platzkomplexität von $O(Q \cdot M^D)$ im Hauptspeicher und eine Zeitkomplexität von $Q \cdot M^{D-1} \cdot O(t \log_t n) + Q \cdot M^D \cdot O(n)$.

Wobei gilt:

- Q ... Anzahl der Elemente in der Anfrage
- M ... Maximale Anzahl von Nachbarknoten eines Knoten
- D ... Tiefe der Suche (Anzahl der Iterationen)
- n ... Anzahl der Schlüssel des Baumes
- $n = V$
- V ... Anzahl der Knoten in der Tabelle
- t ... Tiefe des Baumes

Die Platzkomplexität resultiert aus der Tatsache, dass jeder Knoten, welcher aktiviert wird (die Knoten zur Anfrage eingeschlossen) im Speicher angelegt wird. Werden ausschließlich die Knoten zur Anfrage angelegt ergibt dies eine Platzkomplexität von $O(Q)$. Breitet sich die Aktivierung von den initial aktivierten Anfrageknoten zu deren Nachbarn aus resultiert daraus eine Platzkomplexität von $O(Q \cdot M)$. Für einen weiteren Schritt der Ausbreitung wird draus $O(Q \cdot M \cdot M)$. Für den allgemeinen Fall gilt somit die Platzkomplexität von $O(Q \cdot M^D)$.

Für die Zeitkomplexität können ähnliche Überlegungen getroffen werden. Die Zeit für das Anlegen von Knoten im Hauptspeicher zur Anfrage ist $Q \cdot O(n)$. Für die nachfolgende Aktivierungsausbreitung müssen erst in der Datenbank die Nachbarknoten zu den Anfrageknoten gesucht werden und in weiterer Folge die Ergebnisse dieser Suche im Hauptspeicher als Knoten angelegt werden. Erst dann kann die Weitergabe von Aktivierung zwischen den Knoten im Hauptspeicher erfolgen.

Die Zeitkomplexität für die Suche von Nachbarknoten eines Knoten in der Netzstruktur, welcher in der Datenbank gespeichert ist, beträgt $O(t \log_t n)$, für alle Nachbarknoten zur Anfrage $Q \cdot O(t \log_t n)$. Für das Anlegen der Nachbarknoten im Hauptspeicher beträgt die Zeitkomplexität $Q \cdot M \cdot O(n)$. Hieraus ergibt sich ein Zeitkomplexität von $Q \cdot O(t \log_t n) +$

$Q \cdot M \cdot O(n)$ für den ersten Schritt der Aktivierungsausbreitung. Für den zweiten Schritt der Aktivierungsausbreitung beträgt die Zeitkomplexität $Q \cdot M \cdot O(t \log_t n) + Q \cdot M \cdot M \cdot O(n)$. Im allgemeinen Fall gilt somit $Q \cdot M^{D-1} \cdot O(t \log_t n) + Q \cdot M^D \cdot O(n) = Q \cdot M^{D-1} \cdot (O(t \log_t n) + M \cdot O(n))$.

Geht man von der Annahme aus, dass $Q \cdot M^{D-1}$ eine vernachlässigbare Konstante ist, gilt $O(t \log_t n) + M \cdot O(n)$. Des Weiteren gilt unter der Annahme, dass t und M vernachlässigbare Konstanten sind $O(\log_t n) + O(n) = O(n)$ für die Zeitkomplexität der Aktivierungsausbreitung. Diese Annahme kann dann getroffen werden, wenn $n >> Q$, $n >> M$, $n >> D$ und $n >> t$ ist. Von diesem Fall kann bei der Berechnung der oberen Schranke des Algorithmus ausgegangen werden, da Q, M, D bzw. t im Vergleich zu n vernachlässigbar klein ist. n ist gleich der Anzahl der Knoten im Netz (bzw. in einer Schicht des Netzes) und somit gleich der Anzahl der Dokumente im System oder der Konzepte in der Ontologie. n ist in typischen Anwendungsfällen deutlich größer als die Anzahl der Elemente in der Anfrage (Q), die maximale Anzahl von Nachbarknoten eines Knoten (M), die Anzahl der Iterationen (D) und die Tiefe des Baumes (t).

7.8 Zusammenfassung

In diesem Abschnitt wurde jenes System vorgestellt, welches auf Basis des in Kapitel 6 präsentierten Modells realisiert wurde. Es wurde der Anwendungsfall des Systems (APOSDLE) vorgestellt und auf die Erzeugung und Speicherung der vom System verwendeten semantischen Annotationen eingegangen. Ebenfalls wurde auf die Erzeugung und Verarbeitung der dem System zu Grunde liegenden Netzstruktur eingegangen, das verwendete Gewichtungsschema für die semantischen Annotationen vorgestellt und die verwendeten Ähnlichkeitsmaße beschrieben. Zum Abschluss erfolgte die Beschreibung der Software-Architektur des entwickelten Systems, es wurde auf Details der Implementierung eingegangen und es erfolgte eine Untersuchung der Zeit- und Platzkomplexität des verwendeten Spreading-Activation-Ansatzes.

8 Evaluierung

8.1 Einleitung

In dieser Arbeit wird ein Information-Retrieval-Modell für das Semantic Web entwickelt, welches am Desktop umgesetzt und erprobt wird. Zur Evaluierung des vorgestellten Systems wird auf erprobte Standardmethodiken aus dem Feld des Information Retrieval zurückgegriffen. Im Folgenden wird nach einer Einführung in das Thema Evaluierung im Information Retrieval und der Vorstellung von Standardmaßen aus dem Information Retrieval, die Evaluierung des vorgestellten Systems anhand dieser Standardmaße beschrieben.

8.2 Evaluierung im Information Retrieval

Wie Baeza-Yates u. Ribeiro-Neto (1999) feststellen, werden Information-Retrieval-Systeme meist automatisiert, unter Laborbedingungen evaluiert. Bei dieser Art der Evaluierung wird eine Menge an Anfragen an ein Information-Retrieval-System gesendet, welches Mengen von Ausgabewerten zurückgibt. Die Menge der auf eine Anfrage retournierten Ergebnisse und die Menge der sich im System befindlichen Dokumente werden zur Evaluierung des Systems herangezogen. Auf Basis dieser beiden Mengen (beziehungsweise ihrer Teilmengen an relevanten Dokumenten zu einer Anfrage) können Maße wie *Recall* (zu Deutsch *Vollständigkeit*) und *Precision* (zu Deutsch *Genauigkeit*) berechnet (vgl. Gleichung 8.1 und Gleichung 8.2) werden.

Buckley u. Voorhees (2004) stellen fest, dass der vorherrschenden Forschungsansatz für die Entwicklung von Information-Retrieval-Systemen das *Cranfield-Paradigma* ist. Hierbei wird auf eine Dokumentensammlung (*document collection*) und Mengen an Anfragen zurückgegriffen um unterschiedliche Retrieval-Ansätze miteinander zu vergleichen. Eine grundlegende Annahme des Cranfield-Paradigmas ist, dass die Relevanzbewertungen (relevance judgements) *vollständig* sind, d.h. jedes Dokument auf dessen Relevanz für jede Anfrage bewertet wurde.

Fuhr (2006) stellt fest, dass während die Precision für einen Benutzer eines Information-Retrieval-Systems direkt ablesbar ist, der Recall weder für einen Benutzer direkt erkennbar, noch mit vernünftigem Aufwand präzise zu bestimmen ist. Der Grund hierfür liegt darin, dass zur Bestimmung des Recall alle sich im System befindlichen Dokumente auf deren Relevanz zur Anfrage bewertet werden müssen. Buckley u. Voorhees (2004) untersuchen aus dem oben genannten Grund mehrere Retrieval-Maße auf ihre *Stabilität*, d.h. wie konstant eine Aussage *System A ist besser als System B* bleibt, wenn keine vollständige Bewertung aller sich im System befindenden Dokumente zu allen Anfragen vorhanden ist.

Die Arbeiten von Buckley u. Voorhees (2000, 2004) haben entschieden dazu beigetragen die Evaluierung im Information Retrieval, besonders im Bereich großer Korpora, noch weiter zu entwickeln. Sie inspirierten die Entwicklung neuer Evaluierungsmaße, die eine höhere Stabilität als klassische Maße wie etwa Precision nach 10, Mean Average Precision aber auch das von Buckley u. Voorhees (2004) vorgestellte Maß bpref aufweisen. Zwei dieser Maße sind *Inferred Average Precision* (*infAP*) (Yilmaz u. Aslam, 2006) und *Estimated Mean Average Precision* (εMAP) (Carterette u. a., 2006). Die Bedeutung dieser neuen Maße lässt sich daraus erkennen, dass infAP nur kurze Zeit nach der Veröffentlichung in `trec_eval`[1] integriert wurde und εMAP den Best Paper Award auf der renommierten SIGIR 2006 Konferenz[2] erhielt.

[1] trec_eval ist das Standardwerkzeug für die Auswertung der Ergebnisse von Systemen, welche bei der Text REtrieval Conference eingereicht wurden

[2] SIGIR steht für Special Interest Group on Information Retrieval, die ACM SIGIR Konferenz zählt zu den größten und kompetitivsten Konferenzen im Feld Information Retrieval

8.2.1 Standardmaße

Recall: Recall ist als die Anzahl der bei einer Suche gefunden relevanten Dokumente[3] im Verhältnis zur Anzahl aller sich im System befindlichen relevanten Dokumente definiert.

$$\text{Recall} = \frac{|\{R_E\}| \cap |\{E\}|}{|\{R_S\}|} \tag{8.1}$$

Wobei gilt:

- R_E ... relevante Dokumente im Ergebnis
- E ... gefundene Dokumente
- R_S ... relevante Dokumente im System

Precision: Precision ist als die Anzahl der gefunden relevanten Dokumente zu allen bei der Suche gefunden Dokumente definiert.

$$\text{Precision} = \frac{|\{R_E\}| \cap |\{E\}|}{|\{E\}|} \tag{8.2}$$

Wobei gilt:

- R_E ... relevante Dokumente im Ergebnis
- E ... gefundene Dokumente

Precision nach n: Precision nach n gefunden Dokumenten (P(n)) basiert auf der Anzahl der relevanten Dokumente in den n höchstgereihten Suchergebnissen zu einer Anfrage. Diesem Maß liegt die Annahme zugrunde, dass sich Benutzer nur für die höchstgereihten Suchergebnisse interessieren, wie es in Web-Umgebungen der Fall ist (Buckley u. Voorhees, 2004).

[3] an dieser Stelle kann anstatt Dokument auch allgemeiner Informationsobjekt oder Datensatz verwendet werden

$$P(n) = \frac{|\{R_{E_n}\}| \cap |\{E_n\}|}{n} \tag{8.3}$$

Wobei gilt:

- R_{E_n} ... relevante Dokumente in den ersten n Dokumenten im Ergebnis
- E_n ... ersten n gefundenen Dokumente
- $n \in \mathbb{N}_1$

Mean (Uninterpolated) Average Precision: *Average Precision* (*AP*) ist der Durchschnittswert der Precision-Werte an der Stelle der relevanten Dokumente in der Ergebnisliste, welche auf eine Anfrage retourniert wurde. Für relevante Dokumente im System, welche nicht gefunden werden, wird ein Precision-Wert von 0 verwendet. Gleichung 8.4 zeigt die Berechnung der Average Precision.

$$\text{AP} = \frac{1}{|\{R_S\}|} \sum_{d \epsilon E} P(d) \cdot rel(d) \tag{8.4}$$

Wobei gilt:

- R_S ... relevante Dokumente im System
- d ... Position in der Ergebnismenge
- E ... gefundene Dokumente
- $P(d)$ ... Precision nach d
- $rel(d)$ ... 1 wenn Dokument an Position d relevant, 0 sonst

Mean Average Precision (*MAP*) ist der Mittelwert der Average Precision Werte über alle Anfragen an das System. MAP ist äquivalent zur Fläche unter einer nicht interpolierten Recall-Precision-Kurve (Buckley u. Voorhees, 2004). Laut Buckley u. Voorhees (2004) basiert MAP auf mehr Information als P(10) und liefert somit stabilere Ergebnisse im Vergleich unterschiedlicher Retrieval-Systeme, wie er in (Buckley u. Voorhees, 2000) durchgeführt wurde.

infAP: Maße wie P(10) und MAP unterscheiden nicht zwischen jenen Dokumenten in der Dokumentensammlung, welche für eine Anfrage *nicht relevant* sind und jenen Dokumenten, welche *nicht* auf ihre Relevanz für eine Anfrage *bewertet* wurden. Nicht beurteilte Dokumente werden in den zuvor vorgestellten Maßen als nicht relevant angenommen.

In dieser Eigenschaft unterscheidet sich das Maß *infAP* (*Inferred Average Precision*) (Yilmaz u. Aslam, 2006). infAP wird wie Average Precision berechnet, allerdings wird statt der Precision eines Dokuments an Stelle k in der Ergebnisliste auf die vermuteten (expected) Precision eines Dokuments zurückgegriffen. Gleichung 8.5 zeigt die Berechnung der vermuteten (expected) Precision eines Dokuments an Position k. Gleichung 8.5 leitet sich aus Gleichung 8.6 und Gleichung 8.7 her.

$$E[\text{precission at rank } k] = \frac{1}{k} \cdot 1 + \frac{k-1}{k} \left(\frac{|d100|}{k-1} \cdot \frac{|rel| + \epsilon}{|rel| + |nonrel| + 2\epsilon} \right) \tag{8.5}$$

Wobei gilt:

- k ... Stelle in der Ergebnisliste
- $d100$ ... Dokumente im depth-100 Pool
- rel ... relevante Dokumente im depth-100 Pool
- $nonrel$ ... nicht relevante Dokumente im depth-100 Pool
- ϵ ... kleiner Wert um Division durch 0 zu verhindern (*Lidstone smoothing*)

Der depth-100 Pool ist die Menge der auf Relevanz bewerteten Dokumente für eine Anfrage. In TREC wird der depth-100 Pool aus den 100 höchstgereihten Suchergebnissen aller getesteten Systeme gebildet. In TREC-8 sind das 129 Systeme, der depth-100 Pool für alle 50 Anfragen besteht aus 86830 Relevanzbewertungen (Savell, 2005).

Yilmaz u. Aslam (2006) testen die Stabilität ihres Maßes infAP für den Fall, dass nur Teile der Relevanzbewertungen des depth-100 Pools vorhanden sind. Zu diesem Zweck berechnen sie den Kendall's τ Wert zwischen der Reihung der 129 Systeme, welche an TREC-8 teilgenommen haben, mit vollständigen Relevanzbewertungen im Vergleich zur Reihung für den Fall, dass nur Teile der Relevanzbewertungen abgegeben wurden.

Sie kommen zum Schluss, dass für den Fall von 25% aller möglichen Relevanzbewertungen des depth-100 Pools der Kendall's τ Wert der beiden Reihungen der 129 Systeme größer als 0,9 ist. Yilmaz u. Aslam (2006) gehen davon aus, dass zwei Reihungen mit einem Kendall's τ Wert von größer als 0,9 als gleichwertig angesehen werden können, hierbei verweisen sie auf Voorhees (2001). Die durch infAP erzielte Reihung mit unvollständigen Relevanzbewertungen weißt eine höhere Rangkorrelation zur Reihung mit vollständigen Relevanzbewertungen auf, als jene Reihungen die mit dem von Buckley u. Voorhees (2004) vorgestellte Maß bpref-10 mit der gleichen Menge an unvollständigen Relevanzbewertungen berechnet wurde. Zusätzlich weißt infAP einen höhere Übereinstimmung mit dem Wert von MAP auf als bpref-10.

Die vermutete (expected) Precision eines relevanten Dokuments an Position k in der Ergebnisliste kann wie in Gleichung 8.6 berechnet werden. Mit der Wahrscheinlichkeit $\frac{1}{k}$ handelt es sich um das aktuelle Dokument, dieses ist (per Definition) relevant. Mit der Wahrscheinlichkeit $\frac{k-1}{k}$ handelt es sich um ein Dokument, welches sich in der Ergebnisliste an einem höheren Rang befindet.

$$E[\text{precission at rank } k] = \frac{1}{k} \cdot 1 + \frac{k-1}{k} E[\text{precission above rank } k] \quad (8.6)$$

Wobei gilt:

- k ... Stelle in der Ergebnisliste
- $E[\text{precission above rank } k]$ ... Precision eines Dokuments vor Position k

Die vermutete (expected) Precision eines Dokuments vor Position k in der Ergebnisliste kann wie in Gleichung 8.7 berechnet werden. Mit der Wahrscheinlichkeit $\frac{|non-d100|}{k-1}$ wird ein Dokument gezogen, welches nicht in der Menge der zu bewertenden Dokumente (im depth-100 Pool) ist. Ein derartiges Dokument hat die Precision von 0. Mit der Wahrscheinlichkeit $\frac{|d100|}{k-1}$ wird eine Dokument gezogen, welches sich in der Menge der zu bewertenden Dokumente befindet. Die Wahrscheinlichkeit, dass dieses Dokument relevant ist, beträgt $\frac{|rel|}{|rel|+|nonrel|}$.

$$E[\text{precission above rank } k] = \frac{|non - d100|}{k-1} \cdot 0 + \frac{|d100|}{k-1} \cdot \frac{|rel|}{|rel| + |nonrel|} \tag{8.7}$$

Wobei gilt:

- k ... Stelle in der Ergebnisliste
- $d100$ ... Dokumente im depth-100 Pool
- $non - d100$ ... Dokumente die nicht im depth-100 Pool sind
- rel ... relevante Dokumente im depth-100 Pool
- $nonrel$ ... nicht relevante Dokumente im depth-100 Pool

infAP beruht auf dem Zufallsauswahlverfahren (random sampling). Der Wert von infAP ist mit dem der Average Precision (AP) ident, wenn alle Relevanzbewertungen vorhanden sind. infAP nutzt mehr der zur Verfügung stehenden Information als MAP (vgl. (Carterette u. a., 2006)) und kann dadurch eine höhere Stabilität erzielen. Das Hinzufügen von zusätzlichen, nicht bewerteten Dokumenten zur Dokumentensammlung verändert den Wert von infAP nicht.

8.3 Evaluierung des vorgestellten Systems

Die Evaluierung von Information-Retrieval-Systemen ist ein kritischer Aspekt der Information-Retrieval-Forschung. Neue Such-Paradigmen, wie die Suche im Semantic Web, stellen eine zusätzliche Herausforderung hinsichtlich der Evaluierung von Retrieval-Systemen dar, da keine fertigen Testkorpora zur Evaluierung dieser System existieren.

In diesem Abschnitt wird beschrieben wie ein Testkorpus für die Evaluierung des entwickelten Systems erstellt wird und in weiterer Folge Standardmaße aus dem Information Retrieval zur vergleichenden Bewertung unterschiedlicher Systemkonfigurationen Verwendung finden.

8.3.1 Die zur Evaluierung verwendete Datenbasis und der Testkorpus

Ein zentrales Hindernis für eine einfache Evaluierung von Information-Retrieval-Systemen, welche auf Technologien für das Semantic Web aufbau-

en, ist das Fehlen von standardisierten Testkorpora, wie sie für die Evaluierung von textbasierten Retrieval-System existieren. Aus diesem Grund wird ein eigener Testkorpus aufgebaut, welcher auf den verfügbaren Daten der ersten Version des APOSDLE-Systems (APOSDLE consortium, 2007) (vgl. Abschnitt 7.2) basiert. Der Datensatz setzt sich aus einer Wissensbasis in Form einer Ontologie und einer Menge an Dokumenten zusammen. Elemente aus der Wissensbasis werden dazu verwendet die Dokumente zu beschreiben. Diese Information wird in der Wissensbasis in Form von einer Teil-Ontologie gekapselt.

Die erste Version des APOSDLE-Systems wurde für die Domäne des *Requirements Engineering* entwickelt. Dies resultierte in einer Domänenontologie für dieses Feld und einer Menge an Dokumenten, welche verschiedene Themen aus dem Feld des Requirements Engineering aufbereiten. Dokumente, welche unterstützendes Material zum Thema Requirements Engineering enthalten, wie beispielsweise Definitionen, Beispiele oder Kurse, wurden zum Teil mit Konzepten aus der Domänenontologie annotiert.

Die Dokumentmenge wurde von einem Partner im APOSDLE-Projekt zur Verfügung gestellt, während die Ontologie von einem anderen Partner im Projekt mit Hilfe des Ontologie-Editors Protégé[4] modelliert wurde. Gemeinsam zeichnen die beiden Partner für die Annotation der Dokumente aus der Dokumentenmenge mit Konzepten aus der Ontologie verantwortlich.

Die verwendete Ontologie besteht aus 79 Konzepten, während die Dokumentenmenge 1016 Dokumente beinhaltet. 496 Dokumente sind mit einem oder mehreren Konzepten aus der Wissensbasis annotiert. 21 Konzepte aus der Domänenontologie werden für die Annotation von Dokumenten verwendet. Somit werden nur Teile der Ontologie für die Annotation von Dokumenten verwendet und nur eine Teilmenge der Dokumente ist mit Konzepten aus der Ontologie annotiert.

In ihrer Größe ist die hier verwendete Test-Collection mit Dokumentensammlungen, welche für frühere Information-Retrieval-Experimente Verwendung fanden, wie der Cranfield- oder der CACM-Collection[5], vergleichbar.

[4] `http://protege.stanford.edu/` (18.05.2008)
[5] `http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/` (18.05.2008)

Abgrenzung zu existierenden Test-Collections

Neben dem Fehlen standardisierter Testkorpora für die Evaluierung von Semantic-Web-Information-Retrieval-Systemen sind auch keine Korpora aus dem text-basierten Retrieval verfügbar, mit denen ein System, mit ähnlichen Charakteristika zu dem hier vorgestelltem, evaluierte werden könnte. Es wurde in Erwägung gezogen die Konzepte aus der Ontologie als gleichwertig zu Termen eines Text-Retrieval-Systems zu behandeln, um einen Standardkorpus verwenden zu können. Allerdings wäre hierzu eine Struktur nötig gewesen, welche die Terme untereinander verbindet, wie es bei den hier verwendeten Konzepten durch die Ontologie der Fall ist (vgl. Abschnitt 6.3.1). Hierfür hätte ein Thesaurus Verwendung finden können. Da diese Wissensrepräsentation allerdings unterschiedliche Charakteristika zur verwendeten Ontologie aufweist, und somit auch ein anderes Ähnlichkeitsmaß für die Berechnung der semantischen Ähnlichkeit verwendet hätte werden müssen, hätte diese zur Evaluierung eines System mit anderen Eigenschaften als denen des hier vorgestellten Systems geführt.

Ebenfalls wurde die *INitiative for the Evaluation of XML Retrieval* (*INEX*[6]) Test-Collection für die Evaluierung in Erwägung gezogen. INEX stellt eine Dokumentensammlung von XML-Dokumenten zur Verfügung, welche textuelle Information mit XML-Strukturinformation in Verbindung setzen. Allerdings fehlt in diesem Fall ebenfalls eine Ontologie, welche die Metadaten, die für die XML-Auszeichnungen verwendet werden, miteinander verbindet. Somit wäre es nicht möglich gewesen die Suche nach ähnlichen Konzepten, welche auf der Ontologie aufsetzt, zu verwenden bzw. zu evaluieren.

8.3.2 Für die Evaluierung verwendete Maße

Ein zentrales Problem mit klassischen Evaluierungs-Maßen des Information Retrieval wie *Recall* (vgl. Abschnitt 8.2.1) oder *Mean Average Precision* (vgl. Abschnitt 8.2.1) ist, dass für deren Berechnung vollständige Relevanzbewertungen vorhanden sein müssen. Das bedeutet, dass jedes Dokument im System auf dessen Relevanz für jede Anfrage an das System bewerten werden muss (Buckley u. Voorhees, 2004). Diese Problematik wird auch in

[6] `http://inex.is.informatik.uni-duisburg.de/` (18.05.2008)

der Literatur diskutiert. Fuhr (2006) geht auf die Größe Recall ein:

> *Die Größe des Recall ist dagegen für einen Benutzer weder erkennbar, noch kann sie mit vernünftigem Aufwand präzise bestimmt werden.*
>
> Fuhr (2006)

Carterette u. a. (2006) adressiert darauf aufbauend den aufwendigen Prozess der Erstellung von Testmengen für die Evaluierung:

> *Building sets large enough for evaluation of realworld implementations is at best inefficient, at worst infeasible.*
>
> Carterette u. a. (2006)

Auf Grund dieser Problematik werden für die Evaluierung des hier vorgestellten Systems Evaluierungsmaße verwendet, welche nicht die vollständige Relevanzbewertung aller Dokumente gegen jede Anfrage voraussetzen. Die Entscheidung fällt dabei auf Precision (P) an den Rängen 10, 20 und 30 der Ergebnismenge (vgl. Abschnitt 8.2.1) und infAP (Yilmaz u. Aslam, 2006) (vgl. Abschnitt 8.2.1), welches Average Precision durch ein Zufallsexperiment annähert. Während der Wert von infAP auf allen bewerteten Dokumenten (relevanten und nicht relevanten) basiert, werden für die Berechnung von P(10), P(20) und P(30) nur als relevant bewertete Dokumente in Betracht gezogen. Da für die Berechnung von infAP mehr Information herangezogen wird als für die Berechnung von P(10), P(20) und P(30), wird infAP als ein stabileres Maß angesehen (vgl. Abschnitt 8.2.1).

Um die Werte der Evaluierungsmaße zu berechnen wird das Programm `trec_eval`[7] verwendet, welches die Berechnung einer großen Menge an Standardmaßen für die Evaluierung von Information-Retrieval-Systemen erlaubt. Das Programm stammt von der Text REtrieval Conference (TREC) und wird dort für die Berechnung von Evaluierungswerten verwendet.

[7] `http://trec.nist.gov/trec_eval/` (18.05.2008)

8.3.3 Für die Evaluierung verwendete Anfragen

Zur Evaluierung eines Information-Retrieval-Systems unter Laborbedingungen (vgl. Abschnitt 8.2) werden Mengen von Anfragen an das System gesendet und die erhaltene Ergebnisse auf deren Relevanz zu den Anfragen beurteilt.

Die für die Evaluierung des hier vorgestellten Systems verwendeten Anfragen bestehen aus einer Menge an Konzepten aus der Domänenontologie: Die erste Version des APOSDLE-Systems stellt Wissensarbeitern Ressourcen zur Verfügung um ihnen zu ermöglichen eine bestimmte Kompetenz zu erwerben. Um die Funktionalität der Suche nach Ressourcen zu ermöglichen, welche eine bestimmte Kompetenz aufbauen, werden Kompetenzen durch Mengen an Konzepten aus der Domänenontologie ausgedrückt. Diese Konzeptmengen werden als Anfragen für die Suche nach Ressourcen verwendet.

Für die Evaluierung des hier vorgestellten Systems werden alle unterschiedlichen Mengen an Konzepten (unterschiedliche Kompetenzen können durch die gleiche Menge an Konzepten ausgedrückt werden), welche in der ersten Version des APOSDLE-System Verwendung finden, genutzt. Zusätzlich werden alle Konzepte des Domänenmodells, welche nicht bereits in der Menge an Anfragen (als Menge mit einem Element) vorhanden waren, in die Anfragemenge aufgenommen. Dies führt zu insgesamt 79 verschiedenen Anfragen, welche für die Evaluierung des Systems verwendet werden.

8.3.4 Sammlung und Auswertung der Relevanzbewertungen

12 verschiedene Konfigurationen des vorgestellten Systems werden getestet und basierend auf den vorgestellten Evaluierungsmaßen gegenüber gestellt. 79 unterschiedliche Anfragen, jeweils bestehend aus einer Menge an Konzepten, werden an jede Systemkonfiguration gesandt.

Für jede Anfrage und Systemkonfiguration werden die ersten 30 Ergebnisse in einer Tabelle in einer Datenbank gespeichert. Zu diesem Zweck wird eine Zeile für jedes Anfrage-Dokument-Paar verwendet. Wird ein Dokument zu einer Anfrage von mehr als einer Systemkonfiguration retourniert, wird das entsprechende Anfrage-Dokument-Paar nur einmal in der Tabelle in der Datenbank gespeichert.

Jene Anfrage-Dokument-Paare, welche nach allen Anfragen an alle Sys-

temkonfigurationen schlussendlichen in der Datenbank gespeichert sind, werden durch einen menschlichen Begutachter bewertet. Um eine potentielle Unschärfe durch mehrere Begutachter (Voorhees, 2000) zu vermeiden, werden alle Anfrage-Dokument-Paare durch dieselbe Person beurteilt. Der Begutachter war nicht direkt in die Erstellung der Kompetenz zu Konzept Zuordnungen, und somit in die Definition der Anfragen, involviert, allerdings ist das Verständnis für die Materie gegeben.

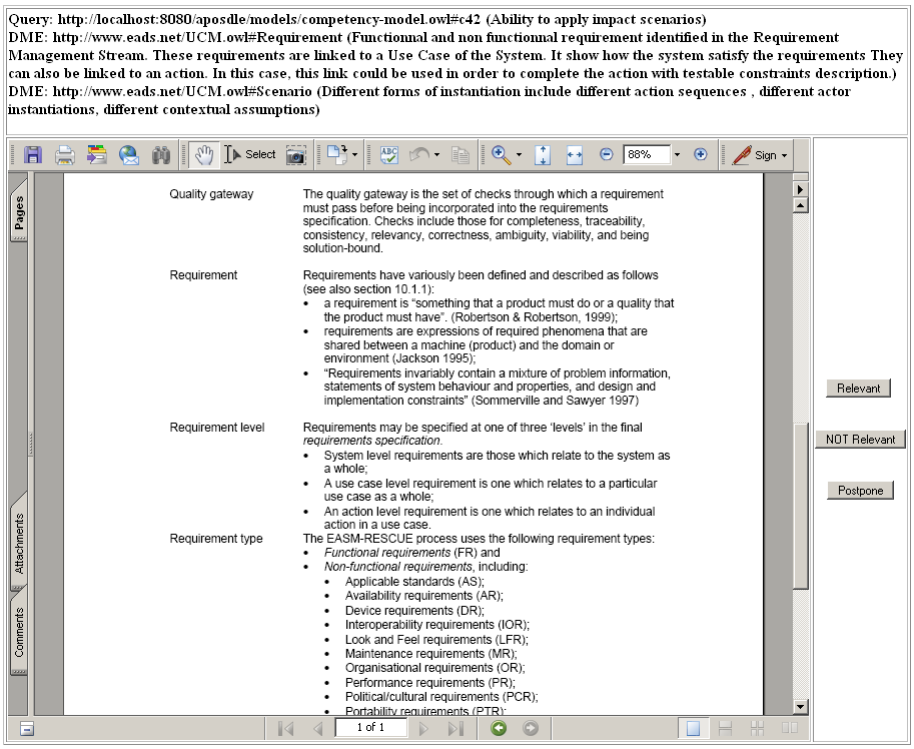

Abbildung 8.1: Die Web-Applikation, welche für die Sammlung von Relevanzbewertungen verwendet wird

Abbildung 8.1 zeigt die Web-Anwendung, welche für die Relevanzbewertung entwickelt wurde. Dem Begutachter wird ein Ergebnis und die Anfrage an das System, welches zu diesem Ergebnis führte, angezeigt. Er hat nun die Möglichkeit das Ergebnis als *relevant* oder als *nicht relevant* zu bewerten oder den Vorgang der Bewertung zu *verschieben*. In allen drei Fällen wird nach der Interaktion des Begutachter mit der Web-Anwendung das nächste zu bewertende Ergebnis angezeigt.

Nach der Relevanzbewertung aller Anfrage-Dokument-Paare in der Datenbank werden sowohl die Ergebnisse, welche von den verschiedenen Systemkonfigurationen erzielt werden, als auch die globalen Relevanzbewertungen in Textdateien gespeichert. Die Formatierung dieser Dateien erfolgt in einem durch das `trec_eval`-Programm interpretierbaren Format. In weiterer Folge werden mit Hilfe von `trec_eval` die Werte zu den Maßen P(10), P(20), (P30) und infAP für die verschiedenen Systemkonfigurationen berechnet.

8.3.5 Durch die Evaluierung ermittelte Reihung der Systemkonfigurationen

Tabelle 8.1 zeigt die Werte, welche jeweils für die Maße P(10), P(20), (P30) und infAP, für die 12 unterschiedlichen Systemkonfigurationen berechnet wurden. Die Spalten *SemSim*, *TxtSim* in Tabelle 8.1 zeigen an ob semantische bzw. textuelle Ähnlichkeit für die Suche Verwendung fand. In der Spalte zur semantischen Ähnlichkeit wird zusätzlich das verwendete Ähnlichkeitsmaß angeführt. Hierbei handelt es sich um das Shortest-Path-Maß, das Maß von Resnik (1999) und das Trento-Maß, welche in Abschnitt 7.5.2 vorgestellt wurden. Der Wert in Klammern in der Spalte *SemSim* steht für den Schwellwert, welcher für die assoziative Suche verwendet wird. *Shortest Path (0.4)* steht somit für das Shortest-Path-Maß aus Abschnitt 7.5.2, wobei für die assoziative Suche nur Konzepte herangezogen werden, deren semantische Ähnlichkeit größer als 0,4 zu den Konzepten in der Anfrage ist. Die verwendeten Schwellwerte wurden empirisch ermittelt.

Konfiguration 1 (Konf_1) repräsentiert die grundlegende Suchfunktionalität des vorgestellten Systems. Sie stellt die Baseline-Konfiguration des Systems dar, gegen welche alle anderen Konfigurationen verglichen werden. Diese Konfiguration ist zu einer Anfrage an eine Wissensbasis (beispielsweise mit SPARQL) gleichwertig, welche eine gereihte Liste an Ergebnisse retourniert. Genau jene Dokumente, welche mit den Konzepten aus der Anfrage annotiert sind bilden die Ergebnismenge. Ein idf-ähnliches Maß, zur Gewichtung der Annotationen (vgl. Abschnitt 7.5.1) von Dokumenten mit Konzepten, erlaubt die Reihung der Ergebnisse zu einer Anfrage.

Alle anderen Konfigurationen beziehen semantische und/oder textuelle Ähnlichkeit mit ein. Während die semantische Ähnlichkeit die Menge der Anfrage erweitert um somit indirekt die Menge der Ergebnisse zu vergrößern,

zielt die textuelle Ähnlichkeit direkt darauf ab die Ergebnismenge zu vergrößern. Die Konfigurationen 3 bis 12 nutzen semantische Ähnlichkeit, welche mit einem der in Abschnitt 7.5.2 beschriebenen Maßen berechnet wird. Die Konfigurationen 2 und 8 bis 12 nutzen textuelle Ähnlichkeit, welche mit dem in Abschnitt 7.5.3 beschriebenen Maß berechnet wird.

Alle Formen der assoziativen Suche, unter Einbeziehung von semantischer Ähnlichkeit (Konfigurationen 3 bis 7), textueller Ähnlichkeit (Konfiguration 2) oder beiden Formen der Ähnlichkeit (Konfigurationen 8 bis 12), erhöhen die Retrieval-Leistung im Vergleich zum Baseline-Ansatz. In jeder dieser Konfigurationen werden zusätzliche relevante Dokumente gefunden, welche nicht mit der grundlegenden Suchfunktionalität gefunden worden sind.

Konf.	SemSim	TxtSim	P(10)	P(20)	P(30)	infAP
Konf_1	Nein	Nein	0,2481	0,2089	0,1726	0,1523
Konf_2	Nein	Ja	0,3127	0,2823	0,2540	0,2706
Konf_3	Shortest Path ($> 0,4$)	Nein	0,3177	0,2620	0,2122	0,2029
Konf_4	Resnik ($> 0,5$)	Nein	0,3228	0,2646	0,2156	0,2134
Konf_5	Resnik ($> 0,7$)	Nein	0,3177	0,2620	0,2122	0,2029
Konf_6	Trento Vector ($> 0,1$)	Nein	0,2772	0,2259	0,1907	0,1809
Konf_7	Trento Vector ($> 0,4$)	Nein	0,2608	0,2228	0,1865	0,1687
Konf_8	Shortest Path ($> 0,4$)	Ja	0,3962	0,3544	0,3135	0,3552
Konf_9	Resnik ($> 0,5$)	Ja	0,3886	0,3456	0,3089	0,3502
Konf_10	Resnik ($> 0,7$)	Ja	0,3962	0,3544	0,3135	0,3551
Konf_11	Trento Vector ($> 0,1$)	Ja	0,3785	0,3278	0,2928	0,3432
Konf_12	Trento Vector ($> 0,4$)	Ja	0,3367	0,2962	0,2679	0,3017

Tabelle 8.1: Evaluierungswerte der Systemkonfigurationen, berechnet mit P(10), P(20), P(30) und infAP

In Tabelle 8.2 ist eine Reihung der 12 evaluierten Systemkonfigurationen nach ihrer Retrieval-Leistung für die unterschiedlichen Maße P(10), P(20), P(30) und infAP angeführt. Die Systemkonfiguration, die von der Mehrzahl der Evaluierungsmaße für einen Rang ermittelt werden, sind fett hervorgehoben.

Rang	P(10)	P(20)	P(30)	infAP
1 (beste)	**Konf_8** **Konf_10**	**Konf_8** **Konf_10**	**Konf_8** **Konf_10**	**Konf_8**
2				**Konf_10**
3	**Konf_9**	**Konf_9**	**Konf_9**	**Konf_9**
4	**Konf_11**	**Konf_11**	**Konf_11**	**Konf_11**
5	**Konf_12**	**Konf_12**	**Konf_12**	**Konf_12**
6	Konf_4	**Konf_2**	**Konf_2**	**Konf_2**
7	Konf_5 Konf_3	**Konf_4**	**Konf_4**	**Konf_4**
8		**Konf_5** **Konf_3**	**Konf_5** **Konf_3**	**Konf_5** **Konf_3**
9	Konf_2			
10	**Konf_6**	**Konf_6**	**Konf_6**	**Konf_6**
11	**Konf_7**	**Konf_7**	**Konf_7**	**Konf_7**
12 (schlechteste)	**Konf_1**	**Konf_1**	**Konf_1**	**Konf_1**

Tabelle 8.2: Reihung der Systemkonfigurationen basierend auf den Maßen P(10), P(20), P(30) und infAP. Jene Systemkonfiguration, welche von der Mehrheit der Maße geliefert wird ist fett hervorgehoben

Abbildung 8.2 zeigt die Retrieval-Leistung aller evaluierten Systemkonfigurationen anhand der Maße infAP, P(10), P(20) und P(30) im Vergleich zueinander. Ein höherer Balken steht für eine höhere Retrieval-Leistung. Jene Konfiguration, welche die höchste Retrieval-Leistung erzielt ist ganz links zu finden. Die Reihung der Systemkonfigurationen in Abbildung 8.2 (von links nach rechts) erfolgt nach dem Maß infAP.

8.3.6 Diskussion der Evaluierung

Im Folgenden werden die verwendeten Evaluierungsmaße diskutiert und es wird darauf eingegangen warum die Anzahl der Anfragen an das System und die Menge an Relevanzbewertungen für eine Evaluierung des Systems ausreichend sind.

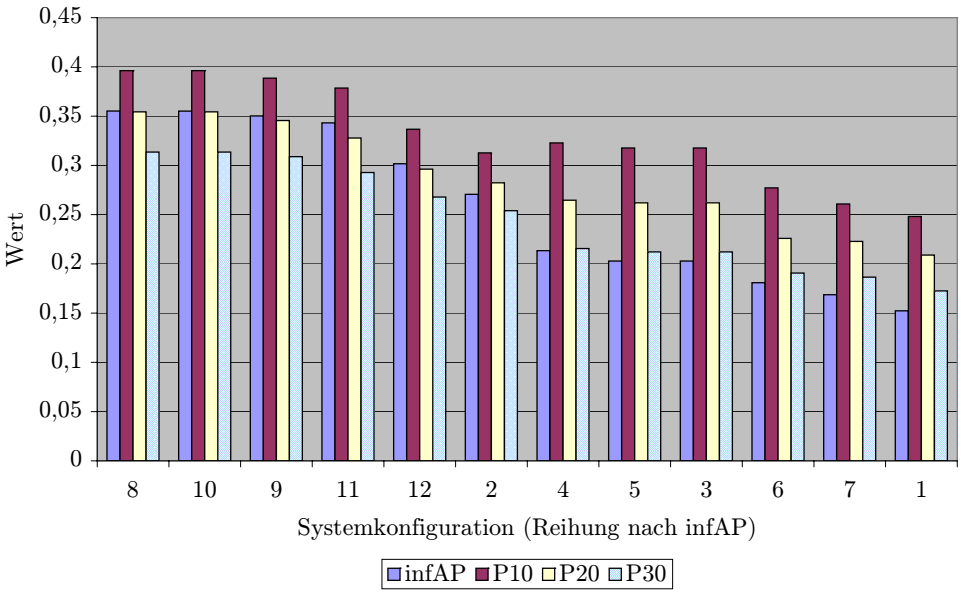

Abbildung 8.2: Retrieval-Leistung der evaluierten Systemkonfiguration

P(10), P(20) und P(30)

Buckley u. Voorhees (2000) evaluieren Evaluierungsmaße hinsichtlich ihrer Stabilität. Sie berechnen die Fehlerrate (error rate) von verschiedenen Maßen basierend auf der Anzahl von Fehlern, welche beim Vergleich von Systemen mittels eines bestimmten Maßes passieren. Als einen Fehler eines Maßes wird es angesehen, wenn System A besser ist als System B, das Maß allerdings für System B einen höheren (besseren) Evaluierungswert berechnet. Zur Berechnung der Fehlerrate wird die Anzahl der Fehler eines Maßes durch die Gesamtanzahl der möglichen Vergleiche zwischen zwei verschiedenen Systemen dividiert. Basieren auf zuvor durchgeführten Studien gehen Buckley u. Voorhees (2000) davon aus, dass eine Fehlerrate von 2.9% minimal akzeptabel ist. Sie zeigen, dass P(30) in ihrem Experiment exakt diese Fehlerrate von 2.9% aufweist, wenn 50 Anfragen verwendet werden. Sie schlagen vor, dass für Maße P(n), wo $n < 30$ gilt, die Anzahl der Anfragen an das System erhöht werden soll. Sie gehen davon aus, dass 100 Anfragen ein sicherer Wert für das Maß P(20) darstellen.

In der vorliegenden Arbeit wird ein Experiment mit 79 unterschiedlichen Anfragen durchgeführt. Die Maße P(10), P(20) und P(30) fanden Verwen-

dung. Den Ergebnissen von Buckley u. Voorhees (2000) folgend, sollte die Anzahl der Anfragen an das hier evaluierte System ausreichend für das Maß P(30) sein. Diese Vermutung wird zusätzlich dadurch bestätigt, dass die Reihung der 12 Systemkonfigurationen für die Maße P(20), P(30) und infAP identisch ist.

infAP

Größe des Pools: Die Trec 8 Ad-Hoc Collection besteht aus 528.155 Dokumenten und 50 Anfragen. Daraus ergibt sich ein Maximum von 26.407.750 rechnerisch möglichen Relevanzbewertungen. Die Menge an Anfrage-Dokument-Paaren wird durch Depth-100-Pooling (vgl. Abschnitt 8.2.1) von 129 Läufen erzeugt. 86.830 Anfrage-Dokument-Paare sind in der Ad-Hoc-Test-Collection auf Relevanz beurteilt. Somit wurden 0,33% der maximal möglichen Relevanzbewertung durchgeführt.

Die hier verwendete Testmenge besteht aus 1016 Dokumenten und 79 Anfragen. Hieraus ergibt sich ein rechnerisches Maximum von 81.054 möglichen Relevanzbewertungen. Für das hier vorgestellte Experiment wurden 2787 Anfrage-Dokument-Paare auf deren Relevanz beurteilt. Diese Menge ergibt sich aus Depth-30-Pooling von 12 Systemkonfigurationen und 855 zusätzlichen Relevanzbewertungen, welche für Läufe abgegeben wurden, welche nicht Teil des hier vorgestellten Experiments waren. Somit wurden 3,44% der rechnerisch maximal möglichen Relevanzbewertungen durchgeführt.

Anzahl der Relevanzbewertungen: Der Depth-100-Pool der 12 evaluierten Konfigurationen würde aus 5393 Anfrage-Dokument-Paaren bestehen[8]. Mit 2787 Anfrage-Dokument-Paaren wurden im hier vorgestellten Experiment 51,68% dieses potentiellen Depth-100-Pools bewertet. Yilmaz u. Aslam (2006) berichten von einer Kendall's τ basierten Rangkorrelation von über 0,9 zwischen infAP und AP bei 25% der maximal möglichen Relevanzbewertungen des Depth-100-Pools der Trec 8 Ad-Hoc Collection (vgl. Abschnitt 8.2.1). Yilmaz u. Aslam (2006) gehen davon aus, dass zwei Reihungen mit einer Rangkorrelation von über 0,9 als gleichwertig angesehen

[8] dieser Wert wurde experimentell ermittelt

werden können.

Da 51,68% des potentiellen Depth-100-Pools beurteilt wurden, wird hier davon ausgegangen, dass infAP eine ausreichend genaue Annährung an AP für die Reihung der Systemkonfigurationen liefert. Wiederum wird diese Vermutung dadurch gestützt, dass die Reihung der 12 Systemkonfigurationen für P(20), P(30) und infAP ident ist.

8.3.7 Abschließende Betrachtung zur Evaluierung

Gegenwärtig ist Information Retrieval im Semantic Web ein inhomogenes Feld (Scheir u. a., 2007c). Obwohl eine immer größer werdende Menge an Ansätzen zur Suche im Semantic Web existiert, werden gegenwärtig unterschiedliche Arten an Anfragen von diesen Systemen unterstützt und unterschiedliche Ausgabewerte generiert. Diese Situation verkompliziert das Erstellen von allgemein gültigen Regeln für die Evaluierung von Information-Retrieval-Systemen für das Semantic Web und die Erstellung einer allgemein gültigen Test-Collection für dieses Anwendungsfeld des Information Retrieval.

Die Evaluierung des hier vorgestellten Systems stellt einen Schritt in die Richtung der standardisierten Evaluierung von Ansätzen zur Suche im Semantic-Web-Umfeld dar. Es wurde eine Test-Collection erstellt, und das hier vorgestellte System wurde anhand von Standardmaßen aus dem Information Retrieval evaluiert. Die Gültigkeit der ermittelten Ergebnisse wurde basierend auf existierender Forschung aus dem Feld des Information Retrieval argumentiert.

8.4 Zusammenfassung

In diesem Abschnitt wurde die Evaluierung des in dieser Arbeit entwickelten Systems beschrieben. Standardmaße aus dem Information Retrieval wurden für die Evaluierung eingesetzt. Unter Berücksichtigung von aktueller Forschung zur Evaluierung von Information-Retrieval-Systemen wurden die durch die Evaluierung ermittelten Ergebnisse betrachtet. Auf Grund des Fehlens von standardisierten Test-Collections für das Semantic Web wurde eine eigene Test-Collection entwickelt. Diese basiert auf den verfügbaren Daten der ersten Version des APOSDLE-Systems.

9 Zusammenfassung und Ausblick

9.1 Einleitung

In dieser Arbeit wurde ein Modell zur Suche im Semantic Web entwickelt, welches in Form eines Systems zur Suche am Semantic Desktop realisiert und evaluiert wurde. Zum Abschluss dieser Arbeit erfolgt eine rückblickende Betrachtung der Forschungshypothese und der Untersuchungsfragen aus Abschnitt 1.3, eine Diskussion der Validität der in dieser Arbeit erzielten Erkenntnisse und ein Ausblick auf mögliche zukünftige Forschung, ausgehend von der vorliegenden Arbeit.

9.2 Rückblickende Betrachtung der Forschungshypothese und der Untersuchungsfragen

In Abschnitt 1.3 wurde die Forschungshypothese der vorliegenden Arbeit aufgestellt und jene Untersuchungsfragen angeführt, die dieser Arbeit zu Grunde liegen. Im folgenden Abschnitt werden Forschungshypothese und Untersuchungsfragen rückblickend betrachtet. Sie werden erneut angeführt (gekennzeichnet durch •) und kommentiert (gekennzeichnet durch ▷). Des Weiteren wird auf jene Teile dieser Arbeit verwiesen, in welchen die Fragestellungen bearbeitet werden.

Die Forschungshypothese, welche der vorliegenden Arbeit zu Grunde liegt lautet:

- Assoziatives Retrieval erhöht die Effektivität eines Information-Retrieval-Modells für das Semantic Web

▷ Aufgrund des durchgeführten Experiments kann diese Forschungshypothese nicht verworfen werden, da es hierfür klare Indizien gibt. Eine detailliere Betrachtung folgt in Abschnitt 9.3.1.

Im Vorfeld der in der Arbeit durchgeführten Modellentwicklung wurden aktuelle Ansätze zur Suche im Semantic Web beleuchtet und assoziative Retrieval-Methoden für ihre Anwendung im Kontext der Arbeit untersucht. Folgende Untersuchungsfragen wurden bearbeitet:

- Welche Suchansätze für Semantic Web bzw. Semantic Desktop gibt es bereits?

▷ Im Rahmen dieser Arbeit wurde eine explorative Studie zu aktuellen Suchansätzen für das Semantic Web und den Semantic Desktop durchgeführt (vgl. Abschnitt 3). Diese diente auf der einen Seite dazu, neue Richtungen für eigene Forschung zu identifizieren und auf der anderen Seite dazu, die Problemdomäne zu strukturieren und die Konstruktvalidität des durchgeführten Experiments (vgl. Abschnitt 9.3.1) zu erhöhen.

- Welche assoziativen Suchansätze gibt es bereits?

▷ Ebenfalls wurden in dieser Arbeit Systeme auf Basis von Assoziativem Retrieval, Assoziativen Netzen und Spreading Activation untersucht. Die betrachteten Systeme wurden einander gegenübergestellt (vgl. Abschnitt 4). Dies diente dazu vorhandene Methoden zu strukturieren und die Konstruktvalidität des durchgeführten Experiments (vgl. Abschnitt 9.3.1) zu erhöhen.

Im Rahmen der in dieser Arbeit durchgeführten Modellentwicklung, Implementierung und Evaluierung wurden folgende Untersuchungsfragen bearbeitet:

- Kann Assoziatives Retrieval für das Semantic Web bzw. den Semantic Desktop eingesetzt werden? Bzw. können assoziative Suchansätze in aktuelle Ansätze zur Suche im Semantic Web bzw. am Semantic Desktop integriert werden?

▷ In der vorliegenden Arbeit wurde ein Modell für die Suche im Semantic Web entwickelt (vgl. Abschnitt 6), welches in Form eines Systems

für den Semantic Desktop umgesetzt (vgl. Abschnitt 7) und evaluiert (vgl. Abschnitt 8) wurde. In das entwickelte Modell und System wurde ein Assoziativer Suchansatz integriert. Beide Fragestellungen können somit mit Ja beantwortet werden.

- Kann Assoziatives Retrieval die Effektivität von Information-Retrieval-Systemen für Semantic Web oder Semantic Desktop steigern?

▷ Unterschiedliche Konfigurationen des entwickelten Systems wurde gegeneinander evaluierte (vgl. Abschnitt 8). Alle Systemkonfigurationen, welche auf Assoziativem Retrieval basierten erzielten eine höhere Retrieval-Leistung als der getestete nicht-assoziative Suchansatz. Für das durchgeführte Experiment (vgl. Abschnitt 8) kann die Frage somit mit Ja beantwortet werden. Für eine Diskussion der Generalisierbarkeit sei auf Abschnitt 9.3.1 verwiesen.

Aus der Anwendung eines Assoziative Netzes für das entwickelte Retrieval-Modell ergaben sich weitere Untersuchungsfragen:

- Wie verhalten sich Netz-basierte Retrieval-Modelle im Vergleich zu klassischen Modellen, wie dem Vektorraummodell? Können Netz-basierte Ansätze äquivalente Ergebnisse liefern?

▷ In dieser Arbeit wurden Netz-basierte Retrieval-Modelle mit dem Vektorraummodell verglichen und Gleichheiten und Unterschiede angeführt (vgl. Abschnitt 5). Es wurde aufgezeigt, dass Netzmodelle äquivalente Ergebnisse zum Vektorraummodell liefern können und wie Netzmodelle für diesen Fall konfiguriert sein müssen.

- Können Netz-basierte Suchansätze für Information Retrieval im Semantic Web eingesetzt werden?

▷ In der vorliegenden Arbeit wurde ein Netz-basierter Suchansatz (vgl. Abschnitt 6) für das Semantic Web entwickelt, welcher im Semantic-Desktop-Umfeld umgesetzt wurde (vgl. Abschnitt 7). Diese Frage kann somit mit Ja beantwortet werden.

9.3 Kritische Betrachtung der Forschungsmethodik

Cook u. Campbell (1979) verstehen unter *Validität* die bestmögliche Annährung an die Wahrheit oder Falschheit von Aussagen aus einem positivistischen Betrachtungspunkt. Das Konzept der Validität findet in letzter Zeit auch in der Informatik seine Verwendung, so betrachten Easterbrook u. a. (2008) empirische Forschungsmethoden für deren Anwendung in der Softwareenwicklungsforschung. Weiters schlagen sie vor einen Abschnitt mit *threats to validity* in positivistischen empirischen Studien anzuführend, um Schwachstellen des Forschungsentwurfs zu diskutieren. Sie begründen diese Vorgehensweise damit, dass jedes Forschungs-Design Schwächen hat und durch den expliziten Hinweis auf diese Forscher zeigen können, dass sie sich dieser Schwächen bewusst sind und was sie getan haben um deren Effekte zu minimieren.

Im Folgenden werden die Erkenntnisse dieser Arbeit unter Berücksichtigung der in (Cook u. Campbell, 1979) und (Easterbrook u. a., 2008) genannten Validitätskriterien diskutiert. Hierbei liegt der Fokus auf den Ergebnissen und der Durchführung der Evaluierung des entwickelten Systems (vgl. Abschnitt 8).

9.3.1 Bedrohungen der Validität[1]

Interne Validität: Die interne Validität steht für jene Validität mit welcher, basierend auf dem durchgeführten Experiment, Aussagen über den kausalen Zusammenhang zwischen zwei Variablen getroffen werden können (Cook u. Campbell, 1979). Im hier durchgeführten Experiment handelt es sich bei den Variablen um die gewählte Systemkonfiguration und die Retrieval-Leistung des Systems. Im durchgeführten Experiment wurden Systemkonfigurationen miteinander, basierend auf der gelieferten Retrieval-Leistung, verglichen. Die Frage, die es hinsichtlich der internen Validität zu beantworten gilt, ist, ob die Änderung der Retrieval-Leistung ausschließlich durch die Änderung der getesteten Systemkonfiguration erzielt wurde

[1] Angelehnt an das englische *threats to validity* aus (Cook u. Campbell, 1979)

oder ob andere Faktoren auch Einfluss auf diese Änderung haben.

Die Systemkonfigurationen basieren auf den gewählten Ähnlichkeitsmaßen und deren Geeignetheit für die verwendete Ontologie und die verwendete Dokumentensammlung. Mit dem durchgeführten Experiment können also Aussagen über den Zusammenhang zwischen Ähnlichkeitsmaßen und Retrieval-Leistung für die verwendete Ontologie, die verwendeten Annotationen und die verwendete Dokumentensammlung getroffen werden. Für diese Situation ist die interne Validität des Zusammenhangs der Systemkonfiguration mit der Retrieval-Leistung hoch. Keine anderen Einflüsse beeinflussen die Retrieval-Leistung.

Externe Validität: Die externe Validität lässt Aussagen darüber zu, in wie weit sich das Ergebnis des Experiments auf andere Situation (Personen, Orte, Zeitpunkte) übertragen lässt (Cook u. Campbell, 1979). Im konkreten Anwendungsfall stellt die Übertragbarkeit der Ergebnisse auf anderen Retrieval-Szenarien die größte Schwierigkeit dar. Das hier beschriebene Experiment wurde in einer Situation mit einer speziellen Dokumentenmenge und speziellen Annotationen durchgeführt. Es lässt sich nicht generalisieren zu welchen Ergebnissen das durchgeführte Experiment unter der Verwendung von anderen Ontologien, Annotationen oder Dokumentensammlungen kommt. Aufgrund des durchgeführten Experiments kann allerdings die Forschungshypothese ,,Assoziatives Retrieval erhöht die Effektivität eines Information-Retrieval-Modells für das Semantic Web" nicht verworfen werden, da es hierfür klare Indizien gibt. Durch die Verwendung unterschiedlicher Retrieval-Maße zur Bewertung der Retrieval-Leistung wurde die externe Validität zusätzlich erhöht. Um die externe Validität weiter zu erhöhen, können zusätzliche Experimente durchgeführt werden, in denen der Retrieval-Ansatz in anderen Situationen erprobt wird.

Ein weiterer potentieller Einflussfaktor auf die externe Validität ist, dass die Ergebnisse zu allen getesteten Systemkonfigurationen von einer Person bewertet wurden. Es kann davon ausgegangen werden, dass unterschiedliche Personen die Relevanz eines Ergebnisses unterschiedlich empfinden, da sie unterschiedliches Vorwissen und Anschauungen mitbringen. Die externe Validität kann durch das Miteinbeziehen anderer Bewertender in derselben Anwendungsdomäne erhöht werden.

Statistische Validität: Die statistische Validität lässt Aussagen darüber zu, ob, basierend auf der angewendeten Methodik, ein Zusammenhang (Kovarianz) zwischen Variablen besteht (Cook u. Campbell, 1979).

Da das Verhalten des getesteten Systems deterministisch ist, bleibt als Quelle für zufällige Fehler die Relevanzbewertung von Suchergebnissen durch Menschen. Für das durchgeführte Experiment wurden erprobte Standardmethoden aus dem Information Retrieval verwendet. Basierend auf existierender Literatur zur Evaluierung von Information-Retrieval-Systemen wurde argumentiert, warum die Anzahl der Anfragen an die unterschiedlichen Systemkonfigurationen und die Anzahl der Relevanzbewertungen ausreichend für eine Reihung der Systemkonfigurationen ist.

Konstruktvalidität: Konstruktvalidität lässt Aussagen darüber zu in wie weit die im Experiment verwendeten Variablen für jene Konzepte stehen, die gemessen werden sollen (Cook u. Campbell, 1979) und ob dieser Zusammenhang von anderen Wissenschaftlern gleich aufgefasst wird.

Durch Literaturstudien im Vorfeld wurde die Konstruktvalidität unterstützt. Grundlegende Begriffe wie Semantic Web, (Assoziatives) Information Retrieval und vor allem die für die Evaluierung genutzten Begriffe wie Retrieval-Leistung wurden basierend auf vorhandener Literatur aufgearbeitet. Erst im Anschluss daran wurden Modell, System und Evaluierung entworfen.

Reliabilität: Reliabilität lässt Aussagen darüber zu in wie weit eine Studie dieselben Ergebnisse erzielt, wenn sie von anderen Forschern wiederholt wird (Easterbrook u. a., 2008). Für das konkrete Experiment sind die folgenden Aspekte zu unterscheiden: (1) die Durchführung des Experiments als solches und (2) die Erhebung der Ergebnisse.

(1) Der Versuchsaufbau wurde detailliert beschrieben, um ihn nachvollziehbar zu machen[2]. Mit `trec_eval` und den Maßen infAP und P(n) wurden Werkzeuge verwendet, die in der Information-Retrieval-Community ge-

[2] Easterbrook u. a. (2008) sehen die nachvollziehbare Dokumentation des Versuchsaufbaus als Teil der Reliabilität. (Bortz u. Döring, 2006) sehen dies als Objektivität an, welche eine Voraussetzung für Reliabilität ist.

bräuchlich sind. Unter diesem Aspekt ist die Reliabilität hoch.

(2) Eine Schwierigkeit für die Reliabilität, die bei jedem Information-Retrieval-Experiment auftritt, ist die Durchführung der Relevanzbewertung. Hierbei spielt das unterschiedliche Empfinden der Relevanz durch die *gleichen* Bewertenden zu unterschiedlichen Zeitpunkten ein Rolle. Ein weiterer potentieller Einwand in diesem Zusammenhang ist, dass die Ergebnisse zu den getesteten Systemkonfigurationen vom Ersteller des Systems bewertet wurden. Dieser könnte bewusst Systemkonfigurationen bevorzugen und somit die Evaluierung beeinflussen. Dies wird aufgrund des gewählten Pooling-Ansatzes vermieden. Ergebnisse, welche durch mehrere Systemkonfigurationen erzielt werden, werden nur einmal bewertet. Zusätzlich erfolgte die Bewertung von Ergebnissen ohne die Anzeige von Information über jene Systemkonfigurationen, welche diese Ergebnisse lieferten (vgl. Abbildung 8.1 in Abschnitt 8.3.4).

Das getestete System ist deterministisch, hat also keinen negativen Einfluss auf die Reliabilität des Experiments. Werden von anderen Forschergruppen dieselbe Ontologie, die selben Annotationen, dieselbe Dokumentenbasis und die selbe Menge an Relevanzbewertungen verwendet, so sind sie in der Lage, auf Basis der Beschreibung des Systems, auf die gleichen Ergebnisse zu kommen. Unter diesen Voraussetzungen ist die Reliabilität des Experiments hoch.

9.3.2 Zusammenfassende Betrachtung

Zusammenfassend betrachtet weist das durchgeführte Experiment eine hohe interne Validität auf, ebenfalls statistische Validität und Konstruktvalidität sind sichergestellt. Aufgrund des durchgeführten Experiments kann die Forschungshypothese ,,Assoziatives Retrieval erhöht die Effektivität eines Information-Retrieval-Modells für das Semantic Web" nicht verworfen werden, da es hierfür klare Indizien gibt. Durch zusätzliche Experimente in anderen Umgebungen (mit anderen Versuchspersonen, welche die Relevanz der Ergebnisse bewerten, einer anderer Ontologie, anderen Annotationen und einer anderen Dokumentenmenge) kann die (externe) Validität, der in der Forschungshypothese getroffenen Aussage, erhöht werden.

Wie Bortz u. Döring (2006) feststellen *haben Laboruntersuchung eine hohe interne Validität* und *verfügen in der Regel über eine geringere externe*

Validität. Somit handelt es sich bei der hier beschriebenen Verteilung der internen und externen Validität um eine typische Situation für Laborexperimente, wie es Untersuchungen von Information-Retrieval-Systemen in der Regel sind (vgl. Abschnitt 8.2 (Baeza-Yates u. Ribeiro-Neto, 1999)).

Hinsichtlich der Reliabilität der Ergebnisse ist festzuhalten, dass durch die Anwendung von erprobten Evaluierungsmethoden und die Dokumentation des Prozesses die Reproduzierbarkeit gewährleistet ist. Eine deutliche Verbesserung der Reliabilität könnte durch die Verwendung eines standardisierten Test-Korpus erzielt werden.

9.4 Mögliche zukünftige Forschung

Dieser Abschnitt gibt einen Ausblick auf mögliche zukünftige Forschung ausgehend von dem in dieser Arbeit entwickelten Retrieval-Ansatz.

9.4.1 Weitere Experimente in anderen Situationen

Wie die kritische Betrachtung der Forschungsmethodik in Abschnitt 9.3 ergeben hat, stellt gegenwärtig die Übertragbarkeit, der durch die Evaluierung ermittelten Ergebnisse, auf anderen Szenarien die größte Schwierigkeit dar. Es lässt sich nicht generalisieren, zu welchen Ergebnissen das durchgeführte Experiment unter der Verwendung von anderen Ontologien, Annotationen oder Dokumentensammlungen kommt. Um die Validität der Erkenntnisse über den entwickelten Ansatz weiter zu erhöhen, muss zukünftige Forschung darauf abzielen zusätzliche Experimente durchzuführen, in welchen das entwickelte Retrieval-Modell in anderen Situationen erprobt wird.

9.4.2 Implementierung des Modells in einem System für die Suche im Semantic Web

In der vorliegenden Arbeit wurde ein Retrieval-Modell für das Semantic Web entwickelt, welches in einem System für die Suche am Semantic Desktop implementiert wurde. Gegenstand weiterer Forschung könnte es sein das entwickeltet Modell in einem System zur Suche im Semantic Web zu implementieren. Für einen Einsatz im Semantic Web ist keine Modifikation des Modells an sich notwendig, da Semantic Web und Semantic Desktop auf den gleichen

Technologien aufbauen. Für eine Implementierung müssen allerdings andere Wege der Datenbeschaffung beschritten werden. Sind am Semantic Desktop Ressourcen auf (wenige) Rechnern im Unternehmen verteilt, so sind diese im Semantic Web auf (zahlreiche) unterschiedliche Web-Server verteilt. Werkzeuge aus dem Web-Retrieval wie Web-Crawler können Verwendung finden um die zu durchsuchenden Daten von unterschiedlichen Web-Servern zusammenzutragen.

9.4.3 Erstellung einer Test-Collection für Information Retrieval im Semantic Web

Ein zentraler Bestandteil dieser Arbeit war die Erstellung, der für die Evaluierung des entwickelten Systems verwendeten Test-Collection. Ähnliche Wege der Evaluierung über die Verwendung von Test-Collections wurden auch in anderen Untersuchungen zum Thema Information Retrieval im Semantic Web eingeschlagen (vgl. (Castells u. a., 2007), (Finin u. a., 2005)). Neben der Ersparnis des hohen Aufwandes der individuellen Erstellung einer Test-Umgebung, würde eine einheitliche Test-Collection auch den Vergleich der Retrieval-Leistung verschiedener Systeme ermöglichen.

Im Vorfeld der Erstellung der Test-Umgebung sollte eine Phase der Anforderungserhebung durchlaufen werden, da, wie in Abschnitt 3.4 beschrieben, unterschiedliche Zugänge zu Information Retrieval im Semantic Web existieren. Ebenfalls existieren, wie in Abschnitt 6.3.1 angeführt, verschiedene Ansätze zur semantischen Annotation, welche in den Entwurf einer Test-Collection für Information Retrieval im Semantic Web einfließen können.

9.4.4 Zuordnung von semantischen Ähnlichkeitsmaßen zu Charakteristika von Ontologien

Ontologien können auf unterschiedliche Art und Weise modelliert werden. Sie können tiefe oder flache Klassenhierarchien aufweisen und die Anzahl der Eigenschaften in der Ontologie kann variieren. Da unterschiedliche semantische Ähnlichkeitsmaße unterschiedliche Charakteristika einer Ontologie für die Ähnlichkeitsbestimmung heranziehen, muss davon ausgegangenen werden, dass sich unterschiedliche Maße für verschiedene Ontologien unterschiedlich gut eignen. Aufgabe weiterer Forschung kann es sein zu identi-

fizieren, auf Basis welcher Charakteristika ein Ähnlichkeitsmaß für die Bestimmung der semantischen Ähnlichkeit in einer bestimmten Ontologie auszuwählen ist.

9.4.5 Finden von zusätzlicher Information zum Ziehen von Schlüssen

Fensel u. Harmelen (2007) postulieren einen neuen Ansatz für das Ziehen von Schlüssen im Semantic Web, welcher die Grundlage für das in Abschnitt 1.2.1 beschrieben LarKC-Projekt darstellt. Sie stellen fest, dass die Menge an semantischer Information (RDF-Tripel) im Web die Fähigkeiten jedes aktuell verfügbaren Reasoners überschreitet und gegenwärtige Ansätze zum Ziehen von Schlüssen nicht skalieren. Aus diesem Grund schlagen sie vor erst eine Stichprobe zu ziehen auf der Reasoning betrieben wird. Wenn genügend Zeit vorhanden ist oder das Resultat nicht vertrauenswürdig erscheint wird eine größere Stichprobe gezogen.

Eine zentrale Frage in diesem Ansatz ist, wie die ursprüngliche Stichprobe erweitert wird. Assoziatives Retrieval für das Semantic Web kann an diesem Punkt ansetzen: Basierend auf semantischer und inhaltsbasierter Ähnlichkeit von Knoten im Semantic-Web-Graph können neue Tripel bzw. Dokumente (welche Tripel enthalten) gesucht werden und somit die lokale Wissensbasis für das Ziehen von Schlüssen erweitert werden. Hierdurch wird verhindert auf der gesamten im Web verfügbaren Wissensbasis arbeiten zu müssen.

9.5 Zusammenfassung

Zum Abschluss dieser Arbeit erfolgte eine rückblickende Betrachtung der Forschungshypothese und der Untersuchungsfragen aus Abschnitt 1.3. Ebenfalls wurde die Validität der in dieser Arbeit erzielten Erkenntnisse diskutiert und, ausgehend von der vorliegenden Arbeit, ein Ausblick auf mögliche zukünftige Forschung gegeben.

Literaturverzeichnis

[Agosti u. Crestani 1993] Agosti, Maristella ; Crestani, Fabio: A methodology for the automatic construction of a hypertext for information retrieval. In: *SAC '93: Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing*. New York, NY, USA : ACM Press, 1993, S. 745–753

[Agosti u. a. 1996] Agosti, Maristella ; Crestani, Fabio ; Melucci, Massimo: Design and implementation of a tool for the automatic construction of hypertexts for information retrieval. In: *Information Processing and Management* 32 (1996), Nr. 4, S. 459–476

[Agosti u. Marchetti 1992] Agosti, Maristella ; Marchetti, Pier G.: User navigation in the IRS conceptual structure through a semantic association function. In: *The Computer Journal* 35 (1992), Nr. 3, S. 194–199

[Alani u. a. 2003] Alani, Harith ; Dasmahapatra, Srinandan ; O'Hara, Kieron ; Shadbolt, Nigel: Identifying Communities of Practice through Ontology Network Analysis. In: *IEEE Intelligent Systems* 18 (2003), Nr. 2, S. 18–25

[Alani u. a. 2002] Alani, Harith ; O'Hara, Kieron ; Shadbolt, Nigel: ONTOCOPI: Methods and Tools for Identifying Communities of Practice. In: *Proceedings of the IFIP 17th World Computer Congress - TC12 Stream on Intelligent Information Processing*, Kluwer, B.V., 2002, S. 225–236

[Alves u. Jorge 2005] Alves, Mario A. ; Jorge, Alipio M.: Minibrain: a generic model of spreading activation in computers, and example specialisations. In: *ECML/PKDD 2005 workshop 'Subsymbolic paradigms for learning in structured domains'*, 2005

[Anderson 1998] Anderson, James A.: Associative networks. In: *The handbook of*

brain theory and neural networks. Cambridge, MA, USA : MIT Press, 1998, S. 102–107

[Anderson 1976] Anderson, John R.: *Language, Memory and Thought*. Hillsdale, N. J. : Erlbaum, 1976

[Anderson 1983] Anderson, John R.: A spreading activation theory of memory. In: *Journal of Verbal Learning and Verbal Behaviour* 22 (1983), S. 261–295

[Anderson u. Bower 1973] Anderson, John R. ; Bower, Gordon H.: *Human associative memory*. Washington : Winston and Sons, 1973

[Anderson u. Pirolli 1984] Anderson, John R. ; Pirolli, Peter L.: Spread of activation. In: *Journal of Experimental Psychology: Learning, Memory, and Cognition* 10 (1984), S. 791–798

[Antoniou u. van Harmelen 2004] Antoniou, Grigoris ; Harmelen, Frank van: *A Semantic Web Primer*. Cambridge, MA, USA : MIT Press, 2004

[APOSDLE consortium 2007] APOSDLE consortium: *First Prototype APOSDLE - Deliverables D1.2, D2.2, D3.2, D4.2, D5.2*. 2007

[Baader u. a. 2003] Baader, Franz (Hrsg.) ; Calvanese, Diego (Hrsg.) ; McGuinness, Deborah L. (Hrsg.) ; Nardi, Daniele (Hrsg.) ; Patel-Schneider, Peter F. (Hrsg.): *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003

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

[Balmin u. a. 2004] Balmin, Andrey ; Hristidis, Vagelis ; Papakonstantinou, Yannis: ObjectRank: Authority-Based Keyword Search in Databases. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, 2004, S. 564–575

[Bamba u. Mukherjea 2004] Bamba, Bhuvan ; Mukherjea, Sougata: Utilizing Resource Importance for Ranking Semantic Web Query Results. In: *Semantic Web and Databases, Second International Workshop, SWDB 2004, Toronto, Canada, August 29-30, 2004, Revised Selected Papers*, 2004, S. 185–198

[Bangyong u. a. 2005] Bangyong, Liang ; Jie, Tang ; Juanzi, Li: Association search in semantic web: search + inference. In: *WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*. New York, NY, USA : ACM Press, 2005, S. 992–993

[Bar-Yossef u. a. 1999] Bar-Yossef, Ziv ; Kanza, Yaron ; Kogan, Yakov A. ; Nutt, Werner ; Sagiv, Yehoshua: Querying Semantically Tagged Documents on the World-Wide Web. In: *NGIT '99: Proceedings of the 4th International Workshop on Next Generation Information Technologies and Systems*. London, UK : Springer-Verlag, 1999, S. 2–19

[Barnden u. a. 2002] *Kapitel* Part III: Articles. In: Barnden, John A. ; Lee, Mark G. ; Viezzer, Manuel: *Semantic networks*. Cambridge, MA, USA : MIT Press, 2002, S. 1010–1013

[Bechhofer u. Goble 2001] Bechhofer, Sean ; Goble, Carole: Towards Annotation using DAML+OIL. In: *K-CAP 2001 workshop on Knowledge Markup and Semantic Annotation*. Victoria B.C, October 2001

[Beckett 2001] Beckett, Dave: *N-Triples, W3C RDF Core WG Internal Working Draft*. 2001. – `http://www.w3.org/2001/sw/RDFCore/ntriples/` (21. 04. 2005)

[Beckett 2004] Beckett, Dave: *RDF/XML Syntax Specification (Revised), W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/rdf-syntax-grammar/`, Abruf: 17.02.2008

[Belew 1986] Belew, Richard K.: *Adaptive information retrieval: machine learning in associative networks (connectionist, free-text, browsing, feedback)*, University of Michigan, Diss., 1986

[Belew 1987] Belew, Richard K.: A connectionist approach to conceptual information retrieval. In: *ICAIL '87: Proceedings of the 1st international conference on Artificial intelligence and law*. New York, NY, USA : ACM Press, 1987, S. 116–126

[Belew 1989] Belew, Richard K.: Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. In: *SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 1989, S. 11–20

[Berger u. a. 2003] Berger, H. ; Dittenbach, M. ; Merkl, D.: Activation on the Move: Querying Tourism Information via Spreading Activation. In: Mařík, V. (Hrsg.) ; Retschitzegger, W. (Hrsg.) ; Štěpánková, O. (Hrsg.): *Proceedings of the 14th International Conference on Database and Expert Systems Applications (DEXA 2003)*. Prague, Czech Republic : Springer-Verlag, September 1-5 2003 (LNCS 2736), S. 474–483

[Berger 2003] Berger, Helmut: *Activation on the Move: Adaptive Information Retrieval via Spreading Activation*, Technischen Universität Wien, Diss., 2003

[Berger u. a. 2004a] Berger, Helmut ; Dittenbach, Michael ; Merkl, Dieter: An Accommodation Recommender System Based on Associative Networks. In: Frew, A. J. (Hrsg.): *Proceedings of the 11th International Conference on Information and Communication Technologies in Tourism (ENTER 2004)*. Cairo, Egypt : Springer-Verlag, January 26–28 2004, S. 216–227

[Berger u. a. 2004b] Berger, Helmut ; Dittenbach, Michael ; Merkl, Dieter: An Adaptive Information Retrieval System based on Associative Networks. In: Hartmann, S. (Hrsg.) ; Roddick, J. (Hrsg.): *Proceedings of the 1st Asia-Pacific Conference on Conceptual Modelling (APCCM 2004)* Bd. 31. Dunedin, New Zealand : Australian Computer Society Inc., January 18–22 2004 (Conferences in Research and Practice in Information Technology), S. 27–36

[Berners-Lee u. a. 2005] Berners-Lee, T. ; Fielding, R. ; Masinter, L.: *Network Working Group - Request for Comments: 3986 - Uniform Resource Identifier (URI): Generic Syntax*. Version: 2005. `http://www.ietf.org/rfc/rfc3986.txt`, Abruf: 17.02.2008

[Berners-Lee u. a. 1994] Berners-Lee, T. ; Masinter, L. ; McCahill, M.: *Network Working Group - Request for Comments: 1738 - Uniform Resource Locators (URL)*. Version: 1994. `http://www.ietf.org/rfc/rfc1738.txt`, Abruf: 17.02.2008

[Berners-Lee 2006] Berners-Lee, Tim: *Artificial Intelligence and the Semantic Web, keynote at The Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*. Version: 2006. `http://www.w3.org/2006/Talks/0718-aaai-tbl/`, Abruf: 17.02.2008

[Berners-Lee 2006] Berners-Lee, Tim: *Notation 3*. Version: 2006. `http://www.w3.org/DesignIssues/Notation3.html`, Abruf: 17.02.2008

[Berners-Lee 2006] Berners-Lee, Tim: *Trust*. Version: 2006. `http://www.w3.org/2000/10/swap/doc/Trust`, Abruf: 17.02.2008

[Berners-Lee u. a. 1999] Berners-Lee, Tim ; Fischetti, Mark ; Dertouzos, Michael L.: *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper San Francisco, 1999

[Berners-Lee u. a. 2001] Berners-Lee, Tim ; Hendler, James ; Lassila, Ora: The Semantic Web. In: *Scientific American* (2001), May. `http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21`, Abruf: 17.02.2008

[Bernstein u. a. 2005] Bernstein, Abraham ; Kaufmann, Esther ; Kiefer, Christoph ; Bürki, Christoph: SimPack: A Generic Java Library for Similarity Mea-

sures in Ontologies / University of Zurich, Department of Informatics. Winterthurerstrasse 190, 8057 Zurich, Switzerland, August 2005. – Forschungsbericht

[Bonino u. a. 2003] Bonino, Dario ; Corno, Fulvio ; Farinetti, Laura: DOSE: A Distributed Open Semantic Elaboration Platform. In: *ICTAI '03: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC, USA : IEEE Computer Society, 2003. – ISBN 0–7695–2038–3, S. 580

[Bonino u. a. 2004] Bonino, Dario ; Corno, Fulvio ; Farinetti, Laura ; Bosca, Alessio: Ontology Driven Semantic Search. In: *WSEAS Transaction on Information Science and Application* 1 (2004), Nr. 6, S. 1597–1605

[Bortz u. Döring 2006] Bortz, Jürgen ; Döring, Nicola: *Forschungsmethoden und Evaluation: für Human- und Sozialwissenschaftler*. 4., überarb. Heidelberg : Springer, 2006

[Brickley u. Guha 2004] Brickley, Dan ; Guha, R.V.: *RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/rdf-schema/`, Abruf: 17.02.2008

[Brin u. Page 1998] Brin, Sergey ; Page, Lawrence: The anatomy of a large-scale hypertextual Web search engine. In: *WWW7: Proceedings of the seventh international conference on World Wide Web 7*. Amsterdam, The Netherlands, The Netherlands : Elsevier Science Publishers B. V., 1998, S. 107–117

[Buckley 1985] Buckley, Chris: Implementation of the SMART Information Retrieval System / Department of Computer Science, Cornell University. Ithaca, NY, USA : Cornell University, 1985 (TR 85-686). – Forschungsbericht

[Buckley u. Voorhees 2000] Buckley, Chris ; Voorhees, Ellen M.: Evaluating evaluation measure stability. In: *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 2000, S. 33–40

[Buckley u. Voorhees 2004] Buckley, Chris ; Voorhees, Ellen M.: Retrieval evaluation with incomplete information. In: *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 2004, S. 25–32

[Carterette u. a. 2006] Carterette, Ben ; Allan, James ; Sitaraman, Ramesh: Minimal test collections for retrieval evaluation. In: *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 2006, S. 268–275

[Castells u. a. 2007] Castells, Pablo ; Fernández, Miriam ; Vallet, David: An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. In: *IEEE Transactions on Knowledge and Data Engineering* 19 (2007), Nr. 2, S. 261–272

[Chen u. Ng 1995] Chen, H. ; Ng, T.: An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound search vs. connectionist Hopfield net activation. In: *Journal of the American Society for Information Science* 46 (1995), Nr. 5, S. 348–369

[Chen u. Dhar 1991] Chen, Hsinchun ; Dhar, Vasant: Cognitive process as a basis for intelligent retrieval systems design. In: *Information Processing and Management* 27 (1991), Nr. 5, S. 405–432

[Chen u. a. 1993] Chen, Hsinchun ; Lynch, Kevin J. ; Basu, Koushik ; Ng, Tobun D.: Generating, integrating, and activating thesauri for concept-based document retrieval. In: *IEEE Expert: Intelligent Systems and Their Applications* 8 (1993), Nr. 2, S. 25–34

[Chirita u. a. 2006] Chirita, Paul-Alexandru ; Costache, Stefania ; Nejdl, Wolfgang ; Paiu, Raluca: Beagle++: Semantically Enhanced Searching and Ranking on the Desktop. In: *The Semantic Web: Research and Applications* Bd. Volume 4011/2006, Springer Berlin / Heidelberg, 2006 (Lecture Notes in Computer Science), S. 348–362

[Chirita u. a. 2005] Chirita, Paul A. ; Gavriloaie, Rita ; Ghita, Stefania ; Nejdl, Wolfgang ; Paiu, Raluca: Activity Based Metadata for Semantic Desktop Search. In: *The Semantic Web: Research and Applications* Bd. Volume 3532/2005, Springer Berlin / Heidelberg, 2005 (Lecture Notes in Computer Science), S. 439–454

[Choudhury u. Phon-Amnuaisuk 2006] Choudhury, Smitashree ; Phon-Amnuaisuk, Somnuk: Semantic Retrieval with Spreading Activation. In: *Proceedings of the MMU International Symposium on Information and Communications Technologies M2USIC 2006, Nov 16 - 17, Kuala Lumpur, Malaysia*, 2006

[Cohen u. Kjeldsen 1987] Cohen, Paul R. ; Kjeldsen, Rick: Information retrieval by constrained spreading activation in semantic networks. In: *Information Processing and Management* 23 (1987), Nr. 4, S. 255–268

[Connolly u. a. 2001] Connolly, Dan ; Harmelen, Frank van ; Horrocks, Ian ; McGuinness, Deborah L. ; Patel-Schneider, Peter F. ; Stein, Lynn A.: *DAML+OIL (March 2001) Reference Description, W3C Note 18 December 2001.* Version: 2001. `http://www.w3.org/TR/daml+oil-reference`, Abruf: 17.02.2008

[Cook u. Campbell 1979] Cook, Thomas D. ; Campbell, Donald T.: *Quasi-*

Experimentation: Design and Analysis Issues for Field Settings. Houghton Mifflin Company, Boston, 1979

[Corby u. a. 2006] Corby, Olivier ; Dieng-Kuntz, Rose ; Faron-Zucker, Catherine ; Gandon, Fabien: Searching the Semantic Web: Approximate Query Processing Based on Ontologies. In: *IEEE Intelligent Systems* 21 (2006), Nr. 1, S. 20–27

[Cormen u. a. 2004] Cormen, Thomas H. ; Leiserson, Charles E. ; Rivest, Ronald L. ; Stein, Clifford: *Algorithmen - Eine Einführung*. Oldenbourg, 2004

[Crestani 1997a] Crestani, Fabio: Application of Spreading Activation Techniques in Information Retrieval. In: *Artificial Intelligence Review* 11 (1997), Nr. 6, S. 453–482

[Crestani 1997b] Crestani, Fabio: Retrieving Documents by Constrained Spreading Activation on Automatically Constructed Hypertexts. In: *EUFIT '97 - 5th European Congress on Intelligent Techniques and Soft Computing*. Germany, 1997

[Crestani u. Lee 2000] Crestani, Fabio ; Lee, Puay L.: Searching the web by constrained spreading activation. In: *Information Processing and Management* 36 (2000), Nr. 4, S. 585–605

[Crestani u. van Rijsbergen 1993] Crestani, Fabio ; Rijsbergen, Cornelis J.: Modelling Adaptive Information Retrieval / Department of Computing Science, University of Glasgow. Glasgow, Scotland, 1993 (IR-93-2). – Departimental Research Report

[Crestani u. van Rijsbergen 1997] Crestani, Fabio ; Rijsbergen, Cornelis J.: A Model for Adaptive Information Retrieval. In: *Journal of Intelligent Information Systems* 8 (1997), Nr. 1, S. 29–56

[Croft 1986] Croft, W. B.: User-specified domain knowledge for document retrieval. In: *SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 1986, S. 201–206

[Croft u. a. 1988] Croft, W. B. ; Lucia, T. J. ; Cohen, Paul R.: Retrieving documents by plausible inference: a priliminary study. In: *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 1988, S. 481–494

[Croft u. Thompson 1987] Croft, W. B. ; Thompson, Roger H.: I3R: a new ap-

proach to the design of document retrieval systems. In: *Journal of the American Society for Information Science* 38 (1987), Nr. 6, S. 389–404

[Crouch u. a. 1994a] Crouch, Carolyn J. ; Crouch, Donald B. ; Nareddy, Krishnamohan: Associative and adaptive retrieval in a connectionist system. In: *International Journal of Expert Systems* 7 (1994), Nr. 2, S. 193–202. – ISSN 0894–9077

[Crouch u. a. 1994b] Crouch, Carolyn J. ; Crouch, Donald B. ; Nareddy, Krishnamohan: A connectionist model for information retrieval based on the vector space model. In: *International Journal of Expert Systems* 7 (1994), Nr. 2, S. 139–163. – ISSN 0894–9077

[Daconta u. a. 2003] Daconta, Michael C. ; Smith, Kevin T. ; Obrst, Leo J.: *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. New York, NY, USA : John Wiley & Sons, Inc., 2003

[Davies u. a. 2003] Davies, John (Hrsg.) ; Fensel, Dieter (Hrsg.) ; Harmelen, Frank van (Hrsg.): *Towards the Semantic Web: Ontology-driven Knowledge Management*. New York, NY, USA : John Wiley & Sons, Inc., 2003

[Davies u. a. 2002] Davies, John ; Krohn, Uwe ; Weeks, Richard: QuizRDF: search technology for the semantic web. In: *WWW2002 workshop on RDF & Semantic Web Applications, 11th International WWW Conference WWW2002*. Hawaii, USA, 2002

[Davies u. Weeks 2004] Davies, John ; Weeks, Richard: QuizRDF: Search Technology for the Semantic Web. In: *40th Hawaii International International Conference on Systems Science (HICSS-37 2004)*. Big Island, HI, USA : IEEE Computer Society, 2004

[Dean u. Schreiber 2004] Dean, Mike ; Schreiber, Guus: *OWL Web Ontology Language Reference, W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/owl-ref/`, Abruf: 17.02.2008

[Decker u. a. 1998] Decker, Stefan ; Erdmann, Michael ; Fensel, Dieter ; Studer, Rudi: Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In: *DS-8: Proceedings of the IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics- Semantic Issues in Multimedia Systems*. Deventer, The Netherlands, The Netherlands : Kluwer, B.V., 1998, S. 351–369

[Decker u. Frank 2004] Decker, Stefan ; Frank, Martin R.: The Networked Semantic Desktop. In: *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004

[Easterbrook u. a. 2008] *Kapitel* Selecting Empirical Methods for Software Engineering Research. In: EASTERBROOK, Steve ; SINGER, Janice ; STOREY, Margaret-Anne ; DAMIAN, Daniela: *Guide to Advanced Empirical Software Engineering*. Springer, 2008

[Eastlake u. Reagle 2002] EASTLAKE, Donald ; REAGLE, Joseph: *XML Encryption Syntax and Processing, W3C Recommendation 10 December 2002*. Version: 2002. `http://www.w3.org/TR/xmlenc-core/`, Abruf: 17.02.2008

[Euzenat u. Shvaiko 2007] EUZENAT, Jérôme ; SHVAIKO, Pavel: *Ontology matching*. Heidelberg (DE) : Springer-Verlag, 2007. – ISBN 3–540–49611–4

[Fensel u. Harmelen 2007] FENSEL, Dieter ; HARMELEN, Frank V.: Unifying Reasoning and Search to Web Scale. In: *IEEE Internet Computing* 11 (2007), March/April, Nr. 2, S. 94–96

[Fensel u. a. 2005] FENSEL, Dieter ; HENDLER, James A. ; LIEBERMAN, Henry ; WAHLSTER, Wolfgang: *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. The MIT Press, 2005

[Ferber 2003] FERBER, Reginald: *Information Retrieval: Suchmodelle und Data-Mining-Verfahren für Textsammlungen und das Web*. Heidelberg : dpunkt Verlag, 2003

[Finin u. a. 2005] FININ, Tim ; MAYFIELD, James ; FINK, Clay ; JOSHI, Anupam ; COST, R. S.: Information Retrieval and the Semantic Web. In: *Proceedings of the 38th International Conference on System Sciences*, 2005. – Received Best mini-track paper award.

[Fowler 2004] FOWLER, Martin: *Inversion of Control Containers and the Dependency Injection pattern*. Version: 2004. `http://www.martinfowler.com/articles/injection.html(01.09.2007)`, Abruf: 17.02.2008

[Fuhr 2006] FUHR, Norbert: *Information Retrieval: Skriptum zur Vorlesung im SS 06, 19. Dezember 2006*. 2006

[Gelgi u. a. 2005] GELGI, Fatih ; VADREVU, Srinivas ; DAVULCU, Hasan: Improving Web Data Annotations with Spreading Activation. In: *Web Information Systems Engineering - WISE 2005, 6th International Conference on Web Information Systems Engineering, New York, NY, USA, November 20-22, 2005, Proceedings*, 2005, S. 95–106

[Gerber u. a. 2006] GERBER, Aurona ; MERWE, Alta van d. ; BARNARD, Andries: Semantic Web Technologies / University South of Africa, School of Computing. 2006 (UNISA-TR-2006-02). – Forschungsbericht

[Ghidini u. a. 2007] Ghidini, Chiara ; Pammer, Viktoria ; Scheir, Peter ; Lindstaedt, Stefanie ; Serafini, Luciano: APOSDLE: learn@work with semantic web technology. In: *I-SEMANTICS 2007*, 2007, S. 262–269

[Ginsberg u. a. 2006] Ginsberg, Allen ; Hirtle, David ; McCabe, Frank ; Patranjan, Paula-Lavinia: *RIF Use Cases and Requirements, W3C Working Draft 10 July 2006*. Version: 2006. `http://www.w3.org/TR/rif-ucr/`, Abruf: 17.02.2008

[Granitzer 2006] Granitzer, Michael: *KnowMiner: Konzeption und Entwicklung eines generischen Wissenserschliessungsframeworks*, Technische Universität Graz, Diss., 2006

[Gruber] Gruber, Tom R.: *What is an Ontology?* `http://www-ksl.stanford.edu/kst/what-is-an-ontology.html`, Abruf: 17.02.2008

[Gruber 1993] Gruber, Tom R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N. (Hrsg.) ; Poli, R. (Hrsg.): *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Deventer, The Netherlands : Kluwer Academic Publishers, 1993

[Guha u. a. 2003] Guha, R. ; McCool, Rob ; Miller, Eric: Semantic search. In: *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA : ACM Press, 2003, S. 700–709

[Handschuh 2005] Handschuh, Siegfried: *Creating Ontology-based Metadata by Annotation for the Semantic Web*, Universität Fridericiana zu Karlsruhe, Diss., 2005

[Handschuh 2007] *Kapitel* Semantic Annotation of Resources in the Semantic Web. In: Handschuh, Siegfried: *Semantic Web Services: Concepts, Technologies, and Applications*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2007. – ISBN 3540708936, S. 135–155

[Handschuh u. Staab 2003] Handschuh, Siegfried (Hrsg.) ; Staab, Steffen (Hrsg.): *Frontiers in Artificial Intelligence and Applications*. Bd. 96: *Annotation for the Semantic Web*. IOS Press, 2003

[Hartmann 2001] Hartmann, Knut: *Text-Bild-Beziehungen in multimedialen Dokumenten: Eine Analyse aus Sicht von Wissensrepräsentation, Textstruktur und Visualisierung*, Otto-von-Guericke-Universität Magdeburg, Diss., 2001

[Hartmann u. a. 2002] Hartmann, Knut ; Schlechtweg, Stefan ; Helbing, Ralf ; Strothotte, Thomas: Knowledge-Supported Graphical Illustration of Texts. In: De Marsico, Maria (Hrsg.) ; Levialdi, Stefano (Hrsg.) ; Panizzi, Emanuele

(Hrsg.): *Proc. of the Working Conference on Advanced Visual Interfaces (AVI 2002; 22.–24. May 2002, Trento, Italy)*. New York : ACM Press, 2002, S. 300–307

[Hartmann u. Strothotte 2002] Hartmann, Knut ; Strothotte, Thomas: A spreading activation approach to text illustration. In: *SMARTGRAPH '02: Proceedings of the 2nd international symposium on Smart graphics*. New York, NY, USA : ACM Press, 2002, S. 39–46

[Hayes 2004] Hayes, Patrick: *RDF Semantics, W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/rdf-mt/`, Abruf: 17.02.2008

[Heflin u. Hendler 2000] Heflin, Jeff ; Hendler, James: Searching the Web with SHOE. In: *Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01.* Menlo Park, CA : AAAI Press, 2000, S. 35–40

[Hendler 2007] Hendler, James: The Dark Side of the Semantic Web. In: *IEEE Intelligent Systems* 22 (2007), Nr. 1, S. 2–4

[Hendler 2004] Hendler, Jim: *Frequently Asked Questions on W3C's Web Ontology Language (OWL)*. Version: 2004. `http://www.w3.org/2003/08/owlfaq`, Abruf: 17.02.2008

[Henninger 1995] Henninger, Scott: Information access tools for software reuse. In: *Journal of Systems and Software* 30 (1995), Nr. 3, S. 231–247

[Henninger 1996] Henninger, Scott: Supporting the construction and evolution of component repositories. In: *ICSE '96: Proceedings of the 18th international conference on Software engineering*. Washington, DC, USA : IEEE Computer Society, 1996, S. 279–288

[Henninger 1997] Henninger, Scott: An evolutionary approach to constructing effective software reuse repositories. In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 6 (1997), Nr. 2, S. 111–140

[Heylighen 2001] Heylighen, Francis: Bootstrapping knowledge representations: from entailment meshes via semantic nets to learning webs. In: *Kybernetes* 30 (2001), Nr. 5/6, S. 691–722

[Heylighen u. Bollen 2002] Heylighen, Francis ; Bollen, Johan: Hebbian Algorithms for a Digital Library Recommendation System. In: *31st International Conference on Parallel Processing Workshops (ICPP 2002 Workshops), 20-23 August 2002, Vancouver, BC, Canada*, 2002, S. 439–446

[Hitzler u. a. 2008] Hitzler, Pascal ; Krötzsch, Markus ; Rudolph, Sebastian ; Sure, York: *Semantic Web - Grundlagen*. Springer-Verlag Berlin Heidelberg, 2008 (eXamen.press)

[Horrocks u. a. 2003] Horrocks, Ian ; Patel-Schneider, Peter F. ; Harmelen, Frank van: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. In: *Journal of Web Semantics* 1 (2003), Nr. 1, S. 7–26

[Huang u. a. 2004a] Huang, Zan ; Chen, Hsinchun ; Zeng, Daniel: Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. In: *ACM Transactions on Information Systems* 22 (2004), Nr. 1, S. 116–142

[Huang u. a. 2004b] Huang, Zan ; Chung, Wingyan ; Chen, Hsinchun: A graph model for E-commerce recommender systems. In: *Journal of the American Society for Information Science and Technology (JASIST)* 55 (2004), Nr. 3, S. 259–274

[Huang u. a. 2002] Huang, Zan ; Chung, Wingyan ; Ong, Thian-Huat ; Chen, Hsinchun: A graph-based recommender system for digital library. In: *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2002, Portland, Oregon, USA, June 14-18, 2002*, 2002, S. 65–73

[Iofciu u. a. 2005] Iofciu, Tereza ; Kohlschütter, Christian ; Nejdl, Wolfgang ; Paiu, Raluca: Keywords and RDF Fragments: Integrating Metadata and Full-Text Search in Beagle++. In: Decker, Stefan (Hrsg.) ; Park, Jack (Hrsg.) ; Quan, Dennis (Hrsg.) ; Sauermann, Leo (Hrsg.): *Proceedings of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6* Bd. 175, 2005

[Jiang u. Tan 2006] Jiang, Xing ; Tan, Ah-Hwee: OntoSearch: A Full-Text Search Engine for the Semantic Web. In: *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, 2006

[Jones u. Furnas 1987] Jones, William P. ; Furnas, George W.: Pictures of relevance: a geometric analysis of similarity measures. In: *Journal of the American Society for Information Science* 38 (1987), Nr. 6, S. 420–442

[Kifer u. a. 1995] Kifer, Michael ; Lausen, Georg ; Wu, James: Logical foundations of object-oriented and frame-based languages. In: *Journal of the ACM* 42 (1995), Nr. 4, S. 741–843

[Kiryakov u. a. 2003] Kiryakov, Atanas ; Popov, Borislav ; Ognyanoff, Damyan ; Manov, Dimitar ; Kirilov, Angel ; Goranov, Miroslav: Semantic Annotation, Indexing, and Retrieval. In: *International Semantic Web Conference*, 2003, S. 484–499

[Kiryakov u. a. 2004] Kiryakov, Atanas ; Popov, Borislav ; Terziev, Ivan ; Ma-

NOV, Dimitar ; OGNYANOFF, Damyan: Semantic annotation, indexing, and retrieval. In: *Journal of Web Semantics: Science, Services and Agents on the World Wide We* 2 (2004), Nr. 1, S. 49–79

[Kjeldsen u. Cohen 1987] KJELDSEN, Rick ; COHEN, Paul R.: The Evolution and Performance of the GRANT System. In: *IEEE Expert* 2 (1987), Nr. 2, S. 73–79

[Kleinberg 1999] KLEINBERG, Jon M.: Authoritative sources in a hyperlinked environment. In: *Journal of the ACM* 46 (1999), Nr. 5, S. 604–632

[Klyne u. Carroll 2004] KLYNE, Graham ; CARROLL, Jeremy J.: *Resource Description Framework (RDF): Concepts and Abstract Syntax, W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/rdf-concepts/`, Abruf: 17.02.2008

[Kogan u. a. 1998] KOGAN, Yakov ; MICHAELI, David ; SAGIV, Yehoshua ; SHMUELI, Oded: Utilizing the multiple facets of WWW contents. In: *Data & Knowledge Engineering* 28 (1998), Nr. 3, S. 255–275

[Kopena u. Regli 2003] KOPENA, Joseph ; REGLI, William: DAMLJessKB: A Tool for Reasoning with the Semantic Web. In: *IEEE Intelligent Systems* 18 (2003), May/June, Nr. 3, S. 74–77

[Kuropka 2004] KUROPKA, Dominik: *Advances in Information Systems and Management Science*. Bd. 10: *Modelle zur Repräsentation natürlichsprachlicher Dokumente - Information-Filtering und -Retrieval mit relationalen Datenbanken*. Berlin : Logos Verlag, 2004

[Lee 1997] LEE, Joon H.: Analyses of multiple evidence combination. In: *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 1997, S. 267–276

[Ley u. a. 2007] LEY, T. ; ULBRICH, A. ; SCHEIR, P. ; LINDSTAEDT, S. N. ; KUMP, B. ; ALBERT, D.: Modelling Competencies for Supporting Work-integrated Learning in Knowledge Work. In: *Journal of Knowledge Management* in press (2007)

[Lindstaedt u. a. 2008] LINDSTAEDT, Stefanie N. ; LEY, Tobias ; SCHEIR, Peter ; ULBRICH, Armin: Applying 'Scruffy' Methods to Enable Work-integrated Learning. In: *to appear in The European Journal for the Informatics Professional* (2008)

[Lindstaedt u. Mayer 2006] LINDSTAEDT, Stefanie N. ; MAYER, Harald: A Storyboard of the APOSDLE Vision. In: *Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Lear-*

ning, EC-TEL 2006, Crete, Greece, October 1-4, 2006, 2006, S. 628–633

[Lindstaedt u. a. 2007] Lindstaedt, Stefanie N. ; Scheir, Peter ; Ulbrich, Armin: Scruffy Technologies to Enable (Work-integrated) Learning. In: Wolpers, Martin (Hrsg.) ; Klamma, Ralf (Hrsg.) ; Duval, Erik (Hrsg.): *Proceedings of the EC-TEL 2007 Poster Session, Crete, Greece, September 17-20, 2007*, 2007

[Liu 2007] Liu, Jian-Guo: A spreading activation approach for collaborative filtering. In: *The Computing Research Repository (CoRR)* abs/0712.3807 (2007)

[Lux 2006] Lux, Mathias: *Semantische Metadaten - Ein Modell für den Bereich zwischen Metadaten und Ontologien*, Technische Universität Graz, Diss., 2006

[Maedche u. a. 2003] Maedche, Alexander ; Motik, Boris ; Stojanovic, Ljiljana ; Studer, Rudi ; Volz, Raphael: Ontologies for Enterprise Knowledge Management. In: *IEEE Intelligent Systems* 18 (2003), Nr. 2, S. 26–33

[Maedche u. Staab 2002] Maedche, Alexander ; Staab, Steffen: Measuring Similarity between Ontologies. In: *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*. London, UK : Springer-Verlag, 2002, S. 251–263

[Mandl 2001] Mandl, Thomas: *Tolerantes Information Retrieval. Neuronale Netze zur Erhöhung der Adaptivität und Flexibilität bei der Informationssuche*, University Of Hildesheim, Diss., 2001

[Manola u. Miller 2004] Manola, Frank ; Miller, Eric: *RDF Primer, W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/rdf-primer/`, Abruf: 17.02.2008

[Mayfield u. Finin 2003] Mayfield, James ; Finin, Tim: Information retrieval on the Semantic Web: Integrating inference and retrieval. In: *Proceedings of the SIGIR Workshop on the Semantic Web*, 2003

[McCool 2005] McCool, Rob: Rethinking the Semantic Web, Part 1. In: *IEEE Internet Computing* 9 (2005), Nr. 6, S. 88–87

[McGuinness 2003] McGuinness, Deborah L.: Ontologies Come of Age. In: *Spinning the Semantic Web*. MIT Press, 2003, S. 171–194

[McGuinness u. Harmelen 2004] McGuinness, Deborah L. ; Harmelen, Frank V.: *OWL Web Ontology Language Overview, W3C Recommendation 10 February 2004*. Version: 2004. `http://www.w3.org/TR/owl-features/`, Abruf: 17.02.2008

[Medler 1998] Medler, David A.: A Brief History of Connectionism. In: *Neural*

Computing Surveys 1 (1998), S. 61–101

[Montaner u. a. 2003] Montaner, Miquel ; López, Beatriz ; Rosa, Josep L.: A Taxonomy of Recommender Agents on the Internet. In: *Artificial Intelligence Review* 19 (2003), Nr. 4, S. 285–330

[Mothe 1994] Mothe, Josiane: Search Mechanisms Using a Neural Network Model, Comparison with the Vector Space Model. In: *RIAO 94 (Recherche d'Information assistée par Ordinateur), Intelligent Multimedia Information Retrieval Systems and Management*. New York, 1994, S. 275–294

[MySQL AB 2008] *Kapitel* How MySQL Uses Indexes. In: MySQL AB: *MySQL 5.0 Reference Manual*. 2008. – `http://dev.mysql.com/doc/refman/5.0/en/index.html` (11.01.2008)

[Noy u. McGuinness 2001] Noy, Natalya F. ; McGuinness, Deborah L.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001

[Object Management Group 2007] Object Management Group: *Unified Modeling Language: Superstructure version 2.1.1*. 2007. – formal/07-02-05

[Obrst 2005] Obrst, Leo: *Re: [ontolog-forum] Updating the Ontolog Charter ... definitions andUBL*. Version: 2005. `http://ontolog.cim3.net/forum/ontolog-forum/2005-03/msg00005.html`, Abruf: 17.02.2008

[O'Hara u. a. 2002] O'Hara, Kieron ; Alani, Harith ; Shadbolt, Nigel: Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition. In: *Proceedings of IFIP*, 2002

[O'Riordan u. Sorensen 1995] O'Riordan, Adrian ; Sorensen, Humphrey: An Intelligent Agent for High-Precision Text Filtering. In: *CIKM '95, Proceedings of the 1995 International Conference on Information and Knowledge Management, November 28 - December 2, 1995, Baltimore, Maryland, USA*, 1995, S. 205–211

[Pammer u. a. 2007] Pammer, Viktoria ; Scheir, Peter ; Lindstaedt, Stefanie: Two Protégé plug-ins for supporting document-based ontology engineering and ontological annotation at document level. In: *10th International Protégé Conference - July 15-18, 2007 - Budapest, Hungary*, 2007

[Passin 2004] Passin, Thomas B.: *Explorer's Guide to the Semantic Web*. Greenwich, CT, USA : Manning Publications Co., 2004

[Patel-Schneider u. a. 2004] Patel-Schneider, Peter F. ; Hayes, Patrick ; Horrocks, Ian: *OWL Web Ontology Language Semantics and Abstract Syntax,*

W3C Recommendation 10 February 2004. Version: 2004. http://www.w3.org/TR/owl-semantics/, Abruf: 17.02.2008

[Pirolli u. a. 1996] Pirolli, Peter ; Pitkow, James ; Rao, Ramana: Silk from a sow's ear: extracting usable structures from the Web. In: *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 1996, S. 118–125

[Popov u. a. 2003] Popov, Borislav ; Kiryakov, Atanas ; Kirilov, Angel ; Manov, Dimitar ; Ognyanoff, Damyan ; Goranov, Miroslav: KIM - Semantic Annotation Platform. In: *International Semantic Web Conference*, 2003, S. 834–849

[Preece 1981] Preece, Scott E.: *A spreading activation network model for information retrieval*, University of Illinois at Urbana-Champaign, Diss., 1981

[Priebe u. a. 2004] Priebe, Torsten ; Schlager, Christian ; Pernul, Gunther: A Search Engine for RDF Metadata. In: *DEXA '04: Proceedings of the Database and Expert Systems Applications, 15th International Workshop on (DEXA'04)*. Washington, DC, USA : IEEE Computer Society, 2004, S. 168–172

[Prud'hommeaux u. Seaborne 2008] Prud'hommeaux, Eric ; Seaborne, Andy: *SPARQL Query Language for RDF, W3C Recommendation 15 January 2008*. Version: 2008. http://www.w3.org/TR/rdf-sparql-query/, Abruf: 17.02.2008

[Quillian 1968] Quillian, M. R.: Semantic Memory. In: Minsky, Marvin (Hrsg.): *Semantic Information Processing*. MIT Press, 1968, S. 227–270

[Reeve u. Han 2005] Reeve, Lawrence ; Han, Hyoil: Survey of semantic annotation platforms. In: *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. New York, NY, USA : ACM Press, 2005, S. 1634–1638

[Reif 2006] Reif, Gerald: Semantische Annotation. In: Pellegrini, Tassilo (Hrsg.) ; Blumauer, Andreas (Hrsg.): *Semantic Web - Wege zur vernetzten Wissensgesellschaft*. Springer Verlag, 2006, S. 405–418

[Resnik 1999] Resnik, Philip: Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. In: *Journal of Artificial Intelligence Res. (JAIR)* 11 (1999), S. 95–130

[Rijsbergen 1979] Rijsbergen, C. J. v.: *Information retrieval*. 2. London : Butterworths, 1979 http://www.dcs.glasgow.ac.uk/Keith/Preface.html

[Robertson u. Spärck Jones 1994] Robertson, S.E. ; Spärck Jones, K.: Simple, proven approaches to text retrieval / University of Cambridge, Computer Laboratory. 1994 (UCAM-CL-TR-356). – Forschungsbericht

[Rocha u. a. 2004a] Rocha, Cristiano ; Schwabe, Daniel ; Aragão, Marcus P.: A hybrid approach for searching in the semantic web. In: *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004*, 2004, S. 374–383

[Rocha u. a. 2004b] Rocha, Cristiano ; Schwabe, Daniel ; Aragão, Marcus P.: Integrating Semantic Concept Similarity in Model-Based Web Applications. In: *Joint Conference 10th Brazilian Symposium on Multimedia and the Web & 2nd Latin American Web Congress, (WebMedia & LA-Web 2004), 12-15 October 2004, Ribeirao Preto-SP,Brazil*, 2004, S. 78–85

[Rose u. Belew 1989] Rose, D. E. ; Belew, Richard K.: Legal information retrieval a hybrid approach. In: *ICAIL '89: Proceedings of the 2nd international conference on Artificial intelligence and law*. New York, NY, USA : ACM Press, 1989, S. 138–146

[Rose u. Belew 1991] Rose, Daniel E. ; Belew, Richard K.: A connectionist and symbolic hybrid for improving legal research. In: *International Journal of Man-Machine Studies* 35 (1991), Nr. 1, S. 1–33

[Sabou u. a. 2006] Sabou, Marta ; d'Aquin, Mathieu ; Motta, Enrico: Using the Semantic Web as Background Knowledge for Ontology Mapping. In: *International Workshop on Ontology Matching (OM-2006)*, 2006

[Salton 1963] Salton, Gerard: Associative Document Retrieval Techniques Using Bibliographic Information. In: *Journal of the ACM* 10 (1963), Nr. 4, S. 440–457

[Salton 1968] Salton, Gerard: *Automatic Information Organization and Retrieval*. McGraw Hill, 1968

[Salton u. Buckley 1988a] Salton, Gerard ; Buckley, Chris: On the Use of Spreading Activation Methods in Automatic Information Retrieval. In: Chiaramella, Yves (Hrsg.): *SIGIR'88, Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Grenoble, France, June 13-15, 1988*, ACM, 1988, S. 147–160

[Salton u. Buckley 1988b] Salton, Gerard ; Buckley, Christopher: Term-weighting approaches in automatic text retrieval. In: *Information Processing and Management* 24 (1988), Nr. 5, S. 513–523. `http://dx.doi.org/http://dx.doi.org/10.1016/0306-4573(88)90021-0`. – DOI http://dx.doi.org/10.1016/0306–4573(88)90021–0

[Sauermann u. a. 2005] Sauermann, Leo ; Bernardi, Ansgar ; Dengel, Andreas: Overview and Outlook on the Semantic Desktop. In: *Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference*, 2005

[Savell 2005] Savell, Robert A.: *On-line Metasearch, Pooling, and System Evaluation*, Department of Computer Science, Dartmouth College, Diss., 2005. – Technical Report TR2005-543

[Scheir 2006] Scheir, Peter: Associative retrieval of resources for work-integrated learning: Integrating domain knowledge with content-based similarities. In: Maillet, Katherine (Hrsg.) ; Klamma, Ralf (Hrsg.): *Proceedings of the 1st Doctoral Consortium in Technology Enhanced Learning, Crete, Greece, October 2, 2006*, 2006, S. 72–81. – http://www.prolearn-academy.org/Academy

[Scheir u. a. 2007a] Scheir, Peter ; Ghidini, Chiara ; Lindstaedt, Stefanie N.: Improving Search on the Semantic Desktop using Associative Retrieval Techniques. In: *Proceedings of I-SEMANTICS 2007*, 2007, S. 221–228

[Scheir u. a. 2007b] Scheir, Peter ; Granitzer, Michael ; Lindstaedt, Stefanie N.: Evaluation of an Information Retrieval System for the Semantic Desktop using Standard Measures from Information Retrieval. In: *LWA 2007, Lernen - Wissensentdeckung - Adaptivität, 24.-26.9. 2007 in Halle/Saale*, 2007, S. 269–272

[Scheir u. a. 2006a] Scheir, Peter ; Hofmair, Philip ; Granitzer, Michael ; Lindstaedt, Stefanie N.: The OntologyMapper plug-in: Supporting Semantic Annotation of Text-Documents by Classification. In: Schaffert, Sebastian (Hrsg.) ; Sure, York (Hrsg.): *Semantic Systems From Vision to Applications - Proceedings of the SEMANTICS 2006*, Österreichische Computer Gesellschaft, 2006 (books@ocg.at 212), S. 291–301

[Scheir u. Lindstaedt 2006] Scheir, Peter ; Lindstaedt, Stefanie N.: A network model approach to document retrieval taking into account domain knowledge. In: Schaaf, Martin (Hrsg.) ; Althoff, Klaus-Dieter (Hrsg.): *LWA 2006, Lernen - Wissensentdeckung - Adaptivität, 9.-11.10.2006 in Hildesheim*, Universität Hildesheim, 2006 (Hildesheimer Informatik-Berichte 1/2006), S. 154–158

[Scheir u. a. 2005a] Scheir, Peter ; Lindstaedt, Stefanie N. ; Hofmair, Philip ; Mittendorfer, Georg: SemaTech - Management Summary / Know-Center, Graz. 2005. – Forschungsbericht

[Scheir u. a. 2005b] Scheir, Peter ; Lux, Mathias ; Lindstaedt, Stefanie: *Associative Retrieval over different Knowledge Organisation Systems.* Vienna : 4th European Networked Knowledge Organization Systems (NKOS) Workshop, ED-CL 2005, 2005

[Scheir u. a. 2007c] Scheir, Peter ; Pammer, Viktoria ; Lindstaedt, Stefanie N.: Information Retrieval on the Semantic Web - Does it exist? In: *LWA 2007, Lernen - Wissensentdeckung - Adaptivität, 24.-26.9. 2007 in Halle/Saale*, 2007, S. 252–257

[Scheir u. a. 2006b] Scheir, Peter ; Pammer, Viktoria ; Lindstaedt, Stefanie N. ; Hofmair, Philip ; Schlatte, Rufolf ; Mayer, Harald: Ontology Learning - Management Summary / Know-Center, Graz. 2006. – Forschungsbericht

[Seaborne 2004] Seaborne, Andy: *RDQL - A Query Language for RDF, W3C Member Submission 9 January 2004.* Version: 2004. `http://www.w3.org/Submission/RDQL/`, Abruf: 04.02.2007

[Sedgewick u. Schidlowsky 2003] Sedgewick, Robert ; Schidlowsky, Michael: *Algorithms in Java, Part 5: Graph Algorithms.* Addison-Wesley, 2003. – ISBN 0201361213

[Shah u. a. 2002] Shah, Urvi ; Finin, Tim ; Joshi, Anupam: Information retrieval on the semantic web. In: *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management.* New York, NY, USA : ACM Press, 2002, S. 461–468

[Sharifian u. Samani 1997] Sharifian, Farzad ; Samani, Ramin: Hierarchical spreading of activation. In: Sharifian, Farzad (Hrsg.): *Proc. of the Conference on Language, Cognition, and Interpretation*, IAU Press, 1997, S. 1–10

[Sharples u. a. 1989] Sharples, Mike ; Hogg, David ; Hutchison, Chris ; Torrance, Steve ; Young, David: *Computers and Thought: A Practical Introduction to Artificial Intelligence.* The MIT Press, 1989

[Sheth u. a. 2002] Sheth, Amit ; Bertram, Clemens ; Avant, David ; Hammond, Brian ; Kochut, Krysztof ; Warke, Yashodhan: Managing Semantic Content for the Web. In: *IEEE Internet Computing* 6 (2002), Nr. 4, S. 80–87

[Smith u. a. 2004] Smith, Michael K. ; Welty, Chris ; McGuinness, Deborah L.: *OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004.* Version: 2004. `http://www.w3.org/TR/owl-guide/`, Abruf: 17.02.2008

[Song u. a. 2005] Song, Jun-feng ; Zhang, Wei-ming ; Xiao, Wei-dong ; Li, Guo-hui ; Xu, Zhen-ning: Ontology-Based Information Retrieval Model for the Semantic Web. In: *2005 IEEE International Conference on e-Technology, e-Commerce, and e-Services (EEE 2005), 29 March - 1 April 2005, Hong Kong, China*, 2005, S. 152–155

[Soo u. a. 2003] Soo, Von-Wun ; Lee, Chen-Yu ; Li, Chung-Cheng ; Chen, Shu L. ; Chen, Ching chih: Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques. In: *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries.* Washington, DC, USA : IEEE Computer Society, 2003, S. 61–72

[Sowa 2000] Sowa, John F.: *Knowledge representation: logical, philosophical and computational foundations*. Pacific Grove, CA, USA : Brooks/Cole Publishing Co., 2000. – ISBN 0–534–94965–7

[Spärck Jones 2004] Spärck Jones, K.: What's new about the Semantic Web?: some questions. In: *SIGIR Forum* 38 (2004), S. 18–23

[Stojanovic u. a. 2001] Stojanovic, Nenad ; Maedche, Alexander ; Staab, Steffen ; Studer, Rudi ; Sure, York: SEAL: a framework for developing SEmantic PortALs. In: *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), October 21-23, 2001, Victoria, BC, Canada*, 2001, S. 155–162

[Stojanovic u. a. 2003] Stojanovic, Nenad ; Studer, Rudi ; Stojanovic, Ljiljana: An Approach for the Ranking of Query Results in the Semantic Web. In: *International Semantic Web Conference*, 2003, S. 500–516

[The Unicode Consortium 2006] The Unicode Consortium: *Unicode Standard, Version 5.0, The (5th Edition)*. Addison-Wesley Professional, 2006

[Tsatsaronis u. a. 2007] Tsatsaronis, George ; Vazirgiannis, Michalis ; Androutsopoulos, Ion: Word Sense Disambiguation with Spreading Activation Networks Generated from Thesauri. In: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 2007, S. 1725–1730

[Ulbrich u. a. 2006] Ulbrich, Armin ; Scheir, Peter ; Lindstaedt, Stefanie N. ; Görtz, Manuel: A Context-Model for Supporting Work-Integrated Learning. In: Nejdl, Wolfgang (Hrsg.) ; Tochtermann, Klaus (Hrsg.): *Innovative Approaches for Learning and Knowledge Sharing - First European Conference on Technology Enhanced Learning, EC-TEL 2006, Crete, Greece, October 1-4, 2006* Bd. 4227, Springer, 2006 (Lecture Notes in Computer Science), S. 525–530

[Uren u. a. 2006] Uren, Victoria S. ; Cimiano, Philipp ; Iria, José ; Handschuh, Siegfried ; Vargas-Vera, Maria ; Motta, Enrico ; Ciravegna, Fabio: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. In: *Journal of Web Semantics* 4 (2006), Nr. 1, S. 14–28

[Uschold u. Grüninger 1996] Uschold, Mike ; Grüninger, Michael: Ontologies: principles, methods, and applications. In: *Knowledge Engineering Review* 11 (1996), Nr. 2, S. 93–155

[Voorhees 2000] Voorhees, Ellen M.: Variations in relevance judgments and the measurement of retrieval effectiveness. In: *Information Processing and Management* 36 (2000), Nr. 5, S. 697–716. – ISSN 0306–4573

[Voorhees 2001] Voorhees, Ellen M.: Evaluation by highly relevant documents. In: *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 2001, S. 74–82

[Wilkinson u. Hingston 1991] Wilkinson, Ross ; Hingston, Philip: Using the cosine measure in a neural network for document retrieval. In: *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA : ACM Press, 1991, S. 202–210

[Wilkinson u. Hingston 1992] Wilkinson, Ross ; Hingston, Philip: Incorporating the vector space model in a neural network used for document retrieval. In: *Library Hi Tech* 10 (1992), Nr. 1-2, S. 69–75

[Wolverton 1995] Wolverton, Michael: An Investigation of Marker-Passing Algorithms for Analogue Retrieval. In: *ICCBR '95: Proceedings of the First International Conference on Case-Based Reasoning Research and Development*. London, UK : Springer-Verlag, 1995. – ISBN 3–540–60598–3, S. 359–370

[Wolverton u. Hayes-Roth 1994] Wolverton, Michael ; Hayes-Roth, Barbara: Retrieving semantically distant analogies with knowledge-directed spreading activation. In: *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*. Menlo Park, CA, USA : American Association for Artificial Intelligence, 1994. – ISBN 0–262–61102–3, S. 56–61

[Wolverton u. Hayes-Roth 1995] Wolverton, Michael ; Hayes-Roth, Barbara: Finding Analogues for Innovative Design. In: *Third International Round-Table Conference on Computational Models of Creative Design*, 1995

[Wolverton 1994] Wolverton, Michael J.: *Retrieving semantically distant analogies*, Stanford University, Diss., 1994. – Adviser-Barbara Hayes-Roth

[Woodruff u. a. 2000] Woodruff, Allison ; Gossweiler, Rich ; Pitkow, James ; Chi, Ed H. ; Card, Stuart K.: Enhancing a digital book with a reading recommender. In: *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA : ACM Press, 2000, S. 153–160

[Wu u. Palmer 1994] Wu, Zhibiao ; Palmer, Martha S.: Verb Semantics and Lexical Selection. In: *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL 1994), 27-30 June 1994, New Mexico State University, Las Cruces, New Mexico, USA*, Morgan Kaufmann Publishers, 1994, S. 133–138

[Yilmaz u. Aslam 2006] Yilmaz, Emine ; Aslam, Javed A.: Estimating average precision with incomplete and imperfect judgments. In: *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*. New York, NY, USA : ACM Press, 2006, S. 102–111

[Zhang u. a. 2005] Zhang, Lei ; Yu, Yong ; Zhou, Jian ; Lin, ChenXi ; Yang, Yin: An enhanced model for searching in semantic portals. In: *WWW '05: Proceedings of the 14th international conference on World Wide Web*. New York, NY, USA : ACM Press, 2005. – ISBN 1–59593–046–9, S. 453–462

[Ziegler u. a. 2006] Ziegler, Patrick ; Kiefer, Christoph ; Sturm, Christoph ; Dittrich, Klaus R. ; Bernstein, Abraham: Detecting Similarities in Ontologies with the SOQA-SimPack Toolkit. In: Ioannidis, Yannis (Hrsg.) ; Scholl, Marc H. (Hrsg.) ; Schmidt, Joachim W. (Hrsg.) ; Matthes, Florian (Hrsg.) ; Hatzopoulos, Mike (Hrsg.) ; Boehm, Klemens (Hrsg.) ; Kemper, Alfons (Hrsg.) ; Grust, Torsten (Hrsg.) ; Boehm, Christian (Hrsg.): *10th International Conference on Extending Database Technology (EDBT 2006)* Bd. 3896. Munich, Germany, March 26-31 : Springer, 2006 (Lecture Notes in Computer Science), S. 59–76

Index