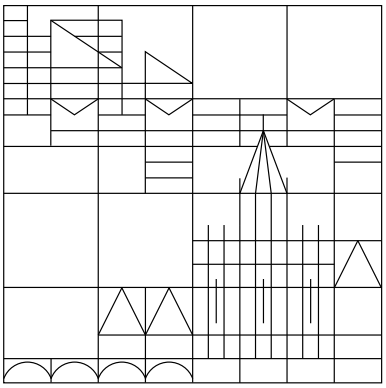
Universität Konstanz



Fachbereich für Informatik und Informationswissenschaft
ALTANA Stiftungs-Lehrstuhl für Angewandte Informatik
Bioinformatik und Information Mining

Masterarbeit

Assoziative Analyse von heterogenen Datenquellen

zur Erlangung des akademischen Grades eines
Master of Science (M.Sc.)

Kilian Thiel

12. September 2006

Gutachter:

Prof. Dr. M. Berthold, Prof. Dr. R. Kuhlen

Universität Konstanz
Fachbereich für Informatik und Informationswissenschaft
D–78457 Konstanz
Deutschland

Thiel, Kilian:
Assoziative Analyse von heterogenen Datenquellen
Masterarbeit, Universität Konstanz, 2006.

Zusammenfassung

Thema:	Assoziative Analyse von heterogenen Datenquellen
Student:	Kilian Thiel Zasiusstrasse 8 78462 Konstanz
Ort:	Universität Konstanz (intern)
Betreuer:	Professor Dr. M. Berthold, Universität Konstanz Thorsten Meinl, Universität Konstanz
Schlagworte:	maschinelles Lernen, heterogene Datenquellen, assoziative Netzwerke Exploration von Texträumen, verbindungsorientierte Modelle, spreading activation, Branch-and-Bound-Algorithmus gene subgroup mining, Genexpressionsdaten

Um umfangreiche Informationen zu einem bestimmten Thema zu erhalten, ist es oft notwendig, in verschiedenen Datenquellen zu recherchieren. Im Falle von Informationen über bestimmte Gene und deren Zusammenhänge mit anderen Genen ist es z.B. nützlich, wissenschaftliche Artikel über diese zu lesen und zusätzlich Genexpressionsdaten und Genontologien zu durchsuchen. Der Prozess der Suche nach relevanten Informationen ist unter Umständen sehr aufwändig. Ein System, das Informationen aus heterogenen Datenquellen erfasst, diese untereinander vernetzt und zu Anfragen relevante Ergebnisse liefert, würde die Suche nach bestimmten Informationen äußerst erleichtern. Assoziative Netze können für eine derartige Aufgabe genutzt werden. Sie bestehen aus Informationsknoten und Verbindungen, die Beziehungen zwischen Informationseinheiten abbilden. Die Grundarchitektur eines solchen Netzes ist der des menschlichen Kortex nachempfunden.

Im Rahmen dieser Masterarbeit wurde ein assoziatives Netz entwickelt, welches durch die Verknüpfung von Wörtern aus wissenschaftlichen Publikationen und Genexpressionsdaten, Beziehungen zwischen diesen repräsentiert. Somit werden Informationen aus heterogenen Datenquellen vernetzt und bestehende Beziehungen können einheitlich analysiert und erkannt werden.

Weiter wurde ein Verfahren entworfen und implementiert, mit welchem es möglich ist, ein bestehendes Netz nach bestimmten Anfragetermen zu durchsuchen und daraufhin verwandte Terme und Gennamen sowie Dokumente oder Experimente mit Genexpressionsdaten, in welchen die Terme, bzw. Gennamen vorkommen, zurückzuliefern. Die Funktionsweise des Verfahrens und des Netzes wurde anhand verschiedener Experimente getestet.

Danksagung

Für die hervorragende Unterstützung und freundliche Betreuung beim Anfertigen dieser Masterarbeit bedanke ich mich bei Herrn Prof. Dr. Berthold.

Herrn Prof. Dr. Kuhlen danke ich, dass er sich als Zweitgutachter zur Verfügung gestellt hat.

Weiterer Dank gilt Fabian Dill, Thorsten Meinl, Thomas Gabriel, Tobias Koetter und Bernd Wiswedel für die exzellente Betreuung am Lehrstuhl. Ohne ihre Hilfe und Unterstützung hätte ich diese Arbeit nicht anfertigen können.

Für Laura Neuser,
meine Eltern, meine Schwester
und meine Freunde.

道德經

Je mehr du weißt, desto weniger begreifst du.[1]

[1] Laotse, Daodejing

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

Um bezüglich eines Themas umfangreiche Informationen zu finden, reicht es oft nicht aus, nur eine Datenquelle danach zu durchsuchen. Oft werden Informationen aus verschiedenen Datenquellen benötigt, um ein umfassendes Bild zu erhalten. Durch die Verwendung von heterogenen Datenquellen bei der Informationsbeschaffung kommt es oft vor, dass sich die Informationen ergänzen und so zu einem umfangreicheren Suchergebnis beitragen.

Werden z.B. Daten über bestimmte Gene und deren Zusammenhänge mit anderen Genen oder Proteinen gesucht, so ist es zum einen nützlich, diverse wissenschaftliche Artikel über diese Gene zu lesen, zum anderen existieren jedoch noch weitere Datenquellen, die Informationen über Gene enthalten, wie beispielsweise Genontologien oder Genexpressionsdaten. Auch das Wissen verschiedener Personen zu diesem Thema ist als Datenquelle denkbar. Allerdings ist es sehr mühsam, die Informationen dieser unterschiedlichen Datenquellen „manuell“ zusammenzutragen, zu explorieren und zu analysieren.

Ein System, das Informationssuchenden diese Arbeit erleichtert bzw. abnimmt wäre daher von großem Nutzen. Das System muss zum einen die Daten der heterogenen Datenquellen repräsentieren und die Informationen und Beziehungen zwischen diesen analysieren und abbilden. Weiter müssen Anfragen bearbeitet und relevante Ergebnisse zurückgeliefert werden können, die den Informationsbedarf der Benutzer, sofern dies durch den Inhalt der Quellen möglich ist, befriedigen.

1.1 Assoziative Netze

Das menschliche Gehirn bzw. der menschliche Kortex, ist in der Lage, eine sehr große Menge an Informationen aufzunehmen, zu speichern, zu verarbeiten und zu analysieren. Vereinfacht ausgedrückt besteht die Großhirnrinde eines Menschen aus ca. 10^{10} Neuronen ([Hau98]), die zu einem Teil durch Synapsen untereinander verbunden sind. Die

Neuronen und deren Verbindungen fungieren als atomare Informationseinheiten.

Verbindungsorientierte Modelle, wie z.B. künstliche neuronale Netze, sind der Funktionalität des menschlichen Gehirns auf vereinfachte Art und Weise nachempfunden. Da das Gehirn die Aufgabe der Verarbeitung und Analyse von Informationen aus heterogenen Datenquellen gut bewältigt, besteht die Annahme, dass derartige Modelle für eine solche Aufgabe geeignet sind. Prinzipiell bestehen verbindungsorientierte Modelle aus Knoten und Verbindungen zwischen diesen. Im Falle der künstlichen neuronalen Netze werden die Knotenpunkte als Neuronen bezeichnet. Den Verbindungen sind Gewichte zugeordnet, welche angeben, wie ausgeprägt diese sind. Wird das Netz der Knoten aktiviert, um z.B. ein eingegebenes Muster zu erkennen, so werden bestimmte Knoten angeregt. Diese Aktivierung der Knoten verbreitet sich schließlich über die Verbindungen zum benachbarten Knoten, welche ebenfalls angeregt werden. Ausgeprägte oder starke Verbindungen transportieren die Erregung dabei besser als schwache. So wird die Aktivierung folglich durch das Netz verbreitet und dessen Knoten dabei stärker oder weniger stark anregen. Diese Verbreitung der Aktivierung wird auch *spreading activation* genannt. Die Knoten, die am Ende der Verbreitung aktiviert sind und deren Grad der Aktivierung repräsentieren das Ergebnis.

Es gibt verschiedene Ausprägungen verbindungsorientierter Modelle. Neben den bereits erwähnten künstlichen neuronalen Netzen gibt es auch assoziative Netze. Diese speichern bestimmte Informationseinheiten und deren Assoziationen zueinander. Die Information wird hier sowohl in den Knoten als auch in den Verbindungen gespeichert. Künstliche neuronale Netze dagegen speichern die Information meist nur in den Verbindungen. Werden assoziative Netze mit Daten trainiert, so wird zum einen jede Informationseinheit der Datenquelle als Knoten im Netz dargestellt und zum anderen werden die Beziehungen zwischen den Einheiten als Verbindungen zwischen den Knoten repräsentiert. Die Stärke der Verbindung hängt vom Grad der Beziehung ab. Werden heterogene Datenquellen verwendet, um das Netz zu trainieren, so werden die Informationen dieser Datenquellen auch untereinander verbunden, sofern Beziehungen zwischen diesen bestehen. Auf diese Weise können die Informationen aus unterschiedlichen Datenquellen und deren Beziehungen untereinander in einem Netz dargestellt, abgefragt und analysiert werden.

Werden z.B. als Datenquellen wissenschaftliche Publikationen und Gengruppen bzw. Genexpressionsdaten verwendet, so repräsentiert ein Knoten als Informationseinheit einen Term eines Dokuments oder einen Gennamen. Je nach Daten, also Textkorpora und Experimentergebnissen, werden nun verschiedene Terme und Gennamen als Knoten in des Netz eingefügt und miteinander verbunden. Ist ein assoziatives Netz aufgebaut, können z.B. durch *spreading activation*-Verfahren die Informationen abgefragt werden. Dabei werden sowohl Informationen aus Dokumenten als auch aus Experimenten mit Genexpressionsdaten als Ergebnis vorkommen.

1.2 Zielsetzung

Die Zielsetzung dieser Arbeit ist es, zu evaluieren, wie zum einen ein assoziatives Netz aus heterogenen Datenquellen aufgebaut werden kann und diese zum anderen mit Hilfe des Netzes analysiert werden können. Dabei sollen erste Erfahrungen mit der Erstellung und der Verwendung eines solchen Netzes gemacht werden. Als Beispieldatenquellen werden sowohl wissenschaftliche Publikationen aus den Bereichen Biologie und Medizin verwendet als auch Gengruppen bzw. Gennamen aus Genexpressionsdaten. Um gedankliche Ansätze für die Planung des assoziativen Netzes und die Behandlung der Texte zu finden, sollten außerdem aktuelle Techniken aus dem Bereich des Textmining bzw. der Exploration von Texträumen mit Schwerpunkt auf den verbindungsbasierten Modellen betrachtet werden. Weiter sollen die Informationen der Datenquellen im Einzelnen und die, welche sich erst bei einer Kombination dieser ergeben, durch das assoziative Netz repräsentiert werden. Es soll möglich sein, durch bestimmte Anfragen an das Netz an diese Informationen zu gelangen, um diese so einfacher analysieren zu können, ohne die verschiedenen Datenquellen „manuell“ durchsuchen zu müssen. Außerdem soll das assoziative Netz im Zuge verschiedener Experimente auf seine Tauglichkeit getestet werden.

1.3 Aufbau der Arbeit

Zuerst werden in Kapitel 2 die Grundlagen der Exploration von Texträumen erläutert und neben Texten weitere Datenquellen beschrieben, wie Gengruppen und Genontologien. Außerdem wird in Kürze auf das *gene subgroup mining* eingegangen, welches Genexpressionsdaten analysiert und sich ähnlich verhaltende Gene zu Gengruppen zusammengefasst. Kapitel 3 beschreibt den grundsätzlichen Aufbau des verwendeten assoziativen Netzes. Weiter wird in Abschnitt 3.2 die Extraktion von Termen als Informationseinheiten aus den Textkorpora erklärt und in Abschnitt 3.3 wird das Einfügen sowohl von Term- als auch von Genknoten in ein assoziatives Netz dargelegt. Die Erläuterung der Bearbeitung von Anfragen an das Netz durch die Verbreitung der anfänglichen Aktivierung sowie eine kurze Beschreibung des erstellten Server- und Clientprogramms zur Handhabung des fertigen Netzes und zum Erstellen von Anfragen bilden den Schluß dieses Kapitels. Anschließend, in Kapitel 4, werden verschiedene Experimente vorgestellt, in denen unterschiedliche Netze getestet wurden. Den Schluß bildet das Kapitel 5 mit Fazit und Ausblick, in welchem unter anderem Verbesserungsmöglichkeiten des in dieser Arbeit erstellten Netzes beschrieben werden.

Kapitel 2

Grundlagen

Ziel dieses Kapitels ist eine Einführung in die Thematik der Exploration von Wissensräumen. Da es besonders viele Arbeiten in Bezug auf die Exploration von Textträumen gibt, wurde speziell darauf eingegangen. Allerdings können wegen der Fülle der bisher veröffentlichten Methoden und Ansätze nur einige näher erläutert werden. Die verbindungsbasierten Modelle stehen hier im Vordergrund. Ansätze, die z.B. auf genetischen Algorithmen basieren, werden nicht erwähnt. Des weiteren werden neben Texten andere Datenquellen vorgestellt, wie Genexpressionsdaten, Daten aus *gene subgroup mining*-Prozessen oder Genontologien. Auf diese wird jedoch nur in Kürze eingegangen.

2.1 Stand der Technik der Exploration von Textträumen

In den folgenden Abschnitten werden das Vektorraummodell, Clustering von Texten durch SOMs, alternative *spreading activation*-Modelle, wie adaptives Information-Retrieval, assoziative Ansätze mit Hopfield-Netzen und probabilistisches Information-Retrieval beschrieben. Dabei wird jeweils ein Einblick in den Aufbau und die Funktionsweise dieser Modelle gegeben.

2.1.1 Vektorraummodell

Im Vektorraummodell werden Dokumente durch Dokumentvektoren beschrieben. Dabei muss von einem festen Vokabular T ausgegangen werden [Fer03]. Diese Dokumentvektoren bestehen in der Regel aus Gewichten, wobei jedem Term ein Gewicht zugeordnet werden kann.

Bei einer Menge D von Dokumenten $D = \{d_1, ..., d_m\}$ und einem Vokabular, bestehend aus einer Menge von Termen $T = \{t_1, ..., t_n\}$, lässt sich zu jedem Term $t_k \in T$ in jedem Dokument $d_i \in D$ ein Gewicht $w_{i,k} \in \mathbb{R}$ zuordnen, wodurch das Dokument d_i

durch einen Gewichtsvektor bzw. Dokumentvektor $w_i = (w_{i,1}, ...w_{i,n}) \in \mathbb{R}^n$ beschrieben wird. Ebenso werden die Anfragen durch Vektoren $q \in \mathbb{R}^n$ ausgedrückt. Diese Vektoren werden Anfragevektoren genannt. Die Anfrage- und Dokumentvektoren werden durch eine Ähnlichkeitsfunktion $s : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ verglichen, welche jedem Paar aus zwei Vektoren $x, y \in \mathbb{R}^n$ einen Ähnlichkeitswert $s(x, y)$ zuweist. Die ähnlichsten Dokumentvektoren bzw. Dokumente können folglich in der Rangfolge ihrer Ähnlichkeit als Ergebnis auf eine Anfrage zurückgeliefert werden.

Gewichtung

Die Bestimmung der Gewichte eines Dokumentvektors kann durch unterschiedliche Methoden erfolgen. Zum einen können lokale Einflüsse, wie die Häufigkeit eines Terms t_j in einem Dokument d_i (*Termhäufigkeit*, *term frequency*) verwendet werden $w_{i,j} = h(d_i, t_j)$. Zum anderen können die Gewichte auch durch globale Einflüsse berechnet werden, wie z.B. durch die invertierte Dokumenthäufigkeit (*inverted document frequency*) $w_{i,j} = idf(t_j) = \frac{1}{d(t_j)}$, mit $d(t_j)$ als Anzahl der Dokumente aus D, die den Term t_j beinhalten. Hier werden in der Praxis oft modifizierte Funktionen verwendet, wie $w_{i,j} = \ln\left(\frac{m}{d(t_j)}\right)$, mit m als Anzahl der Dokumente. Oft fließen auch lokale und globale Einflüsse kombiniert in die Gewichtsberechnung mit ein, was hier jedoch nicht weiter vertieft wird. Weiteres ist dazu in [Fer03] zu finden.

Im boolschen Retrieval ist die Grundidee Mengenoperationen auf Mengen von Dokumenten anzuwenden. Hier können die Gewichte nur die Werte 0 oder 1 annehmen $w_{i,j} \in \{0, 1\}^n$. Wenn der Term t_j im Dokument d_i vorkommt, so wird $w_{i,j} = 1$ gesetzt, tritt der Term nicht im Dokument auf, wird $w_{i,j} = 0$ gesetzt. Auch die Werte der Anfragevektoren können nur die Werte 0 und 1 annehmen. Komplexe Anfragevektoren werden durch Verknüpfung von elementaren Anfragen durch die boolschen Operatoren *AND*, *OR* und *NOT* gebildet. Die Menge der Ergebnisdokumente zu einer Anfrage ergibt sich durch die Anwendung der zugehörigen Mengenoperationen, $\cap$ für *AND*, $\cup$ für *OR* und für *NOT*, siehe [Fer03].

Zipfsches Gesetz

Das Zipfsche Gesetz beschreibt annähernd die Verteilung der Wörter in einem Korpus. Danach ist die Häufigkeit eines Wortes umgekehrt proportional zu seiner Rangstelle, wenn die Worte nach ihrer Häufigkeit in einer Rangfolge aufgelistet werden. Oder anders gesagt, das Produkt der Häufigkeit und des Häufigkeitsranges sind in etwa konstant.

$$r(w) \cdot h(w) \approx c, \ \forall w \in W(C)$$

Dabei ist $W(C)$ die Menge der Wörter in einem Textkorpus T, $r(w)$ der Rangplatz des Wortes $w \in W(C)$ und $h(w)$ dessen Häufigkeit.

Die Häufigkeit der Terme nimmt nach dem Zipfschen Gesetz also mit

$$h(w) \approx \frac{c}{r(w)}$$

ab. Der Großteil eines Textes wird also durch eine kleine Anzahl von sehr häufigen Wörtern gebildet und nur ein kleiner Teil eines Textes durch eine große Anzahl von seltenen Wörtern. Demzufolge sind häufige Terme keine guten Such- bzw. Indizierungsterme für einen Text, da sie nicht spezifisch genug sind.

Ähnlichkeitsfunktionen

Um die Ähnlichkeit zweier Vektoren, z.B. eines Dokumentvektors $w_i = (w_{i,1}, ...w_{i,n}) \in \mathbb{R}^n$ und eines Anfragevektors $q = (q_1, ...q_n) \in \mathbb{R}^n$ zu bestimmen, gibt es verschiedene Methoden.

Zum einen kann das Skalarprodukt zwischen den beiden Vektoren berechnet werden.

$$w_i \cdot q = \sum_{k=1}^{n} w_{i,k} q_k$$

Beim Skalarprodukt liegen Vektoren, welche die gleiche Ähnlichkeit zu einem Referenzvektor haben auf einer Hyperebene, die orthogonal zu diesem verläuft. Beispielsweise sei der Referenzvektor (a, b) und der Vektor (x, y) gegeben, welche die Ähnlichkeit c haben.

$$ax + by = c$$

So gilt folglich:

$$y = -\frac{a}{b}x + \frac{c}{b}$$

Diese Gerade bildet somit zum Referenzvektor (a, b) mit der Steigung $\frac{b}{a}$ einen rechten Winkel. Parallele Hyperebenen ergeben sich für verschiedene Werte von c.

Zum anderen kann als Ähnlichkeitsmaß das Cosinus-Maß verwendet werden. Beim Cosinus-Maß hat, im Gegensatz zum Skalarprodukt, die Länge der zu vergleichenden Vektoren keinen direkten Einfluß auf die Ähnlichkeit.

$$\cos(w_i, q) = \frac{\sum_{k=1}^{n} w_{i,k} q_k}{\sum_{k=1}^{n} w_{i,k}^2 \sum_{k=1}^{n} q_k^2}$$

Die Ähnlichkeitswerte von Vektoren liegen hier stets im Intervall $[-1, 1]$. Sie hängen nur von der Richtung der Vektoren ab, nicht von deren euklidischer Länge. Wenn zwei Vektoren die gleiche Richtung haben, der Winkel zwischen ihnen also sehr klein bzw. 0

ist, so ist deren Ähnlichkeitswert am größten.

Weitere Ähnlichkeitsfunktionen, wie das Overlap-Maß, das Dice-Maß oder das Jaccard-Maß sind in [Fer03] zu nachzulesen; hier wird darauf allerdings nicht weiter eingegangen.

2.1.2 SOM Clustering

Um Dokumente zu klassifizieren bzw. verschiedenen Gruppen oder Themengebieten zuzuordnen, können auch Clustering-Algorithmen verwendet werden, wie instanzbasierte Lernverfahren, z.B. Nearest-Neighbour Methoden, welche einem Dokument die Kategorie seiner k nächsten Nachbarn zuordnen. Jedoch werden bei diesen Verfahren die Cluster nicht semantisch gekennzeichnet [CHL+97], was es für den Benutzer schwer macht, diese sinnvoll zu durchsuchen, um für ihn interessante Gruppen zu finden.

Kohonens selbst-organisierende Karten (SOM) ([Koh89], [Koh95]) bieten hier eine gute Alternative als unüberwachte Clustering-Verfahren. Wie auch im Vektorraummodell werden die Dokumente durch n-dimensionale Dokumentvektoren $w_i = (w_{i,1}, ...w_{i,n}) \in \mathbb{R}^n$ beschrieben, welche auf Neuronen bzw. Knoten abgebildet werden, die in einem zwei-dimensionalen Gitter angeordnet sind. Jedes dieser Neuronen ist durch gewichtete Verbindungen mit n Eingabeneuronen verbunden. Die Dokumentvektoren im n-dimensionalen Raum werden während des Lernprozesses in einen zwei-dimensionalen Raum abgebildet, wobei die Nachbarschaftsinformationen nicht verloren gehen. Dokumente, die derselben Gruppe angehören, werden durch Knoten repräsentiert, die nahe zusammen liegen, während Dokumente aus komplett unterschiedlichen Gruppen weiter auseinander liegen werden. Die Größe einer Gruppe wird ebenfalls berücksichtigt. Für Gruppen mit einer großen Anzahl an Dokumenten wird auf der SOM mehr Platz zur Verfügung gestellt.

Eine SOM wird anfangs mit zufälligen Gewichten initialisiert und durchläuft dann folgenden iterativen Lernprozess [Koh89]:

1. Ein Inputvektor $w_i = (w_{i,1}, ...w_{i,n}) \in \mathbb{R}^n$ wird zufällig ausgewählt

2. Das Gewinnerneuron $n_j = (n_{j,1}, ...n_{j,n}) \in \mathbb{R}^n$, dessen Gewichte den kleinsten Abstand zum Inputvektor haben, wird ermittelt. Als Abstandsmaß wird oft die Euklidische Distanz verwendet $d(w_i, n_j) = \sqrt{\sum_{k=1}^{n} (w_{i,k} - n_{j,k})^2}$.

3. Die Gewichte des Gewinnerneurons werden angepasst, indem sie weiter in Richtung der Werte des Inputvektors bewegt werden $n_j(\tau + 1) = n_j(\tau) + \eta * (w_i - n_j(\tau))$, mit η als Lernrate.

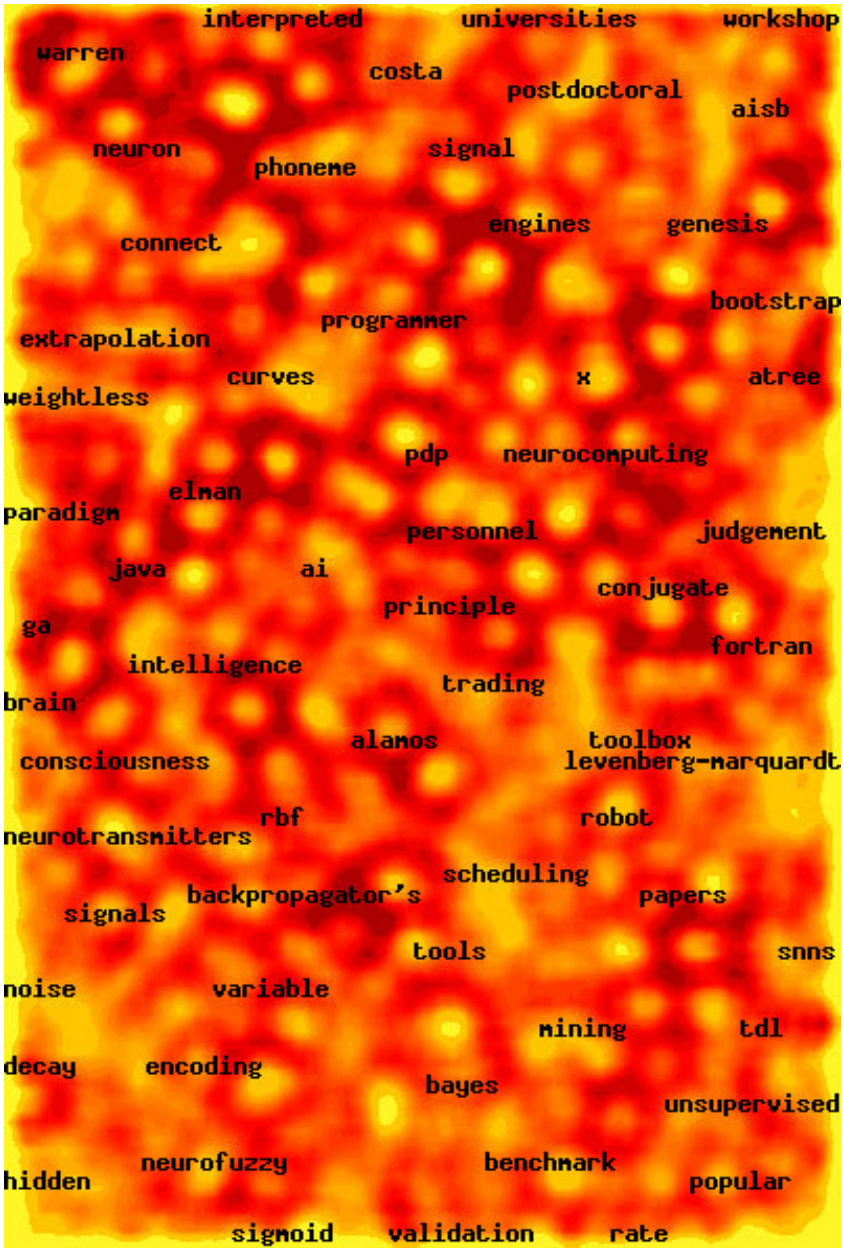


Abbildung 2.1: WEBSOM map - comp.ai.neural-nets (aus http://websom.hut.fi/websom/comp.ai.neural-nets-new/html/root.html), erzeugt aus 12088 Artikeln mit automatisch generierten Bezeichnungen.

4. Die Gewichte der Nachbarneuronen des Gewinnerneurons werden ebenfalls in Richtung des Inputvektors angeglichen.

Bei der Anpassung der Gewichte wird eine Lernrate η verwendet, welche sich im Laufe der Iterationen verringert. Wenn die Lernrate, also auch die Änderungen, sehr klein werden und gegen 0 gehen, wird der Lernvorgang abgebrochen. Neue Dokumente

können nun dem Neuron zugeordnet werden, welches den geringsten Abstand zu diesen hat. Der Cluster, in dem sich das Neuron befindet, wird schließlich diesen Dokumenten zugeordnet. Kommen viele neue Dokumente hinzu, so kann der Lernvorgang fortgeführt werden, um die Karte an neue Gruppen bzw. Clustern anzupassen [CHL+97].

Auf der fertigen SOM haben sich die Trainingsdokumente zu Clustern verschiedener Dichte und Größe gruppiert, je nach Beschaffenheit der Trainingsdaten. Abbildung 2.1 zeigt eine mit 12088 Artikeln über maschinelles Lernen und künstliche Intelligenz trainierte Karte.

Eine SOM ist weniger ein Suchinstrument, um Dokumente mit bestimmten Termen zu finden, sondern mehr eine Unterstützung für den Benutzer beim Durchsuchen von Dokumenten bestimmter Themengebiete. Die Anfrage eines Benutzers wird auf eine oder mehrere Regionen bzw. Neuronen der Karte abgebildet und die Dokumente, die diesen Neuronen zugeordnet sind, werden als Ergebnis präsentiert.

2.1.3 Adaptives Information-Retrieval

„Connectionist“, also verbindungsbasierte Modelle, wurden ebenfalls in verschiedenen Ausführungen bezüglich Information-Retrieval-Anwendungen erprobt. Eines dieser Modelle namens AIR (Adaptive Information Retrieval) geht auf Richard K. Belew zurück ([Bel86], [Bel89], [Bel00]). Hier werden Dokumente und ihre Attribute, wie Terme, Autoren oder Verlage als Knoten dargestellt, die untereinander verbunden sind. Eine Anfrage verursacht eine anfängliche Aktivität bei verschiedenen Knoten, welche dann durch das Netz propagiert, bis schließlich bestimmte Abbruchbedingungen erreicht sind. Die Knoten mit der größten Aktivität werden als Ergebnis zurückgeliefert, welche dann von den Benutzern bewertet werden. Durch diese Bewertungen (*Relevance Feedback*) wird das Netz trainiert. Fallen die Bewertungen positiv aus, so werden die Gewichte der Verbindungen zwischen den Knoten erhöht, fallen sie negativ aus, so werden die Gewichte verringert.

Wie bei den meisten verbindungsbasierten Modellen, liegt auch bei AIR ein gewichteter Graph als Datenstruktur zugrunde. Dieser Graph wird anfänglich als ein Netzwerk aus Dokumenten und deren Autoren und Termen aufgebaut. Diese initialen Verbindungen sind nötig, um mit AIR als Information-Retrieval System im Initialisierungszustand sinnvoll arbeiten zu können.

Initialisierung des Netzwerkes

Jedes Zitat in einem Dokument bewirkt, dass ein weiterer Dokumentknoten mit dem zitierten Dokument gebildet wird. Weiter werden für jeden Autor des neuen Dokuments

Autorenknoten gebildet und für jeden Term im Titel werden Termknoten gebildet, nachdem Stopwörter entfernt und Pluralformen in Sigularformen umgewandelt wurden. Die Autoren- und Termknoten werden dann mit dem neuen Dokumentknoten gewichtet verbunden. Die Gewichte werden durch die inverse Häufigkeit (*inverse frequency*) bestimmt, wobei die Summe aller gewichteten Verbindungen, die einen Knoten verlassen, eine Konstante a sein muss, nach [Bel89] $a = 1$. In [Bel89] bildeten in Experimenten 1600 Dokumente etwa 5000 Knoten. Die Bedingung, dass die Summe aller ausgehenden Verbindungen eine Konstante sein muss, hat den Vorteil der *Aktivitätserhaltung*. Dies bedeutet, dass die ausgehende Aktivität eines Knotens immer a ist und sich der Betrag der Aktivität somit niemals erhöht oder verringert, was sehr nützlich ist, um die Ausbreitung der Aktivität im Netzwerk zu kontrollieren.

Anfragen an das Netzwerk

Die Benutzer beschreiben durch eine einfache Anfragesprache ihren Informationsbedarf. Es ist möglich, einen oder mehrere Anfrageteile zu einer ganzen Anfrage zusammen zu stellen. Jeder Anfrageteil kann aus einem Attribut, also einem Term, einem Autor oder aus einem Dokument bestehen. Alle bis auf den ersten Anfrageteil können verneint werden. Eine solche Anfrage erzeugt bei den Knoten, die mit den Anfrageteilen übereinstimmen, eine Aktivität, welche dann durch das Netzwerk propagiert. Die Knoten mit der höchsten Aktivität werden als Ergebnis zurückgeliefert, in der Annahme, dass diese am relevantesten in Bezug auf die Anfrage sind.

Relevanzbewertung

Nachdem ein Resultat auf eine Anfrage vorliegt, bewertet der Benutzer, welche Knoten seiner Ansicht nach relevant sind und welche nicht. Hierfür liegen vier Abstufungen vor: $++$, $+$, $-$ und $--$ für *sehr relevant*, *relevant*, *irrelevant* und *sehr irrelevant*. Daraufhin erzeugt das System eine neue Anfrage, basierend auf der Bewertung des Benutzers, in der zuerst die Anfrageteile der alten Anfrage übernommen werden und außerdem die als positiv markierten Knoten aus dem Resultat. Die als negativ bewerteten Knoten werden verneint in die Anfrage aufgenommen. Dadurch durchsucht der Benutzer sozusagen das Netzwerk nach für ihn relevanten Ergebnissen, wobei er die Richtung der als irrelevant markierten Knoten vermeidet und die der als relevant markierten Knoten bevorzugt.

Training des Netzwerkes

Das Training eines AIR Netzwerkes unterscheidet sich vom Training traditioneller verbindungsbasierter Modelle, wie z.B. Hopfield-Netzen [TH87] dadurch, dass es keinen anfänglichen, einheitlichen Lernalgorithmus gibt. Die Veränderung der Gewichte, also der Lernprozess wird durch den Benutzer gesteuert, der ein Resultat bewertet. Knoten, die als relevant bzw. irrelevant bewertet wurden, verbreiten ein Signal, welches nun

rückwärts durch das Netz entlang der gewichteten Verbindungen läuft. Die Gewichte der Verbindungen, die direkt oder indirekt in den Anfragevorgang miteinbezogen wurden, werden dann durch eine lokale Lernregel modifiziert. In [Bel89] wurde eine Lernregel verwendet, welche die Aktivität des „pre-synaptischen“ Knoten n_i mit dem Feedbacksignal des „post-synaptischen“ Knoten n_j in Beziehung setzt:

$$w_{ij} \propto Corr\left(n_i \; active, n_j \; relevant\right)$$

Der Aktivitätsgrad der Knoten am Ende der Propagierungsphase wird als Prognose der Wahrscheinlichkeit, dass dieser Knoten als relevant in Bezug auf die Anfrage bewertet wird, erachtet. Ein Gewicht w_{AB} zwischen zwei Knoten n_B und n_A ist also die bedingte Wahrscheinlichkeit, dass Knoten n_B relevant ist, wenn Knoten n_A als relevant gilt. Die Interaktionen mit dem System seitens der Benutzer werden als Experimente betrachtet. Bei einer Anfrage prognostiziert AIR, welche Knoten relevant sind und der Benutzer bestätigt oder verneint diese Annahme.

Werden die Bewertungen von nur wenigen Benutzern durchgeführt, so wird das System die Meinungen über die Relevanz bezüglich der Anfragen von diesen Benutzern adaptieren. Es muss also darauf geachtet werden, dass viele unvoreingenommene Benutzer dem System Bewertungen liefern, um die Meinungen vieler in das System einfließen zu lassen. Weiter kann es sein, dass sich Resultate auf Anfragen während der Laufzeit des Systems ändern, da sich die Gewichte der Verbindungen der Knoten an verschiedenen Bewertungen anpassen. Dies kann als Nachteil des Trainings durch relevance feedback gesehen werden.

2.1.4 Hopfield-Netze

Weitere verbindungsbasierte Modelle liegen Hopfield-Netzen [TH87] zugrunde ([Che95], [CBN95], [CPS98]). Hier werden allerdings nur Terme miteinander assoziiert, Autoren bzw. andere Attribute der Dokumente werden, anders als im AIR System nicht berücksichtigt, was generell jedoch auch möglich wäre. Hopfield-Netze können zur automatischen Thesauruserstellung verwendet werden. Dabei werden die aus Dokumenten extrahierten Terme als Netzknoten untereinander durch gewichtete Verbindungen vernetzt. Je nach dem, in welcher Relation die Wörter zueinander stehen sind die Gewichte größer oder kleiner. Verbindungen zwischen Termen, die oft in Kombination mit anderen Termen in Dokumenten auftreten werden größere Gewichte haben, als Verbindungen zwischen Termen, die so gut wie nie zusammen in Dokumenten auftauchen. Abbildung 2.2 zeigt ein Hopfield-Netz aus den drei Termen „Textmining“, „Term“ und „Korpus“. Im Falle einer Anfrage werden die Terme bzw. Knoten des Netzes, die in der Anfrage existieren, angeregt und die Aktivität verbreitet sich schließlich durch das Netz, bis es einen stabilen Zustand einnimmt. Die Knoten mit der höchsten Aktivität werden als Ergebnis zurückgeliefert. Die Terme dieser Ergebnisknoten werden also mit den Termen der Anfrage durch das Netz assoziiert.

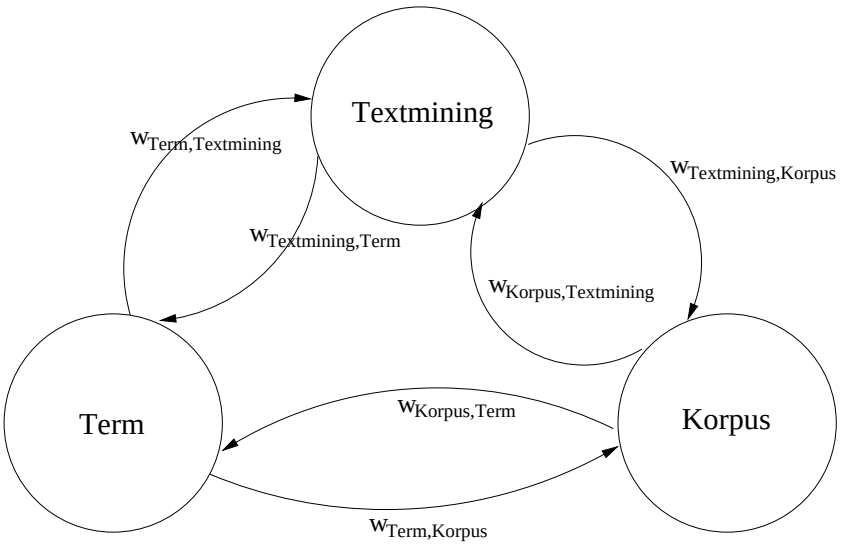


Abbildung 2.2: Ein Hopfield-Netz aus den drei Termen „Textmining“, „Term“ und „Korpus“.

Erstellung des Netzes

In einem Hopfield-Netz mit m Knoten sind alle Knoten durch gewichtete Verbindungen $w_{ij},\ i,j \in \{1,...,m\}$ miteinander verbunden. Zu sich selbst haben die Knoten jedoch keine Verbindung bzw. ist deren Gewicht 0 $w_{ii} = 0$. Ein solches Netz wird durch eine $m \times m$ Gewichtsmatrix dargestellt, in welcher die Gewichte im allgemeinen symmetrisch sind $w_{ij} = w_{ji}$, ([TH87], [Sch97]), die gewichteten Verbindungen demnach ungerichtet. In [Che95], [CBN95] und [CPS98] werden jedoch gerichtete Verbindungen verwendet, was bedeutet, dass die Gewichte nicht symmetrisch sind. Weiter ist jedes Gewicht eine reelle Zahl zwischen 0 und 1, welche folgendermaßen bestimmt wird:

$$w_{ij} = \frac{\sum_{k=1}^{m} e_{kij}}{\sum_{k=1}^{m} e_{ki}}$$

$$w_{ji} = \frac{\sum_{k=1}^{m} e_{kij}}{\sum_{k=1}^{m} e_{kj}}$$

w_{ij} ist das Gewicht der Verbindung von Term i zu Term j, wobei e_{ki} anzeigt ob der Term i im Dokument k vorkommt oder nicht. e_{ki} wird 1 gesetzt, falls der Term auftritt, ansonsten 0. Analog dazu gilt $e_{kij} = 1$, falls sowohl Term i, als auch Term j in Dokument k vorkommt bzw. $e_{kij} = 1$, falls nicht.

Anfragen an das Netzwerk

Eine Benutzeranfrage besteht aus einer Menge an Termen $\{t_1, t_2, ..., t_k\}$. Jeder Knoten des Netzwerks, der mit einem dieser Terme übereinstimmt wird mit einem Gewicht von 1 aktiviert.

$$\mu_i(0) = x_i,\ 1 \leq i \leq m$$

Der Output des Knotens i zum Zeitpunkt τ ist μ_i und der Input des Knotens i ist x_i, was zwischen 0 und 1 liegt. Zum Zeitpunkt 0 beträgt der Input für alle Knoten, die mit Termen aus der Anfrage übereinstimmen, 1. m sei hier die Anzahl der Knoten bzw. der Terme im Netzwerk.

Jeder Zustand des Netzes in einer Iteration liegt dem Zustand der vorhergehenden Iteration zugrunde,

$$\mu_j(\tau + 1) = f_s\left[\sum_{i=1}^{m} w_{ij}\mu_i(\tau)\right],\ 1 \leq j \leq m$$

wobei f_s eine kontinuierliche Sigmoid Funktion ([Kni90], [DD]) ist.

$$f_s(net_j) = \frac{1}{1 + exp\left[\frac{-(net_j - \theta_j)}{\theta_0}\right]}$$

θ_j ist ein Bias bzw. Grenzwert, weiter gilt $net_j = \sum_{i=1}^{m} w_{ij}\mu_i(\tau)$. θ_0 dient dazu, die Form der Sigmoid Funktion zu modifizieren. In jeder Iteration werden aufgrund der Eigenschaft der parallelen Relaxation alle Knoten zur gleichen Zeit aktiviert. Basierend auf der parallelen Aktivierung wird für jeden neuen angeregten Knoten dessen Input durch die Summe der Produkte der Gewichte zu seinen Nachbarknoten und deren Outputs berechnet.

Konvergenz des Hopfield-Netzes

Dieser iterative Prozess wird solange wiederholt, bis das Netz einen stabilen Zustand erreicht und somit keine starke Veränderung der Outputwerte der Knoten von Iteration zu Iteration erkennbar ist. Dies wird durch die folgende Formel aus [Che95] überprüft.

$$\sum_{j=1}^{m} |\mu_j(\tau + 1) - \mu_j(\tau)| \leq \epsilon$$

ϵ ist die maximale Differenz der Outputs zwischen zwei Netzzuständen, was in der Regel eine kleine Zahl ist. Das endgültige Resultat stellt die Terme dar, welche am relevantesten in Bezug auf die Anfrageterme sind.

2.1.5 Probabilistische Modelle

Deduktive Netzwerke [TC90] bewerten Dokumente nach der Wahrscheinlichkeit, dass sie den Informationsbedarf eine Benutzers befriedigen [CHL+97]. Die Struktur solcher Systeme besteht aus vier Schichten. Die oberste Schicht enthält Knoten, die Dokumente repräsentieren. Diese Knoten sind mit Knoten aus der zweiten Schicht verbunden, welche die Terme der Dokumente repräsentieren. Die Dokumentknoten sind jedoch nur mit Termknoten verbunden, wenn die Terme im entsprechenden Dokument auftreten. Die Verbindung kann mit der Häufigkeit gewichtet sein, mit der ein Term in einem Dokument auftritt. Dokumente, welche dieselben Terme enthalten, sind auch mit denselben Termknoten verbunden. In der dritten Schicht befinden sich Knoten, welche die Anfrageterme des Benutzers repräsentieren und in der vierten befinden sich Knoten, die den Informationsbedarf darstellen. Die Knoten aus der dritten und vierten Schicht werden für jede Anfrage neu erstellt, sind also nicht wie die aus der ersten und zweiten Schicht von Dauer.

Wird nun eine Anfrage gestellt, wird das Netz von der ersten Schicht bis zur vierten Schicht durchlaufen, wobei Wahrscheinlichkeiten berechnet werden, welche besagen wie relevant ein Dokument bezüglich eines Informationsbedarfs ist. Diese Wahrscheinlichkeiten basieren meist auf Bayes[1] bzw. Dempster-Shafer Modellen[2] [CHL+97]. Die Dokumente, die somit als Ergebnis einer Anfrage ermittelt werden, werden in der Rangfolge ihrer Relevanz ausgegeben.

Probabilistisches Indizieren

Das Verhältnis der bedingten Wahrscheinlichkeiten, dass bei gegebener Relevanz $(+R)$ bzw. Irrelevanz $(-R)$ bezüglich einer Anfrage ein Dokument d_i gefunden wird, kann folgendermaßen beschrieben werden.

$$P\left(d_i|+R\right)/P\left(d_i|-R\right)$$

Diese Theorie, basierend auf dem Bayeschen Theorem, wurde erstmals in [RJ88] und [vR77] vorgestellt und setzt zwei Annahmen voraus:

1. Die Indexterme eines Dokumentes bzw. einer Anfrage sind unabhängig.

2. Die Dokumentvektoren enthalten nur binäre Werte, beschreiben also nur, ob ein Term in einem Dokument auftritt, nicht aber wie oft.

[1] Bayessche Netze stellen eine spezielle Form der Formulierung von wahrscheinlichkeitstheoretischen Modellen dar. Durch sie lassen sich unsicheres Wissen und die daraus möglichen Schlussfolgerungen abbilden [Jen01].

[2] Durch die Evidenztheorie von Dempster und Shafer können Informationen unterschiedlicher Quellen unter Berücksichtigung der Glaubwürdigkeit dieser Quellen zu einer Gesamtaussage zusammengesetzt werden [Sha76].

Den Anfragetermen werden jedoch Gewichte zugeordnet, welche aus dem oben genannten Verhältnis hervorgehen, wenn diese als unabhängig angesehen werden. Um diese Wahrscheinlichkeiten zu bestimmen, werden anfangs relevante Beispieldokumente benötigt, welche z.B. durch eine Relevanzbeurteilung durch den Benutzer bestimmt werden können. Als irrelevante Beispieldokumente werden alle restlichen Dokumente verwendet. Das Gewicht eines Terms t_k einer Anfrage q_a wird wie folgt berechnet:

$$g_{ak} = g^r_{ak} + g^s_{ak}$$

$$g_{ak} = \log\left[r_{ak}/\left(1 - r_{ak}\right)\right] + \log\left[\left(1 - s_{ak}\right)/s_{ak}\right]$$

Hier gilt $r_{ak} = P\left(t_k\ present| + R\right)$ und $s_{ak} = P\left(t_k\ present| - R\right)$. Dies sind die bedingten Wahrscheinlichkeiten, dass bei gegebener Relevanz bzw. Irrelevanz bezüglich einer Anfrage q_a der Term t_k in den Dokumenten gefunden wird. Für ein Dokument d_i ist das optimale Gewicht, um dessen Rang festzustellen:

$$W_i = \sum_k w_{ik} g_{ak}$$

Das Dokument d_i besteht aus einem Dokumentvektor der Form $w_i = (w_{i1}, ..., w_{ik}, ...)$, für welchen bei $w_{ik} = 1$ gilt, dass Term t_k in d_i auftritt bzw. bei $w_{ik} = 0$ gilt, dass der Term t_k in d_i nicht auftritt. Dabei geht die Summe über alle Terme, die sowohl in d_i als auch in q_a vorkommen.

Das probabilistische Modell nach [RJ88] und [vR77] hat allerdings zwei entscheidende Nachteile: zum einen enthalten die Dokumentvektoren nur binäre Werte, was zur Folge hat, dass die Information bezüglich der Termhäufigkeit eines Dokuments verloren geht. Zum anderen werden, um die anfängliche Gewichtung der Anfrageterme zu bestimmen, relevante Beispieldokumente benötigt, die vorher durch Benutzer bereitgestellt werden müssen. Diese und einige andere Nachteile werden durch den Ansatz in [Kwo85], [Kwo86] und [KK88] abgeschwächt. Dabei wird ein Dokument durch verschiedene Komponenten dargestellt. Weiter wird nicht mehr mit einer Menge von Dokumenten gearbeitet, sondern mit einem Raum von Dokumentkomponenten. Diese Komponenten können Phrasen oder auch Terme sein und sind unabhängig und eindeutig.

Muss nun überprüft werden, ob ein Dokument d_i bezüglich einer Anfrage q_a relevant ist, werden wie bereits oben beschrieben, die Gewichte der Anfrageterme t_k berechnet, jedoch mit:

$$r_{ak} = q_{ak}/L_a,\ \ s_{ak} = F_k/N_W$$

Hier ist nun r_{ak} die Termhäufigkeit q_{ak} des Terms t_k innerhalb der Anfrage q_a, dividiert durch die Länge L_a der Anfrage. s_{ak} ist die Häufigkeit F_k des Terms t_k innerhalb der Komponentenkollektion, dividiert durch die Größe der Kollektion N_W. Um die Werte r_{ak} und s_{ak} zu berechnen, muss also keine Beispielmenge an relevanten Dokumenten mehr erhoben werden. Vielmehr liegt der Fokus bei der Berechnung von r_{ak} auf der Anfrage,

da deren Länge und die Häufigkeit der Terme benötigt werden. Dadurch wird nun das optimale Gewicht für das Dokument d_i wie folgt berechnet:

$$WQ_i = \sum_k (d_{ik}/L_k)\, g_{ak}$$

Wobei die Summe wiederum über alle Terme geht, die sowohl in d_i als auch in q_a auftauchen. Das Q in WQ_i soll daran erinnern, dass der Fokus der Berechnung des Gewichtes auf der Anfrage liegt. Es können jedoch nicht nur die Anfrageterme, sondern analog dazu auch die Terme eines Dokuments d_i gewichtet werden. Somit wird bestimmt, ob eine Anfrage q_a relevant, bezogen auf d_i ist oder nicht. Das Gewicht eines Terms t_k wird mit Fokus auf dem Dokument d_i folgendermaßen berechnet:

$$g_{ik} = g^r_{ik} + g^s_{ik}$$

$$g_{ik} = \log\left[r_{ik}/\left(1 - r_{ik}\right)\right] + \log\left[\left(1 - s_{ik}\right)/s_{ik}\right]$$

mit

$$r_{ik} = d_{ik}/L_i$$

$$s_{ik} = \left(F_k - d_{ik}\right)/\left(N_W - L_i\right)$$

r_{ik} und s_{ik} haben dieselbe Bedeutung wie r_{ak} und s_{ak}, jedoch bezogen auf das Dokument d_i. d_{ik} ist hier die Termhäufigkeit des Terms t_k in d_i und L_i die Länge von d_i. Das Gewicht von d_i ist somit:

$$WD_i = \sum_k (q_{ak}/L_a)\, g_{ik}$$

Auch hier wird wieder über alle Terme summiert, die sowohl in d_i als auch in q_a auftauchen. Werden die Formeln zur Bestimmung von WQ_i und WD_i zusammengefasst, so ergibt sich nach [Kwo89] folgende Methode zu Berechnung des Gewichtes:

$$W_i = \sum_k (q_{ak}/L_a)\, g_{ik} + (d_{ik}/L_i)\, g_{ak}$$

2.1.6 Probabilistisches Information-Retrieval mit neuronalen Netzen

In [Kwo89] wird ein neuronales Netz für probabilistisches Information-Retrieval vorgestellt, welches drei Ebenen hat. Eine Ebene für Anfragen, eine für Terme und eine für Dokumente. In der jeweiligen Ebene repräsentiert stets ein Neuron eine Anfrage, einen Term oder ein Dokument, was in Abbildung 2.3 zu sehen ist. Die Verbindungen zwischen den Ebenen sind bidirektional und asymmetrisch, außerdem werden Anfragen und Dokumente als Neuronen derselben Kategorie angesehen und können sowohl als Input- oder Outputneuronen agieren. Verbindungen von Neuronen innerhalb einer Ebene existieren nicht und als Output- bzw. Aktivierungsfunktion wird die Identitätsfunktion verwendet.

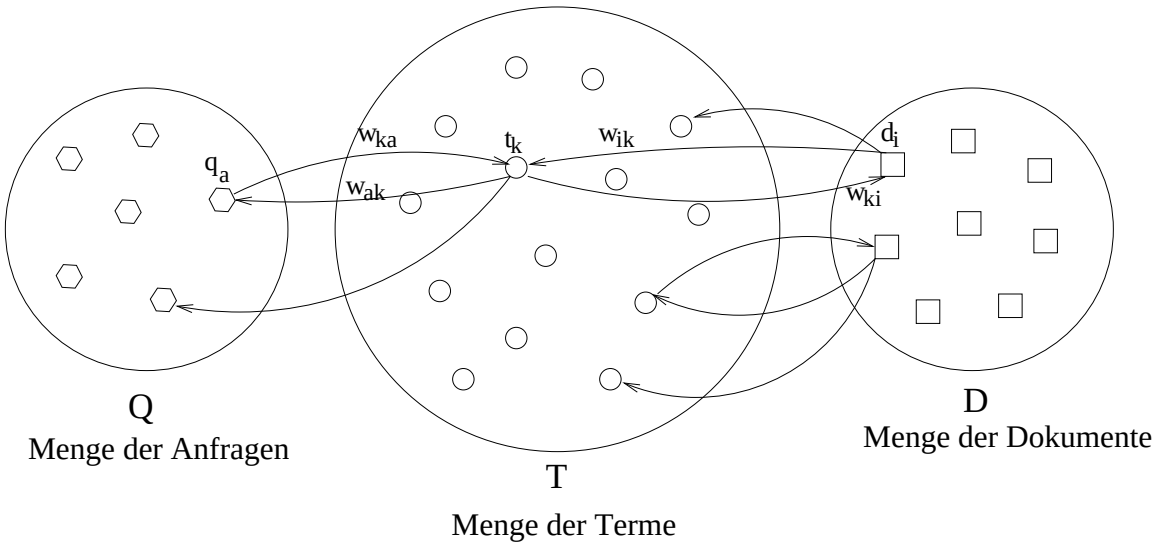


Abbildung 2.3: Ein dreischichtiges neuronales Netz für probabilistisches Information-Retrieval. (Es werden nicht alle Verbindungen angezeigt.)

Initialisierung der Gewichte

Das Gewicht der Verbindung eines Neurons der Anfrageschicht n_a bzw. der Dokumentschicht n_i zu einem Neuron n_k der Termschicht wird mit $w_{ka} = q_{ak}/L_a$ bzw. $w_{ki} = d_{ik}/L_i$ initialisiert. Durch diese initialen Gewichte kommt zum Ausdruck, dass bei einer gegebenen Anfrage q_a bzw. einem gegebenen Dokument d_i die Wahrscheinlichkeit q_{ak}/L_a bzw. d_{ik}/L_i besteht, dass ein einzelner Term t_k verwendet wird. Das Verbindungsgewicht w_{ak} bzw. w_{ik} von einem Neuron n_k der Termschicht zu einem Neuron der Anfrageschicht n_a bzw. der Dokumentschicht n_i setzt sich aus zwei Teilen zusammen $w_{ak} = w^r_{ak} + w^s_{ak}$ bzw. $w_{ik} = w^r_{ik} + w^s_{ik}$, wobei gilt $w^s_{ak} = w^s_{ik} = \log(1 - S_k)/S_k$, mit $S_k = F_k/N_W$. w^r_{ak} und w^r_{ik} werden kleine Werte, wie $\log[p/(1-p)]$ zugeordnet, mit p als kleiner positiver Konstante. Die Bestimmung der Gewichte w_{ak} bzw. w_{ik} für einzelne Terme als Dokumentkomponenten geschieht also nach dem Schema der inversen Dokumenthäufigkeit. Die Gewichte bieten somit die vollständige Information bezüglich des Nutzens eines Terms t_k, in Verbindung mit einer Anfrage q_a bzw. einem Dokument d_i, gemessen an dessen Häufigkeit [Kwo89]. Weitere inhaltliche Informationen beinhalten die Gewichte jedoch nicht.

Neben der Methode, die Gewichte wie oben aufgeführt zu initialisieren, wird in [Kwo89] außerdem ein Lernverfahren vorgestellt, welches die Gewichte durch einen iterativen Prozess bestimmt. Dieses Verfahren kann als eine Art Hebbsches Lernverfahren [Heb49] angesehen werden und ist dem aus [Bel86] sehr ähnlich. Dies wird hier allerdings nicht weiter erläutert.

Verwendung des Netzes

Ist das Netz initialisiert, kann es durch eine anfängliche Aktivierung bestimmter Neuronen aus einer der äußeren Schichten genutzt werden. Diese Aktivierung breitet sich dann, je nach gewünschtem Ergebnis, von der ersten Schicht bis zur letzten Schicht (*feed-forward*) bzw. von der letzten bis zur ersten (*feed-backward*) aus, bis ein Resultat in Form von Dokumenten oder Anfragen gefunden ist. Um für eine Anfrage q_a relevante Dokumente zu finden, kann das Netz sowohl von hinten nach vorne durchlaufen werden als auch umgekehrt. Der erste Fall kann als analog zu der Formel zur Berechnung von WQ_i gesehen werden. Der Fokus liegt hier auf der Anfrage und alle Neuronen in der Dokumentschicht werden mit dem Input 1 aktiviert. Die Aktivität breitet sich nun über die Neuronen der Termschicht zu den Neuronen der Anfrageschicht aus, sofern die Verbindungsgewichte nicht 0 sind. Jedes Dokument wird somit, basierend darauf, ob die Aktivität das Anfrageneuron n_a erreicht oder nicht, auf dessen Relevanz geprüft. Der Wert der Aktivität, der ausgehend von einem Dokumentneuron am Anfrageneuron ankommt, wird verwendet, um die Rangfolge der Ergebnisdokumente festzustellen. Wird das Netz *feed-forward* verwendet, so wird das Neuron n_a der Anfrageschicht mit einem Wert von 1 aktiviert, worauf sich die Aktivität bis zu den Neuronen der Dokumentschicht ausbreitet. Wiederum wird die Aktivität, welche die Dokumentneuronen erreicht, verwendet, um die Rangfolge der Ergebnisdokumente festzustellen. Dieser Fall kann als analog zur Formel zur Berechnung von WD_i angesehen werden, da hier der Fokus auf den Dokumenten liegt.

2.2 Andere Datenquellen

Dieser Abschnitt beschreibt neben Texten, bzw. Methoden der Exploration von Textträumen, weitere Datenquellen wie Genexpressionsdaten, Genontologiedaten und ein Verfahren zur Analyse und Gewinnung von Daten, wie *gene subgroup mining*. Diese Daten werden zusätzlich zu den Termen aus Dokumenten in ein assoziatives Netz eingebunden, welches in Abschnitt 3.3.2 beschrieben ist. Generell sind weitere Datenquellen denkbar, deren Daten in ein assoziatives Netz integriert werden können, wie das Wissen einzelner Personen etc. Innerhalb dieser Arbeit wird darauf allerdings nicht weiter eingegangen.

2.2.1 Gene subgroup mining

Im Folgenden wird das *gene subgroup mining* kurz erläutert. Details werden hier jedoch nicht erläutert, da dies den Rahmen dieser Arbeit sprengen würde. Das *gene subgroup mining* analysiert Genexpressionsdaten und basiert auf dem Konzept des *association rule mining*, was selbst wiederum auf der *Warenkorbanalyse* basiert. Durch die Warenkorbanalyse wird festgestellt, welche Artikel innerhalb einer Transaktion überdurchschnittlich oft zusammen gekauft werden bzw. welcher Artikel am wahrschein-

lichsten gekauft wird, wenn eine bestimmte Kombination anderer Produkte vorliegt. So wurde in den USA herausgefunden, dass Windeln und Bier oft gemeinsam eingekauft werden.

Generell geht es darum, Verbindungen und Abhängigkeiten zwischen Objekten zu finden. In der Warenkorbanalyse sind diese Objekte Waren, im *gene subgroup mining* sind es Gene. Dabei wird nach Assoziationsregeln gesucht, welche z.B. besagen, dass ein Objekt x zu einem bestimmten Prozentsatz auftritt, wenn auch ein anderes Objekt y auftritt. Die Warenkorbanalyse untersucht, welche Objekte zusammen gekauft werden, während durch das *gene subgroup mining* herausgefunden werden kann, welche Gene sich in Abhängigkeit voneinander verändern bzw. *overexpressed* oder *underexpressed* sind. Gene werden bezüglich eines Experiments als *overexpressed* bezeichnet, wenn deren RNA-Abschnitt in der zu untersuchenden Zelle häufiger auftritt als in Vergleichsexperimenten. Analog dazu werden Gene in Bezug auf ein Experiment als *underexpressed* bezeichnet, wenn deren RNA-Abschnitt in einer Zelle im Vergleich zu anderen Experimenten weniger häufig auftritt.

Die bekanntesten Algorithmen, um Assoziationsregeln zu finden, sind der *Apriori*-Algorithmus [AIS93] und der *Eclat*-Algorithmus [ZPOL97]. Die Struktur von Genexpressionsdaten ist für diese Algorithmen, die Transaktionen und deren Objekte analysieren, jedoch ungeeignet und muss erst an diese angepasst werden. Wie schon erwähnt, sind die Gene, die zu untersuchenden Objekte und die Transaktionen sind in diesem Fall die Genexpressionsexperimente. Weiter wird nach Genen gesucht, welche in Abhängigkeit voneinander in genügend Experimenten als *overexpressed* oder *underexpressed* erkannt wurden bzw. bei welchen sich in Abhängigkeit voneinander das *Expressionsniveau* verändert hat. Eine ausführliche Beschreibung zu *gene subgroup mining* und Genexpressionsdaten ist in [Dil06] zu finden.

2.2.2 Genontologien

Genontologien bestehen aus einem kontrolliertem Vokabular, welches die Hierarchie von Genfunktionen und die Biologie der genetischen Prozesse und Erzeugnisse beschreibt. Das kontrollierte Vokabular besteht aus drei unabhängigen Gruppen. Diese Gruppen beschreiben die molekularen Funktionen genetischer Erzeugnisse, die biologischen Prozesse, an denen diese Erzeugnisse teilhaben und die zellularen Bestandteile, in welchen diese Erzeugnisse gefunden werden können.

Die Ontologien werden durch einen gerichteten, azyklischen Graphen repräsentiert, in welchem Knoten mehrere Vorgängerknoten und verschiedene Beziehungen zu diesen haben können. Eine Beziehung eines Knotens zu einem anderen wird durch die Kante zwischen diesen ausgedrückt. Zusätzlich erbt ein Knoten alle Beziehungen der Vorgängerknoten. Z.B. hat der biologische Prozess „pheromone processing“ zwei

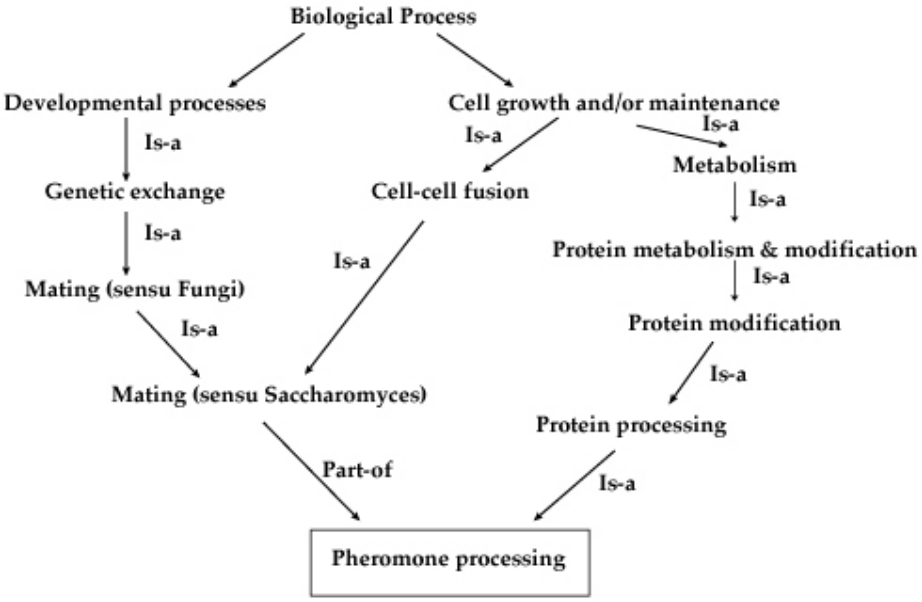


Abbildung 2.4: Ein Auszug aus einer Genontologie, aus [Sac].

Vorgängerknoten, „protein processing“ und „mating (yeast)“ mit unterschiedlichen Beziehungen zu diesen. Während „pheromone processing“ ein Teil des „mating (yeats)“ Prozesses ist, also eine *part-of* Verbindung zu diesem Knoten aufweist, ist er außerdem ein Unterprozess von „protein processing“, was durch eine *is-a* Beziehung ausgedrückt wird. In Abbildung 2.4 wird dieser Zusammenhang verdeutlicht.

Wie Gennamen, können auch Terme aus Genontologien als Knoten in ein assoziatives Netz eingebunden werden, um mit anderen Knoten verknüpft zu werden und somit Beziehungen zu Genen oder Termen aus Texten und Genontologien zu repräsentieren. Ein solches Netz wird im folgenden Kapitel 3 im Detail beschrieben.

Kapitel 3

Das assoziative Netz

In diesem Kapitel wird detailliert auf die Erstellung, Erweiterung und Benutzung des assoziativen Netzwerkes eingegangen. Dabei werden zuerst die Elemente erläutert, aus denen ein solches Netz besteht. Weiter wird auf die Verfahren eingegangen, die im Rahmen dieser Arbeit verwendet wurden, um Terme aus Dokumenten zu gewinnen, beginnend mit der Vorverarbeitung der Texte. Danach wird auf die Erweiterung des Netzes durch Genknoten aus *gene subgroups* und Termknoten des Textkorpus, sowie die Benutzung des Netzes und die Verarbeitung von Anfragen an das Netz eingegangen. Letztlich werden die Funktionsweisen der Server- und Clientkomponenten in Kürze geschildert.

3.1 Die Elemente des Netzes

Ein assoziatives Netz ist ein gerichteter Graph, der aus drei Grundelementen besteht: Knoten, Kanten und Annotationen. Diese Grundelemente und ihre Verwendung werden in den zwei folgenden Abschnitten genauer beschrieben.

3.1.1 Knoten

Die Knoten des Graphen bzw. des assoziativen Netzes repräsentieren die Wissenseinheiten, die durch Kanten, auch Links genannt, miteinander verbunden sind. Diese Knoten können z.B. Terme sein, die aus Dokumenten extrahiert wurden, Namen von Genen, die durch *gene subgroup mining* gefunden wurden oder Genontologieterme. Jeder Knoten hat einen Namen und einen Typ, damit jederzeit festgestellt werden kann, um was für eine Art von Knoten es sich handelt. Innerhalb dieser Arbeit wurde jedoch nur mit zwei verschiedenen Typen gearbeitet: *TERM* und *GENE*.

Weiter wird den Knoten eine oder mehrere Annotationen zugeordnet. Diese Annotationen geben Auskunft darüber, in welchem Kontext die Knoten auftreten. Einem Termknoten werden beispielsweise Dokumentannotationen zugeordnet, welche Daten wie Titel, Autoren, Datei und Häufigkeit beinhalten. Durch diese Dokumentannotationen kann

folglich festgestellt werden, in welchen Dokumenten der Term wie oft aufgetreten ist und in welcher Datei sich dieses Dokument befindet. Einem Genknoten dagegen werden *gene subgroup*-Annotationen zugeordnet. Diese enthalten Informationen über die Experimente, aus denen die Gene hervorgegangen sind bzw. durch welche sie in bestimmte subgroups eingeteilt wurden. Diese Informationen sind z.B.: support, organism, overexpression value und underexpression value. Eine ausführliche Erklärung dieser Werte ist in [Dil06] zu finden.

3.1.2 Links

Wie schon in Abschnitt 3.1.1 erwähnt, sind die Knoten des assoziativen Netzes durch Links verbunden, welchen Gewichte zwischen 0 und 1 zugeordnet werden. Besteht eine starke Beziehung zwischen zwei Knoten, so wird der Link zwischen diesen ein hohes Gewicht haben. Ist die Beziehung zwischen jenen unbedeutend, so wird das Gewicht entsprechend geringer ausfallen. Wie diese Gewichte berechnet werden ist in Abschnitt 3.3.1 beschrieben.

Außer einem Gewicht werden den Kanten sowie den Knoten ein Typ und Annotationen zugeordnet. Der Linktyp besagt, von welcher Art der Link ist. Eine Kante, die durch die Analyse von Dokumenten im Netz erstellt wurde, wird mit dem Typ *TEXT* gekennzeichnet, während eine Kante, die durch *gene subgroup mining* erstellt wurde, vom Typ *GENE_EXPRESSION_EXPERIMENT* ist. Insgesamt wurden drei verschiedene Linktypen verwendet: *TEXT*, *GENE_EXPRESSION_EXPERIMENT* und *SYNONYM*, wobei letzterer eine Synonymbeziehung zwischen Gennamen darstellt.

Für jedes Gen gibt es in der Regel mindestens drei Bezeichnungen. Neben dem üblichen Namen eines Gens existiert eine oder mehrere Affymetrix-Ids. Diese sind vom Hersteller von Genchips namens Affymetrix festgelegte Nummern, die einzelne Gene identifizieren. Außerdem wird jedem Gen zusätzlich eine Beschreibung zugeordnet. In Dokumenten treten sowohl Gennamen als auch dessen Beschreibungen auf. Auch im assoziativen Netz müssen alle Variationen als Knoten repräsentiert werden. Um sicherzustellen, dass erkannt wird, dass es sich nicht um verschiedene Gene handelt, sondern um eines, werden zwischen diesen Knoten Synonymlinks angelegt. Abbildung 3.1 zeigt eine solche Synonymgruppe des Gens „il6“, welches alternativ als „205207_at >“ oder „interleukin 6“ bezeichnet werden kann. Die Knoten sind untereinander mit Kanten vom Typ *SYNONYM* verbunden. Diese haben stets ein Gewicht von 1.0, um die starke Beziehung auszudrücken.

Die Annotationen sind nötig, um festzuhalten, wodurch ein Link zwischen zwei Knoten entstanden ist. Eine Kante kann mehrere Annotationen haben. Tauchen die Namen zweier Knoten in einem Dokument auf, so wird dem entsprechenden Link, der diese Knoten verbindet, eine Dokumentannotation zugewiesen, die wiederum Daten

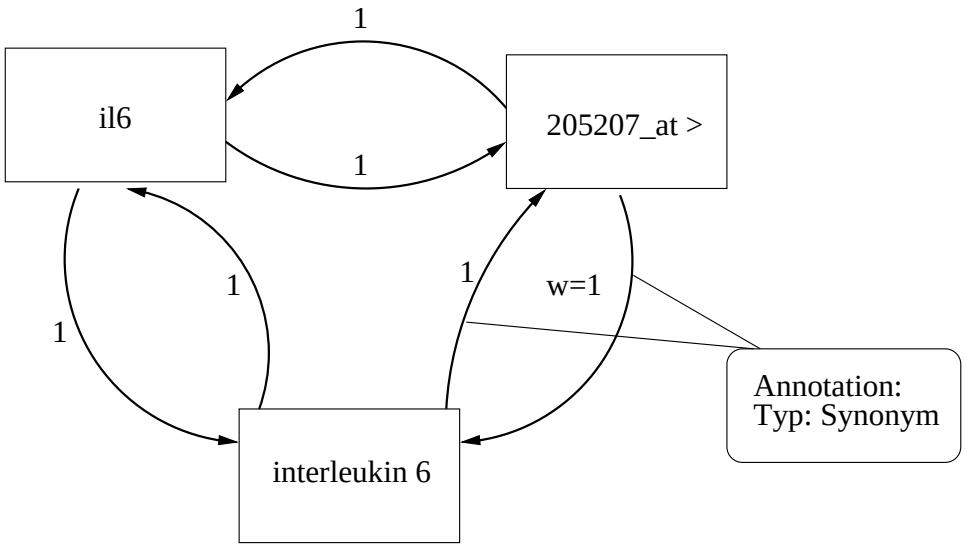


Abbildung 3.1: Eine Synonymgruppe des Gens „il6“, mit den alternativen Bezeichnungen „205207_at >“ und „interleukin 6“. Alle Knoten sind untereinander mit Links von Typ *SYNONYM* verbunden.

wie Titel, Autoren und Dateinamen beinhaltet. Wird durch *gene subgroup mining* eine Beziehung zwischen zwei Genen festgestellt, so wird der Kante, welche die beiden Genknoten verbindet, eine *gene subgroup*-Annotationen zugeordnet. Dadurch kann bei der Auswertung des Netzes festgestellt werden, durch was ein bestimmter Link zustande gekommen ist.

In Abbildung 3.2 ist ein assoziatives Netz mit den vier Knoten „Mensch“, „Diabetes“, „il6“ und „il8“ zu erkennen. Die Termknoten sind durch Kreise gekennzeichnet, die Genknoten durch Rechtecke. Weiter sind den Knoten sowie den Links Annotationen bzw. *AnnotationEntries* zugeordnet. Die Links besitzen außerdem jeweils ein Gewicht.

3.2 Termgewinnung

In den folgenden Abschnitten wird das Verfahren beschrieben, das im Rahmen dieser Arbeit verwendet wurde, um Terme aus bestimmten Textkorpora zu extrahieren. Dabei wird die Vorverarbeitung der Texte und die Termextraktion erläutert.

3.2.1 Vorverarbeitung

Im Folgenden wird die Vorverarbeitung der Texte des Textkorpus dargestellt. Dabei werden zuerst verschiedene Filter und deren Zweck erklärt, gefolgt von der Beschreibung des verwendeten Stemmingverfahrens.

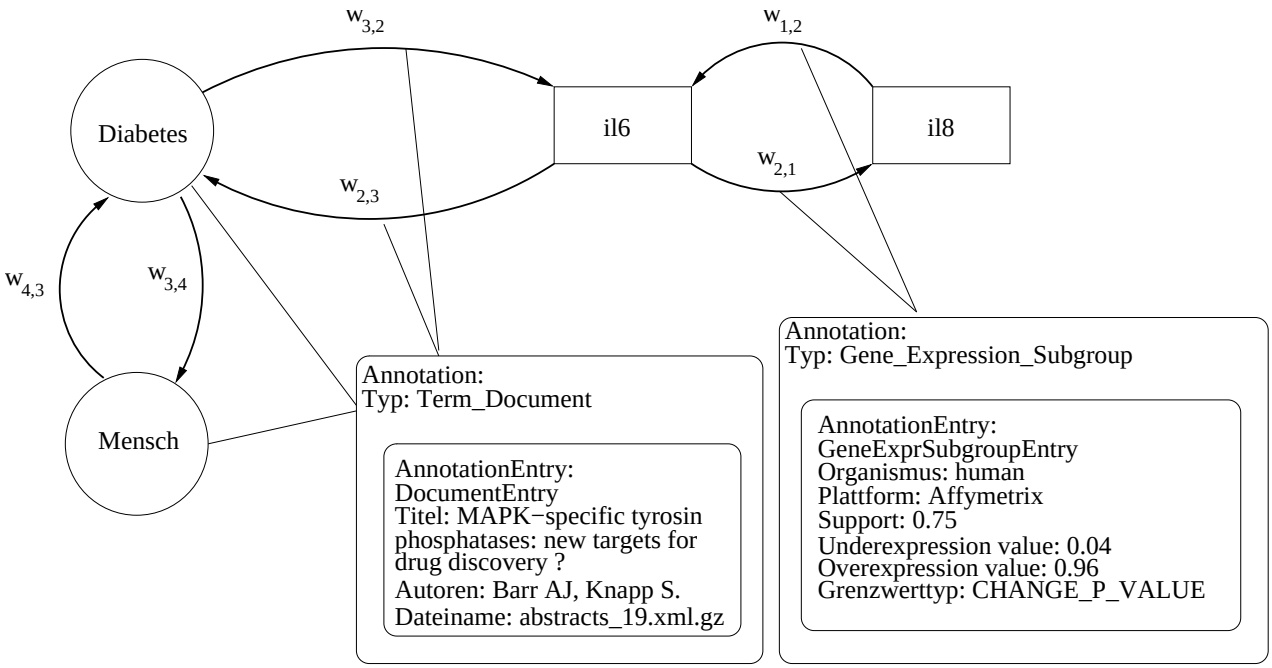


Abbildung 3.2: Ein assoziatives Netz, bestehend aus vier Knoten, sechs Links und zwei Annotationen.

Filter

Die Texte der Textkorpora stammen aus der Artikeldatenbank *PubMed* [pub], auf welche in Abschnitt 4.1 kurz eingegangen wird. Die Texte bestehen aus Zusammenfassungen der Artikel aus der PubMed Datenbank. In diesen Zusammenfassungen tauchen neben den gewünschten Termen unter anderem Wörter und Zeichen auf, die nicht als Term verwendet werden können bzw. sollten. Darunter fallen Wörter, die generell sehr häufig in vielen Texten auftreten und deswegen nur sehr wenig Aussagekraft haben (siehe Zipfsches Gesetz in Abschnitt 2.1.1), wie z.B.: „und“, „oder“, „ich“, „du“ und so weiter. Diese Wörter müssen aus den Texten herausgefiltert werden, bevor die zu verwendenden Terme extrahiert werden. Dazu wird eine Stopwortliste angelegt mit allen Wörtern, die nicht als Terme gelten sollen. Da innerhalb dieser Arbeit mit englischen Texten gearbeitet wurde, besteht die Stopwortliste aus englischen Wörtern.

Weiter müssen alle Satzzeichen eliminiert werden, da diese ebenfalls nicht als Terme gelten und es keinen Sinn machen würde, Satzzeichen als Knoten mit in das assoziative Netz aufzunehmen. Es sind noch weitere Filterarten denkbar, wie beispielsweise Filter, die Ziffern oder generell alle Zeichen entfernen, die keine Buchstaben sind. Dabei muss jedoch darauf geachtet werden, dass nicht zu viel eliminiert wird und evtl. nützliche Informationen bzw. Terme dadurch verloren gehen. In dieser Arbeit wurden deshalb nur Stopwortfilter und Satzzeichenfilter verwendet.

Stemming

Wörter treten in den zu analysierenden Texten durchweg in unterschiedlichen Formen bzw. Morphologien auf. Verben erscheinen in verschiedenen Konjugationsformen und Nomen in verschiedenen Deklinationsformen. Weiter werden die verschiedenen morphologischen Varianten eines Wortes durch Komposition, Dekomposition, Flexion und Hinzufügen von Affixen erzeugt. Jedoch werden die verschiedenen Formen eines Wortes als zusammengehörig oder sogar identisch betrachtet. Würden Terme samt ihrer Form als Termknoten in das assoziative Netz eingebunden werden, so würde ein Term mehrmals in diesem auftreten, für jede denkbare morphologische Form, die in den Texten gefunden wird, einmal. Dies hätte zur Folge, dass sich das Netz unnötig „aufbläht“. Außerdem würden Informationen verloren gehen oder nicht vollständig durch das Netz repräsentiert werden, da ein Wort in einer bestimmten Form zu weiteren Termen Verbindungen aufweisen könnte, derselbe Term, in einer anderen morphologischen Form zu diesen jedoch keine.

Um diese Probleme zu vermeiden, müssen die Terme, bevor sie in das Netz eingebunden werden, auf eine Grund- oder Stammform reduziert werden. Somit fallen die formtypischen Endungen, die z.B. durch Konjugation bzw. Deklination oder das Anhängen von Affixen entstehen, weg. Jeder Term wird folglich nur einmal in das Netz integriert, egal in welcher Form dieser in den Dokumenten auftritt. Der Vorgang der Reduktion von Wörtern auf ihre Grund- oder Stammform wird auch Lemmatisierung oder Stemming genannt ([Fer03], [Lew05]). Dies hat eine starke Reduktion der Anzahl der unterschiedlichen Terme zufolge, wodurch wiederum die Größe des Termindex bzw. des Termnetzes, abhängig vom verwendeten Stemmingverfahren, um 10 bis 50 Prozent reduziert wird [Bel00].

Es gibt verschiedene Stemmingmethoden. Für jede Sprache wird eine individuelle Methode benötigt. Das Problem des Stemmings ist von Sprache zu Sprache unterschiedlich schwer. Während im Englischen dieser Vorgang zufriedenstellend algorithmisch gelöst werden kann, wie durch das Verfahren von Kuhlen [Kuh77] oder den Porter-Stemmer-Algorithmus, [Por97], so ist im Deutschen ein Wörterbuch von Nöten, um diese Aufgabe zu bewerkstelligen. Der Grund liegt in der Unregelmäßigkeit der morphologischen Veränderung der Wörter. In einer Sprache, in der häufig unregelmäßige Verben auftreten oder die Formveränderung nicht stets nach bestimmten Mustern abläuft, kann das Stemming nicht rein algorithmisch geschehen. Es wird ein Wörterbuch benötigt, in welchem die Grund- oder Stammformen für bestimmte Wörter verzeichnet sind. Da innerhalb dieser Arbeit mit englischen Texten gearbeitet wurde, war dies nicht nötig. Als Stemmingverfahren wurde der Porter-Stemmer-Algorithmus eingesetzt.

Termextraktion

Nachdem die Texte durch diverse Filter von unbedeutenden Wörtern und Zeichen bereinigt und die Wörter der Texte durch Stemming auf ihre Stammform reduziert wurden, müssen nun die bezüglich eines Dokumentes einschlägigen Terme erkannt und in das Netz eingebunden werden.

Der Vorgang der Termextraktion kann auf verschiedene Art und Weise geschehen. Essentiell müssen jedoch die Terme als gewichtig oder nicht gewichtig eingestuft werden. Dabei kann beispielsweise die Häufigkeit eines Wortes im Text untersucht werden. Nachdem die häufigen aber oft nicht gewichtigen Wörter, wie z.B.: „und“, durch einen Stopwortfilter entfernt wurden, ist die Wahrscheinlichkeit größer, dass die übrigen häufig auftretenden Wörter auch einschlägig sind. Es stellt sich nur die Frage, wie oft ein Wort auftreten muss, um als gewichtig zu gelten. In [Fer03] wird aufgrund des Zipfschen Gesetzes (Abschnitt 2.1.1), angenommen, dass die Terme mittlerer Häufigkeit bezüglich eines Dokuments am einschlägigsten sind. Neben der Bestimmung der Häufigkeit gibt es weitere Ansätze, die Relevanz eines Terms innerhalb eines Dokuments zu bestimmen, wie durch die Einbeziehung von Strukturinformationen. So könnten die im Titel auftretenden Terme stets als einschlägig gewertet werden, egal mit welcher Häufigkeit sie auftreten. Weiter könnten semantische Informationen in die Relevanzbewertung einfließen. Auf diese und weitere Ansätze wird hier jedoch nicht eingegangen.

Innerhalb dieser Arbeit wurde ein Algorithmus verwendet, welcher zum einen lediglich die Häufigkeit eines Terms innerhalb eines Dokuments bestimmt und diesen ab einer gewissen Grenze als gewichtig beurteilt. Dabei wird keine Rücksicht auf Satzgrenzen, Wortstellung oder Semantik genommen. Der Standardwert, der hier als Häufigkeitsgrenze verwendet wurde, liegt bei 3. Terme, die mindestens mit dieser Häufigkeit auftreten, gelten als gewichtig und werden in das assoziative Netz als Knoten eingefügt. Zum anderen wurde die Anzahl der Dokumente gezählt, in welchen die Wörter vorkommen und eine Ober- und Untergrenze festgelegt. Terme, die in allen oder sehr vielen Dokumenten auftreten, wurden ignoriert, da sie keine spezielle Information enthalten. Terme, die nur in einem oder sehr wenigen Dokumenten auftreten, wurden ignoriert, um die Anzahl der Terme und somit auch der Knoten weiter zu beschränken, da die Verknüpfung der Knoten untereinander sehr viel Rechenzeit beansprucht, was in Abschnitt 3.3.1 beschrieben ist.

Es hat sich jedoch gezeigt, dass die Anzahl der Terme, die in nur wenigen Dokumenten auftauchen, wesentlich größer ist, als die Anzahl jener, die in vielen Dokumenten auftreten. Dies zeigte ein Versuch, in dem 99552 Zusammenfassungen medizinischer wissenschaftlicher Artikel zum Thema „Mensch und Diabetes“ der Datenbank *PubMed* untersucht wurden. Abbildung 3.3 illustriert, dass der Großteil der Terme nur in maximal drei unterschiedlichen Dokumenten vorkommt. Durch das Setzen einer Untergrenze werden demnach sehr viele Wörter ignoriert, was von großem Nachteil sein kann. In den

Experimenten, welche in Kapitel 4 beschrieben sind, wurde stets versucht einen Kompromiss zwischen dem Verlust zu vieler Terme und der benötigten Rechenzeit zu finden.

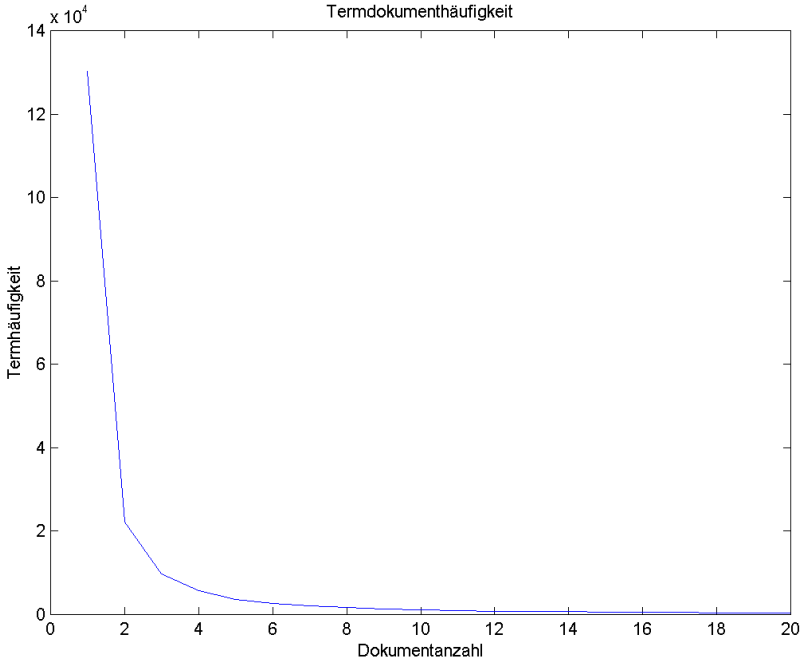


Abbildung 3.3: Die Anzahl der Wörter, die nur in sehr wenigen Dokumenten auftreten, ist bedeutend größer als die, der Wörter, welche in vielen auftreten.

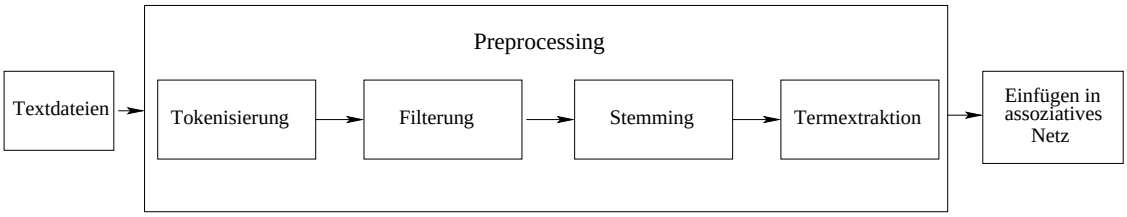


Abbildung 3.4: Die Pipeline mit den Verarbeitungsschritten zur Gewinnung von Termen aus Dokumenten beginnend mit der Auswahl der Textdateien als Korpus gefolgt von verschiedenen Vorverarbeitungsschritten wie Filterung, Stemming und Termextraktion, schließlich endend mit der Einfügung der Terme in das assoziative Netz als Termknoten.

In Abbildung 3.4 ist eine Pipeline an Verarbeitungsschritten zur Gewinnung von Termen aus Dokumenten zu erkennen. Diese beginnt mit der Auswahl der Texte und der Zusammenstellung des Korpus, gefolgt von verschiedenen Vorverarbeitungsschritten wie der Filterung der Texte von nicht einschlägigen Wörtern und Zeichen, der Reduktion der Terme auf deren Stammform durch Stemming und schließlich der Relevanzbewertung der Terme bzw. deren Extraktion. Am Ende der Pipeline werden die gewichtigen Terme in das assoziative Netz als Termknoten eingebracht.

3.3 Einfügen von Knoten

Wie Terme als Termknoten bzw. Gennamen als Genknoten in das assoziative Netz eingefügt werden, wird in den folgenden Abschnitten beschrieben.

3.3.1 Einfügen von Termknoten

Nachdem die Vorverarbeitungsphase abgeschlossen wurde, müssen nun die extrahierten Terme in das assoziative Netz als Termknoten eingefügt werden. Dazu muss für jeden einzufügenden Term ein Termknoten erstellt werden, falls für diesen noch keiner existiert. Weiter muss dem Knotenpunkt eine Dokumentannotation zugeordnet werden, welche Daten über den Titel des Textes, in dem das Wort auftritt, dessen Autoren usw. enthält. Dies ist nötig, um später festzustellen, in welchen Dokumenten ein Term vorkommt. Existiert bereits ein Termknoten für den zu bearbeitenden Term, so wird lediglich eine weitere Annotation hinzugefügt.

Sind Termknoten und Annotationen erstellt, muss der Knotenpunkt durch Links mit anderen Term- oder Genknoten verbunden werden. Besteht eine Beziehung mit anderen Knotenpunkten so wird eine Kante eingefügt. Je stärker diese Beziehung ist, desto stärker muss das Gewicht des Links ausfallen. Besteht keine Beziehung zwischen zwei Knoten wird auch kein Link eingetragen. Folglich muss zwischen den neuen Knotenpunkten und allen bereits existierenden ein Gewicht berechnet werden, welches innerhalb dieser Arbeit stets zwischen 0 und 1 liegt. Wird ein Gewicht von 0 berechnet wird keine Kante eingefügt. Da für jeden Knotenpunkt die Gewichtung der Kanten zu allen bestehenden Knoten berechnet werden muss, um evtl. einen Link einzutragen, ist der Aufwand hierfür quadratisch.

Gewichtsberechnung

Für die Gewichtsberechnung zwischen zwei Termknoten oder zwischen Term- und Genknoten wird das gemeinsame Auftreten der Terme bzw. der Terme und Gennamen, innerhalb der Dokumente des Korpus bestimmt. Je öfter zwei Terme gemeinsam in verschiedenen Texten auftreten, desto größer ist deren Beziehung zueinander. Die Linkgewichte w_{ij} und w_{ji} zwischen zwei Knoten k_i und k_j werden dabei folgendermaßen berechnet:

$$w_{ij} = \frac{\sum_{x=1}^{n} c_{xij}}{\sum_{x=1}^{n} c_{xi}}$$

$$w_{ji} = \frac{\sum_{x=1}^{n} c_{xij}}{\sum_{x=1}^{n} c_{xj}}$$

Mit

$$c_{xi} = \begin{cases} 1 & wenn\ Term\ t_i\ in\ Dokument\ d_x\ auftritt \\ 0 & sonst \end{cases}$$

und

$$c_{xij} = \begin{cases} 1 & wenn\ Term\ t_i\ und\ t_j\ in\ Dokument\ d_x\ auftreten \\ 0 & sonst \end{cases}$$

Dabei ist n die Anzahl der Dokumente. Analog zu c_{xij} wird c_{xji} bestimmt. Diese Art der Gewichtsberechnung ist an das Verfahren aus [CBN95] angelehnt und berücksichtigt lediglich das gemeinsame Auftreten von Termen oder Gennamen innerhalb eines Textes. Es wird nicht berücksichtigt, ob die Wörter im selben Satz auftreten oder einen semantischen Zusammenhang aufweisen. In diesen Punkten kann die Gewichtsberechnung verbessert werden.

Ein weiteres Problem ist bei häufig auftretenden Termen abzusehen. Existieren Wörter, welche in sehr vielen Texten des Korpus vorkommen, so sind diese bezüglich eines Themas weniger einschlägig als Terme die nur in bestimmten Texten erscheinen, die dieses Thema behandeln. Wird beispielsweise ein Korpus aus allen Abstracts der PubMed Artikeldatenbank zum Thema „gene“ zusammengestellt, so kommt das Wort „gene“ in so gut wie allen Texten vor. Die Relevanz des Terms ist aber sehr gering, da ohnehin alle Texte über das Thema „gene“ berichten. Da der Term in vielen Texten auftritt, werden dementsprechend viele gemeinsame Vorkommen mit anderen Termen festgestellt, was dazu führt, dass viele Knoten Links mit hohen Gewichten zum Termknoten „gene“ aufweisen. Dies verzerrt jedoch die Beurteilung der Einschlägigkeit des Terms, da dieser von sehr vielen Knoten aus über Links mit hohen Gewichten erreicht werden kann.

Um dieses Problem zu lösen wurde im Rahmen dieser Arbeit die Gewichtsbestimmung insofern erweitert, als die durch die oben aufgeführte Formel berechneten Gewichte mit der inversen Dokumenthäufigkeit *IDF* (*inverse document frequency*), aus [Kor97], des entsprechenden Terms multipliziert werden. Das Gewicht w_{ij} des Links von Knoten k_i zu Knoten k_j wird also folgendermaßen bestimmt:

$$w_{ij} = \frac{\sum_{x=1}^{n} c_{xij}}{\sum_{x=1}^{n} c_{xi}} \cdot IDF_j$$

Mit

$$IDF_j = \log\left(1 + \frac{n}{\sum_{x=1}^{n} c_{xj}}\right).$$

Durch das Einführen der inversen Dokumenthäufigkeit werden Gewichte zu Termen, die in sehr vielen Dokumenten auftreten, geringer bewertet.

3.3.2 Einfügen von Genknoten

An dieser Stelle wird beschrieben, wie Genknoten in das assoziative Netz eingefügt werden. Das *gene subgroup mining* aus [Dil06] liefert Gruppen von Genen zurück, die in Beziehung zueinander stehen. Diese Gene werden als Genknoten in das assoziative Netz eingefügt. Das Gewicht der Links untereinander entspricht dem Support-Wert, der durch das *gene subgroup mining* bestimmt wurde. Dieser Wert liegt ebenfalls zwischen 0 und 1. Je größer er ist, desto stärker stehen die Gene in Beziehung.

Die Genknoten wurden innerhalb dieser Arbeit vor den Termknoten eingefügt. Falls in Texten bereits bestehende Gennamen auftreten, werden diese dadurch nicht als Termknoten in das Netz eingebunden, sondern es werden Kanten von in Beziehung stehenden Wörtern zu diesen Genknoten eingefügt. Somit werden Termknoten mit Genknoten verbunden. Dabei wird das Gewicht zwischen Termknoten und Genknoten wie das Gewicht zwischen Termen berechnet. Werden Wörter aus Dokumenten und Gennamen aus *gene subgroup mining*-Prozessen als Knotenpunkte miteinander verbunden, so findet an dieser Stelle die assoziative Verknüpfung von Informationen aus zwei heterogenen Datenquellen statt. Wenn später bei der Bearbeitung von Anfragen diesen Kanten nachgegangen wird, so kann eine Verbindung zwischen Gengruppen und Termen bzw. Dokumenten und *gene subgroup mining*-Experimenten erkannt werden.

Ein Gen kann durch verschiedene Synonyme beschrieben werden, wie schon in Abschnitt 3.1.2 geschildert. Für jedes Synonym wird ein Genknoten in das Netz eingefügt und die Knotenpunkte werden mit Links vom Typ *SYNONYM* verbunden. Die Gewichte dieser Synonymlinks haben stets den Wert 1, um die starke Beziehung untereinander zum Ausdruck zu bringen. Ein Beispiel für eine solche Synonymgruppe ist in Abbildung 3.1 zu sehen.

3.4 Bearbeitung der Anfragen

Dieser Abschnitt beschreibt die Bearbeitung der Anfragen bzw. das Durchsuchen des Netzes. Hierzu gibt es viele verschiedene Methoden. Es wird jedoch im Detail nur auf einen Ansatz eingegangen, einer modifizierten Form des *Branch-and-Bound-Spreading-Activiation*-Algorithmus aus [CBN95].

Generell sind verschiedene Möglichkeiten denkbar, verbindungsbasierte Modelle zu durchsuchen. Dazu gehören auch Algorithmen, mit denen Hopfield-Netze ausgewertet werden, wie im Abschnitt 2.1.4 beschrieben. Das Netz besteht allerdings aus einem gerichteten Graph, was bedeutet, dass die Linkgewichte nicht symmetrisch sind. Eine unsymmetrische Gewichtsmatrix als Hopfield-Netz kann zu Problemen bei der Konvergenz des Netzes führen ([Roj93], [Sch97], [TH87]). Es kann passieren, dass das Hopfield-Netz im Falle einer Anfrage, also der Aktivierung bestimmter Knoten, keinen stabilen Zustand

mehr einnimmt und zu oszillieren beginnt.

3.4.1 Branch-and-Bound-Suche

In dieser Arbeit wurde eine modifizierte Form des *Branch-and-Bound-Spreading-Activiation*-Algorithmus aus [CBN95] verwendet. Anfänglich werden durch den Benutzer verschiedene Anfrageterme ausgewählt. Zu diesen Termen sollen weitere Terme oder Gene gefunden werden, die mit den Anfragetermen in Beziehung stehen. Diese werden *related terms*, also zugehörige oder verwandte Terme genannt. Die Dokumente oder *gene subgroup mining*-Experimente, die die zugehörigen Terme oder Gennamen enthalten, sind für den Benutzer potentiell relevant.

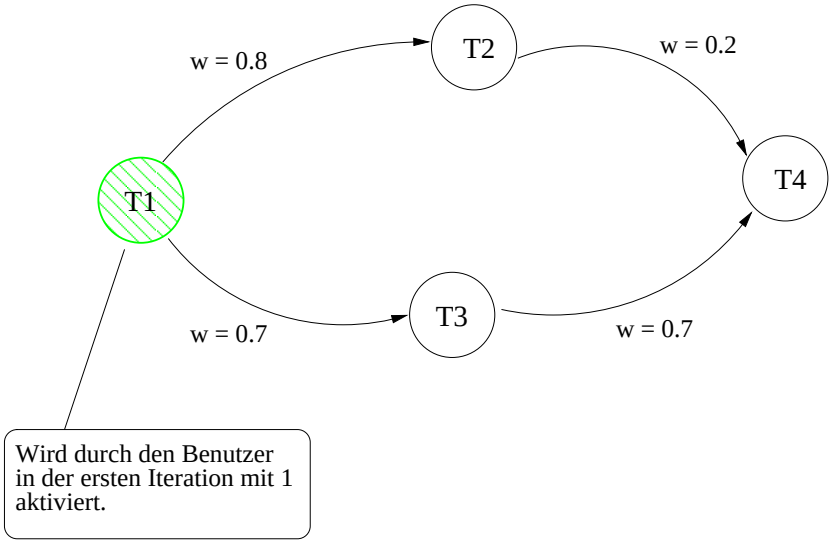


Abbildung 3.5: Die erste Iteration einer Branch-and-Bound-Suche, in welcher der Knoten T1 angeregt wird. Aktivierte Knoten sind grün eingefärbt.

Die Knotenpunkte, welche die vom Benutzer ausgewählten Anfrageterme repräsentieren, werden in der ersten Iteration mit einem Gewicht von 1 aktiviert. Abbildung 3.5 zeigt eine solche Situation. Der Knoten T1 wird in der ersten Iteration angeregt, was durch eine grüne Einfärbung gekennzeichnet ist.

Diese Knoten regen in der zweiten Iteration ihre direkten Nachbarn an. Dabei wird das Gewicht des Links zum ersten Knoten mit dem Gewicht des Links zu dessen Nachbarn multipliziert. Der daraus resultierende Wert wird als Aktivierungswert der Nachbarn verwendet. Da die ersten Knoten nicht über Verbindungen angeregt wurden, sondern über eine Benutzereingabe, wird hier statt des Linkgewichts eine anfängliche Erregung von 1 verwendet. Dieser Sachverhalt ist in Darstellung 3.6 illustriert. Hier wird „T1“ durch den Benutzer mit dem Wert 1 angeregt und verbreitet eine Aktivierung von

0.8 ($1 * 0.8$) zu „T2“ und von 0.7 ($1 * 0.7$) zu „T3“. Es kann vorkommen, dass ein Knoten in einer Iteration durch verschiedene Nachbarknoten angeregt wird. Abbildung 3.7 zeigt wie „T2“ und „T3“ den Knoten „T4“ anregen. Der gesamte Wert der Aktivierung ergibt sich aus der Summe der einzelnen Aktivierungen, in diesem Fall $0.4 + 0.49$. Es muss allerdings entschieden werden, welches Gewicht der eingehenden Links verwendet wird, um die Verbreitung der Aktivität von „T4“ in der nächsten Iteration zu berechnen. Würde ein Mittelwert gebildet, so würden Knoten mit wenigen starken und vielen schwachen eingehenden Verbindungen benachteiligt, gegenüber Knoten, welche nur wenige starke eingehende Kanten besitzen. Aus diesem Grund wird stets das höchste Gewicht als Aktivierungswert für dessen Nachbarn in der nächsten Iteration verwendet, was im Falle von „T4“ 0.49 ($1 * 0.7 * 0.7$) ist. Dieser Wert muss jedoch ebenfalls, wie bereits beschrieben, mit dem Linkgewicht zu diesen Nachbarn multipliziert werden. Abbildung 3.7 zeigt wie von „T4“ aus ein weiterer Knoten mit dem Wert $0.49 * 0.5$ angeregt werden kann.

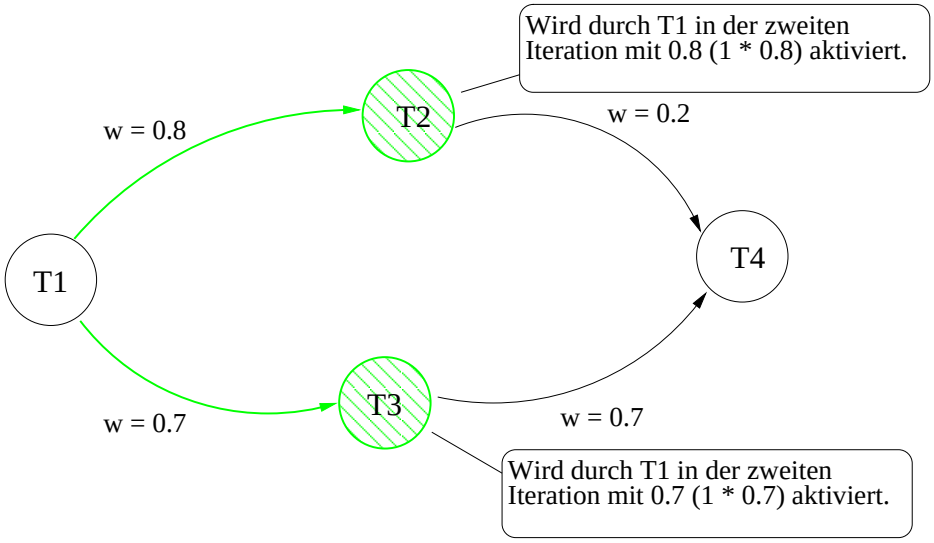


Abbildung 3.6: Die zweite Iteration einer Branch-and-Bound-Suche, in welcher die direkten Nachbarknoten des Knotenpunktes T1 angeregt werden. Aktivierte Knoten sind grün eingefärbt.

In der dritten Iteration werden wiederum die direkten Nachbarn der bereits aktivierten Knoten angeregt. Werden Knoten in verschiedenen Iterationen aktiviert, so werden die Aktivierungswerte aufsummiert. Durch diese Addition können die Werte folglich über 1 steigen. Deshalb wird, wie bereits erwähnt, nicht der Aktivierungswert des Quellknotens verwendet, um einen direkten Nachbarn anzuregen, sondern das Gewicht des Links zu dem Quellknoten, über den dieser angeregt wurde. Somit bleibt der Wert, mit dem ein Knotenpunkt einen anderen anregt stets zwischen 0 und 1 und nimmt von Iteration zu Iteration stetig ab. Dies kann als Abbruchkriterium benutzt werden, was im folgenden noch beschrieben wird.

Abbildung 3.7 illustriert die dritte Iteration einer Branch-and-Bound-Suche, in welcher die Nachbarknoten der in der zweiten Iteration angeregten Knotenpunkte aktiviert werden.

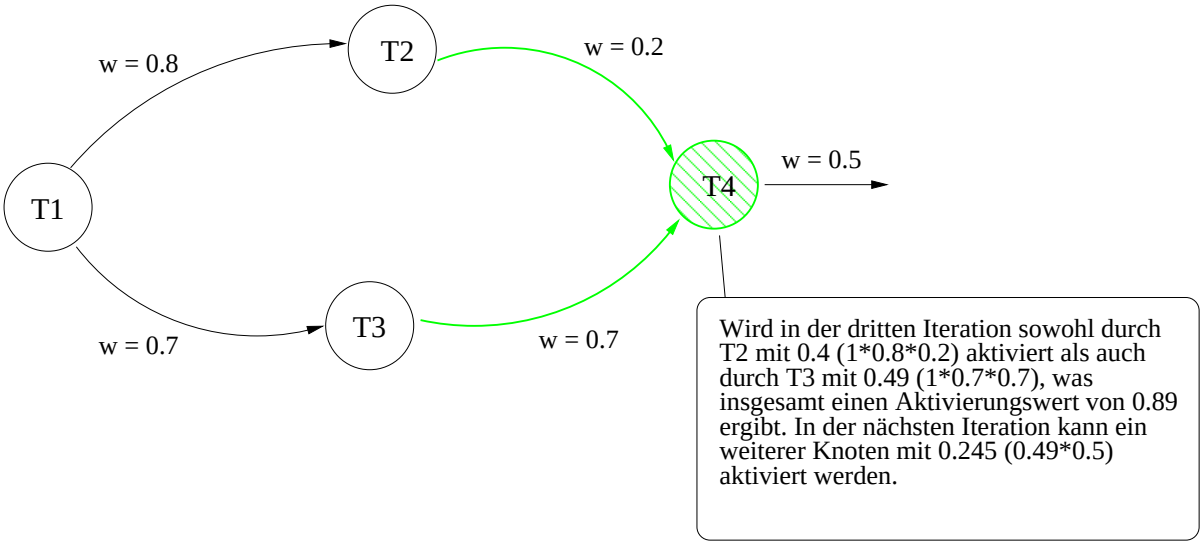


Abbildung 3.7: Die dritte Iteration einer Branch-and-Bound-Suche, in welcher der direkte Nachbar der bereits angeregten Knoten aktiviert wird. Aktivierte Knoten sind grün eingefärbt.

Die Suche kann aufgrund unterschiedlicher Kriterien abgebrochen werden: zum einen kann eine maximale Anzahl an Iterationen festgelegt werden. Wenn diese erreicht ist, werden die aktivierten Knoten absteigend nach ihren aufsummierten Aktivierungswerten sortiert und als Ergebnisterme ausgegeben. Zum anderen kann eine maximale Anzahl an Ergebnistermen festgelegt werden. Wurden dementsprechend viele Knoten aktiviert, wird die Suche beendet und die Ergebnisterme, wiederum sortiert nach deren Aktivierungswerten, zurückgeliefert. In dieser Arbeit wurde zudem ein minimales Linkgewicht eingeführt. Wird dieser Wert bei der Multiplikation des Aktivierungswertes mit dem Gewicht des Links unterschritten, wird dem Link nicht gefolgt und der Knoten am Ende des Links wird nicht aktiviert. Dadurch werden nur Knoten aktiviert, die durch genügend starke Links miteinander verbunden sind, also eine engere Beziehung aufweisen. Knoten, deren Beziehung nur schwach ist, werden somit nicht in die Ergebnisliste aufgenommen. Da die Aktivierungswerte von Iteration zu Iteration geringer werden, wird somit auch die räumliche Ausbreitung der Aktivierung im Netz eingeschränkt und es werden die Knoten, die nur über viele Verbindungen zu erreichen sind, einen geringen Wert aufweisen und somit nicht oder am Ende der Ergebnisliste stehen. Dies gilt jedoch nur wenn die Links nicht vom Typ *SYNONYM*, mit einem Gewicht von 1 sind.

Durch eine Branch-and-Bound-Suche entsteht ein Teilgraph, der sich aus allen aktivierten Knoten und den verfolgten Links zusammen setzt. Ein solcher Teilgraph wird in Abbildung 3.8 gezeigt. Hier wurde nach den Gennamen „il6“ und „il8“ gesucht. Diese Anfrageterme sind in der Darstellung grün gekennzeichnet. Termknoten sind durch Ellipsen dargestellt und Genknoten durch Rechtecke. Die Synonymgruppen verschiedener Gene, bestehend aus Affymetrix-Id, Genbeschreibung und Genname, sind deutlich zu erkennen. In Kapitel 4 wird genauer auf weitere Experimente und deren Resultate eingegangen.

NOT-Terme

NOT-Terme, also Terme die durch den Benutzer von der Suche ausgeschlossen werden, werden anfänglich mit dem Wert -1 aktiviert. Die Verbreitung der Anregung wird wie bei normalen Termen berechnet, mit dem Unterschied, dass diese hier negativ ist. Dies hat zur Folge, dass bestimmte Bereiche an Knoten des Netzes nicht in die Ergebnisliste kommen bzw. am Ende dieser erscheinen, wenn sich deren positiver Aktivierungswert durch negative Anregung verringert.

Ein Abbruchkriterium, wie ein minimaler Aktivierungswert, hätte bei negativen Anregungen wenig Sinn. Deshalb wird hier der absolute Wert mit diesem verglichen. Somit verbreitet sich die negative Aktivierung genau so wie die positive. Dadurch kann es allerdings passieren, dass anfänglich positiv angeregte Knoten im Laufe der Iterationen einen negativen Aktivierungswert bekommen und vice versa.

3.4.2 Nachverarbeitung des Resultats

Sind nun verwandte Terme und Gennamen gefunden worden, so müssen zusätzlich die Dokumente und *gene subgroup mining*-Experimente ausgemacht werden, in denen die Terme oder Gennamen auftauchen. Dies geschieht durch die Auswertung der Knotenannotationen. Für jeden Knoten werden alle Annotationen bestimmt. Jeder wird der gesamte Aktivierungswert des Knoten zugeordnet. Sind bestimmte Annotationen mehreren Knoten zugeordnet, werden die Aktivierungswerte für diese aufsummiert. Sortiert man die Annotationen schließlich absteigend nach ihren zugeordneten Aktivierungswerten, so erhält man die Dokumente und *gene subgroup mining*-Experimente, die durch die Annotationen repräsentiert werden, in der Reihenfolge ihrer Relevanz bezüglich der Anfrage. In dieser Arbeit wurde außerdem die Möglichkeit implementiert, verwandte Terme von Knoten mit negativen Aktivierungswerten gänzlich aus der Ergebnisliste zu tilgen.

3.5 Server und Client

Da die Erstellung eines assoziativen Netzes aufgrund des quadratischen Aufwands unter Umständen sehr lange dauern kann, vor allem wenn große Mengen an Texten

oder Gengruppen verarbeitet werden müssen, ist es sinnvoll, ein solches Netz nur einmal auf einem Rechner zu erzeugen. Benutzer, die an dieses Netz Anfragen senden wollen, können dies über ein spezielles Clientprogramm, was auf deren Rechner läuft, bewerkstelligen.

Aus diesem Grund wurde ein Serverprogramm entwickelt, welches ein assoziatives Netz aus verschiedenen Datenquellen, hier Dokumente und Gengruppen aus *gene subgroup mining*-Experimenten, aufbauen und Anfragen an dieses Netz bearbeiten kann. Das Serverprogramm hört auf einem bestimmten Port und nimmt Anfragen eines Clientprogramms entgegen. Diese werden nach dem in Abschnitt 3.4 beschriebenen Prozess bearbeitet und es wird schließlich eine Antwort an den Client zurückgeschickt.

Eine Anfrage besteht aus Anfrage- oder Suchtermen, einem minimalen Linkgewicht, einer maximalen Anzahl an Ergebnisdokumenten und Ergebnistermen und einer Angabe, welche Methode zur Behandlung der NOT-Terme angewendet werden soll. Eine Antwort des Servers besteht aus verwandten Termen, Dokumenten mit Angaben zu Titel, Autoren und Dateinamen und *gene subgroup mining*-Experimenten. Sowohl das Anfrage- als auch das Antwortformat sind XML-Formate. Die DTDs dazu sind im Anhang A zu finden.

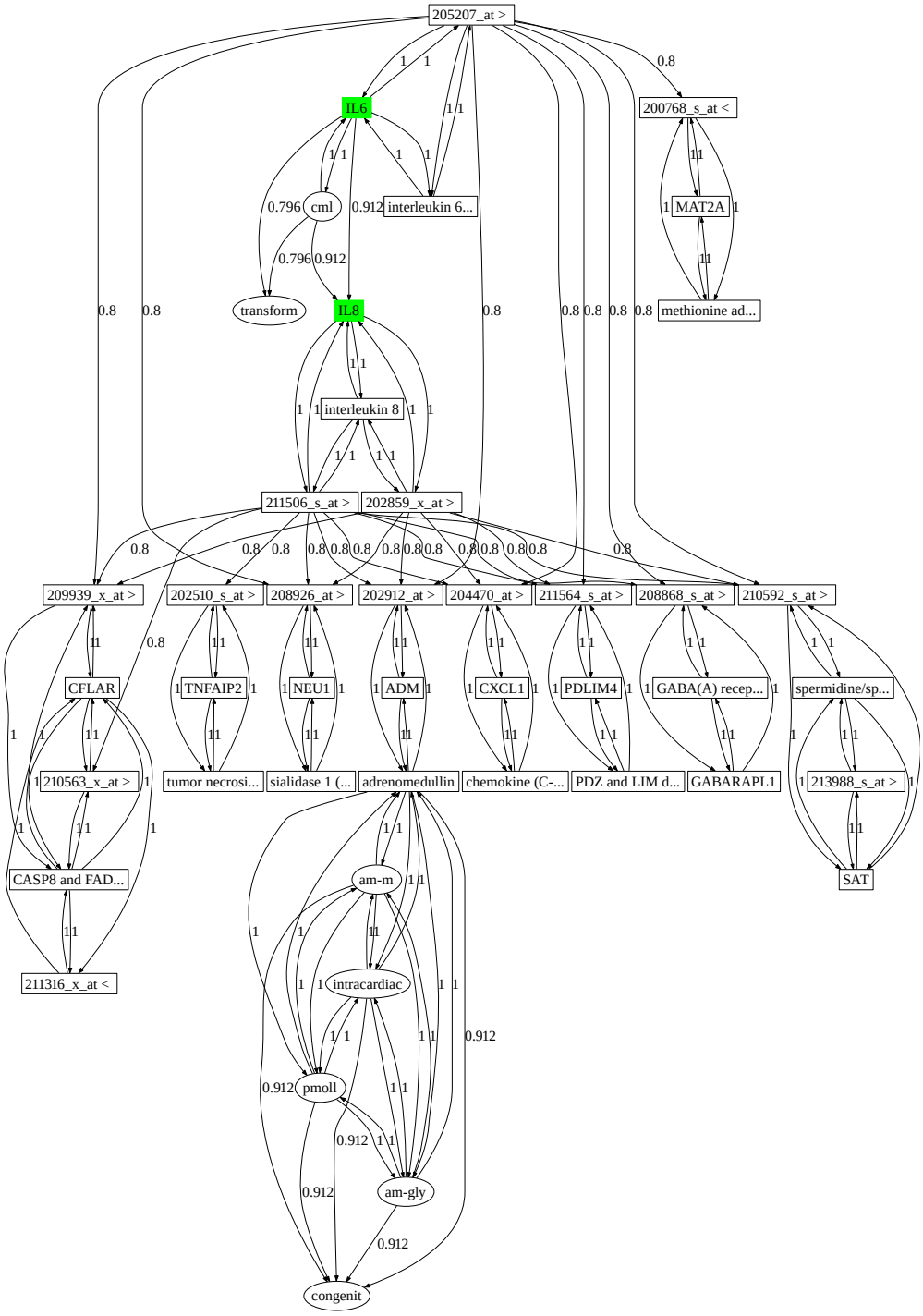


Abbildung 3.8: Teilgraph einer Suche mit den Gennamen „il6“ und „il8“ als Anfrageterme (grün eingefärbt). Termknoten sind als Ellipsen gekennzeichnet und Genknoten als Rechtecke.

Kapitel 4

Experimente

Dieses Kapitel beschreibt die Experimente, die mit dem assoziativen Netz durchgeführt wurden, um dieses auf seine Tauglichkeit zu testen. Zuerst wird kurz auf die Datenquellen und ihre Herkunft eingegangen, mit deren Daten das Netz trainiert wurde. Anschließend werden die verschiedenen Experimente und ihre Ergebnisse erläutert, die im Einzelnen durchgeführt wurden.

4.1 PubMed

PubMed (www.pubmed.com), die kostenfreie Version von MEDLINE, ist eine textbasierte, bibliografische Referenzdatenbank, in der englischsprachige Artikel der Sachgebiete Medizin, Zahnmedizin, Veterinärmedizin, öffentliches Gesundheitswesen, Psychologie, Biologie, Genetik, Biomedizin etc., aus Fachzeitschriften dokumentiert und Links auf Volltextzeitschriften gespeichert werden. Derzeit enthält PubMed etwa 15 Millionen Referenzen auf Zeitschriftenartikel. Die Datenbank wurde vom nationalen Zentrum für Biotechnologische Informationen (National Center for Biotechnology Information, kurz NCBI) für die nationale medizinische Bibliothek der USA (United States National Library of Medicine, kurz NLM) entwickelt. PubMed ist über das zentrale, textbasierte Suchsystem NCBI-Entrez erreichbar. Außerdem können über NCBI-Entrez Nukleotid- und Protein-Sequenzen, Protein-Strukturen, komplexe Genome und weitere wichtige Datenbanken, wie PubChem durchsucht werden.

4.2 Genexpressionsdaten

Die Genexpressionsdaten, die für *gene subgroup mining*-Experimente verwendet wurden, um Gengruppen zu finden, sind Public Domain Daten, die aus Affymetrix-Chips mit Menschengenen, mit Schwerpunkt auf Diabetes, gewonnen wurden und stammen aus der Gendatenbank RAD (*RNA Abundance Database*)[1].

[1] Url der Daten: www.cbil.upenn.edu/RAD/php/displayStudy.php?study_id=2160

4.3 Experimente

Im folgenden werden verschiedene Experimente erläutert, die durchgeführt wurden, um die Tauglichkeit des assoziativen Netzes zu prüfen. Hierzu wurden als heterogene Datenquellen zum einen menschliche Genexpressionsdaten, deren Gene durch *gene subgroup mining* zu Gengruppen zusammengefasst wurden und zum anderen Zusammenfassungen wissenschaftlicher Artikel, die bei einer Suche zu bestimmten Suchtermen in der PubMed Datenbank als Ergebnis gefunden wurden, verwendet. Die Suchterme wurden so ausgesucht, dass zwischen ihnen und den Genen bzw. Gengruppen, die in einem Experiment verwendet wurden, ein Zusammenhang besteht. Z.B. wurde bei der Verwendung von menschlichen Genen, die in Bezug zu Diabetes stehen, in der PubMed Datenbank nach Dokumenten mit den Termen „human“ und „diabetes“ gesucht.

Für jedes Experiment wurde sowohl die Zeit zur Erstellung des Netzes und die zur Berechnung der Anfragen gemessen als auch der Speicherbedarf eines fertigen Netzes. Hierfür wurde ein Rechner verwendet, dessen wichtigste Daten in Tabelle 4.1 aufgeführt sind.

CPU	Intel Pentium 4, 2,8 GHz
RAM	1 GB (2 GB Swap)
Betriebssystem	SuSE Linux 9.3, Kernel: 2.6.11.4-21.13-smp

Tabelle 4.1: Daten des Rechners, der zur Erstellung der Netze verwendet wurde.

4.3.1 CCL20-Experiment

In diesem Experiment wurden 231 Zusammenfassungen der PubMed Datenbank zu dem Suchterm „CCL20“ als Textkorpus verwendet. *Chemokine (C-C motif) Ligand 20*, kurz „CCL20“, ist ein menschliches Gen, das Funktionalitäten im Zusammenhang mit dem Immunsystem aufweist. Weiter wurden Ergebnisse eines *gene subgroup mining*-Prozesses mit menschlichen Genexpressionsdaten verwendet.

Zuerst wurde ein assoziatives Netz aus den Genen erstellt, anschließend wurden 1606 verschiedene Terme aus dem Textkorpus extrahiert und als Knoten in das Netz eingefügt. Die verwendete minimale Termhäufigkeit, die Häufigkeit mit der ein Term in einem Dokument auftritt, betrug 2. Eine minimale oder maximale Termdokumenthäufigkeit, die Anzahl der Dokumente in denen ein Term auftritt, wurde nicht benutzt. Insgesamt bestand das Netz aus 1949 Knoten und 105626 Links. Tabelle 4.2 führt die Gene auf, welche sowohl in den Genexpressionsdaten auftreten als auch in den verwendeten Dokumenten. Würden keine Übereinstimmungen dieser Art auftreten, so wären das Netz der Gennamen und das der Terme separiert und es könnten keine zusätzlichen Informationen aus der Zusammenführung der Datenquellen gewonnen werden, da

keine gemeinsame Schnittmenge der Datenquellen besteht. Um die Verknüpfung der Informationen dieser beiden Datenquellen zu analysieren, ist es also sinnvoll, nach den gemeinsam auftretenden Termen bzw. Genen zu suchen.

Genname	Häufigkeit
IL8	6
IL6	6
CCL20	343
CXCL1	4
GAPDH	1
SET	7

Tabelle 4.2: Gennamen, die sowohl in den Genexpressionsdaten als auch in den Dokumenten des Textkorpus „CCL20“ auftreten.

Mit dem Testrechner (siehe Tabelle 4.1) wurden insgesamt 1414, 3 Sekunden (23, 6 Minuten) benötigt, um den Textkorpus vorzuverarbeiten, die extrahierten Terme als Knoten in das Netz einzufügen und die Links und deren Gewichte zu den anderen Knoten zu berechnen. 22, 7 Sekunden dauerte die Vorverarbeitung und 1391, 6 Sekunden (23, 3 Minuten) das Einfügen der Knoten und Links. Für den benötigten Speicherplatz des gesamten Netzes, nach der Erstellung wurden 9633720 Bytes (ca. 9, 2 MB) gemessen.

Il6 Il8

Die Gennamen „Il6 Il8“ wurden als Suchterme verwendet und die Suche im assoziativen Netz mit den Einstellungen aus Tabelle 4.3 gestartet und dauerte 0, 86 Sekunden. Das minimale Linkgewicht ist die wichtigste Einstellung, da diese festlegt, welche und zum Teil auch wie viele Terme gefunden werden. Ein zu hohes minimales Linkgewicht verhindert, dass genug Terme aus Dokumenten in die Ergebnisliste aufgenommen werden, da die Gewichte der Verbindungen zwischen Genen aus Gengruppen, die durch den Support aus den *gene subgroup mining*-Experimenten bestimmt werden, meist höher sind, als die Linkgewichte zu Knoten von Dokumenttermen. So kann es sein, dass den Verbindungen zu diesen Termen aufgrund des zu hohen minimalen Linkgewichts nicht nachgegangen wird, obwohl diese als einschlägig einzustufen wären. Die Stärke des Gewichts der Verbindungen zu Dokumenttermen hängt jedoch wesentlich von den Dokumenten des Textkorpus ab. Ein thematisch breit gefasster Korpus wird geringere Linkgewichte zur Folge haben, da viele der einzelnen Terme keinen Bezug zueinander haben. Weiter kann die Folge eines zu hohen minimalen Linkgewichts sein, dass generell weniger Gennamen oder Terme als Ergebnis zurückgeliefert werden als erwünscht. In diesem Fall muss die Suche wiederholt mit einem niedrigerem minimalen Linkgewicht wiederholt werden. Als Verbesserung wäre denkbar, dass der Algorithmus eigenständig das minimale Linkgewicht in dem Maße reduziert, bis eine bestimmte Mindestanzahl an Ergebnistermen erreicht ist. Dies ist jedoch bei in dieser Arbeit nicht der Fall.

Ein zu niedriges minimales Linkgewicht kann dagegen zur Folge haben, dass nicht einschlägige Terme in die Ergebnisliste aufgenommen werden. Dies kann teilweise unterbunden werden, indem die maximale Anzahl an Ergebnistermen verringert wird. Dadurch, dass nur eine bestimmte Anzahl an Termen der am stärksten aktivierten Knoten zurückgeliefert wird, werden die gering angeregten Knoten nicht als Ergebnis ausgegeben.

Einstellung	Wert
Minimales Linkgewicht	0,3
Maximale Anzahl an Ergebnisdokumenten	30
Maximale Anzahl an Ergebnistermen	10

Tabelle 4.3: Einstellungen der Suche nach „Il6 Il8“ im „CCL20“-Experiment.

Als Resultat wurden die zehn am stärksten aktivierten Knoten als verwandte Ergebnisterme zurückgeliefert sowie die Dokumente und *gene subgroup mining*-Experimente, in denen die Terme bzw. Gennamen vorkommen. Diese sind in Tabelle 4.4 aufgelistet.

Ergebnisterme
„211506_s_at >“, „cell-wall“, „interleukin 6 (interferon, beta 2)“, „IL6“, „IL8“ „205207_at >“, „interleukin 8“, „cw“, „bcg-cws“, „202859_x_at >“

Dokumente und *gene subgroup mining*-Experimente	Terme und Gennamen
Gene-inducing program of human dendritic cells in response to BCG cell-wall skeleton (CWS), which reflects adjuvancy required for tumor immunotherapy.	cell-wall, IL6, IL8, cw, bcg-cws
Effects of Salmonella enterica serovars Typhimurium (ST) and Choleraesuis (SC) on chemokine and cytokine expression in swine ileum and jejunal epithelial cells.	IL8
Addition of interleukin 1 (IL1) and IL17 soluble receptors to a tumour necrosis factor alpha soluble receptor more effectively reduces the production of IL6 and macrophage inhibitory protein-3alpha and increases that of collagen in an in vitro model of rheumatoid synoviocyte activation.	IL6
RAD Human U133A	211506_s_at >, 205207_at >, 202859_x_at >

Tabelle 4.4: Ergebnisterme, -dokumente und -*gene subgroup mining*-Experimente zur Suche nach „Il6 Il8“ im „CCL20“-Experiment.

Abbildung 4.1 zeigt den Teilgraph, bestehend aus Knoten und Links, welchen der

Branch-and-Bound-Algorithmus auf der Suche nach „Il6 Il8“ gefolgt ist und aktiviert hat. Dargestellt sind alle durch die Verbreitung der Aktivierung angeregten Knoten und deren Verbindungen zueinander. Termknoten werden durch Ellipsen repräsentiert, Genknoten durch Rechtecke. Die Knoten, nach denen gesucht wurde, sind grün gefärbt. Die Gene und Terme des Graphen mit den zehn höchsten Aktivierungswerten sowie die Dokumente und *gene subgroup mining*-Experimente, in denen diese auftreten, wurden als Resultat der Suche zurückgeliefert. Das Ergebnis zeigt die Synonyme der Gene „Il6“ und „Il8“, „interleucon 6“, „interleucin 8“ und deren Affymetrix-Ids „205270_at“, „211506_s_at“ und „202859_*x_at*“, die ebenfalls durch die rechteckige Darstellung als Genknoten zu erkennen sind. Außerdem ist zu sehen, dass weitere Verbindungen zu den Termen „bcg-cws“ und „bcg-cws-“ bestehen. Es existieren hier zwei Schreibweisen des gleichen Wortes, da Bindestriche nicht von Satzzeichenfilter und Stemmingverfahren entfernt wurden. Das Protein „BCG-CWS“ (*bacillus Calmette-Guerin cell wall skeleton*) wirkt regulierend auf IL Gene [KKTA+05]. Abbildung 4.1 zeigt, dass sowohl Links von beiden „bcg-cws“ Knoten zu den Termknoten „cell-wall“ und „skeleton“ führen als auch zu „up-regulated“.

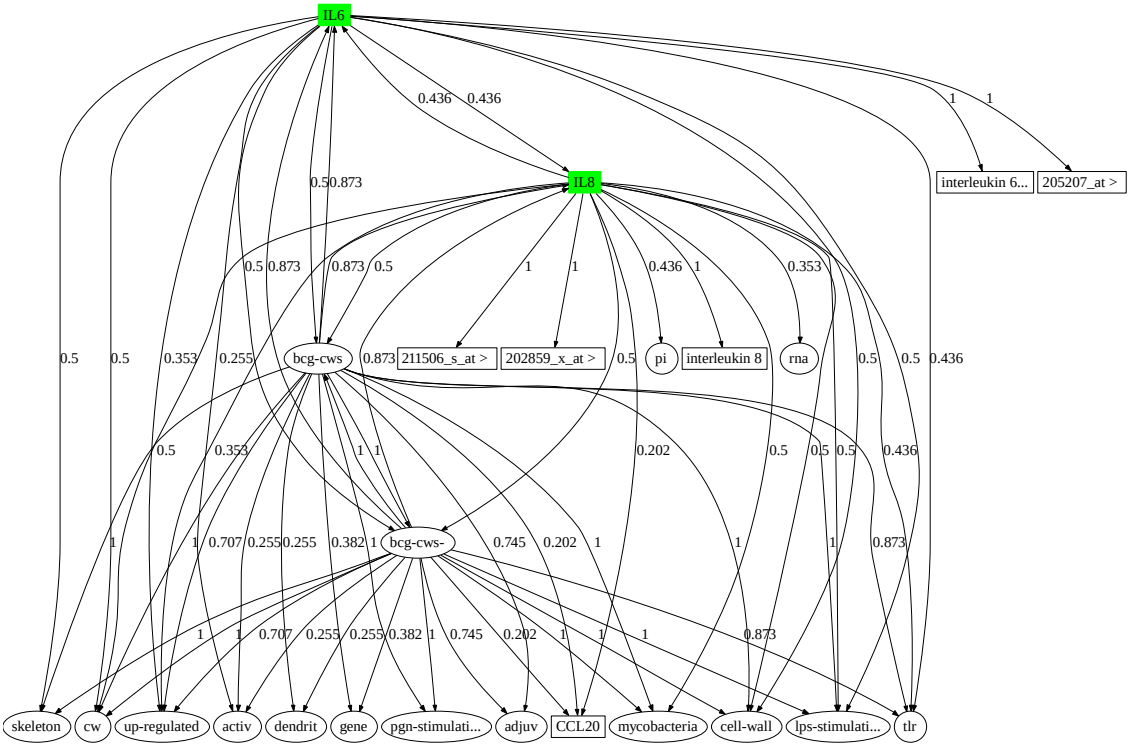


Abbildung 4.1: Der Teilgraph, welcher durch den Branch-and-Bound-Algorithmus, während der Suche nach „Il6 Il8“ im „CCL20“-Experiment aktiviert wurde.

Informationen beider Datenquellen wurden hier erfolgreich vernetzt. Mit einer Suche in den *gene subgroup mining*-Daten könnten nur die Synonyme, Affymetrix-Ids, Gengruppen und *gene subgroup mining*-Experimente bezüglich der Gene „Il6“ und „Il8„ gefunden werden. Im Textkorpus dagegen könnten nur Dokumente in Beziehung mit den Genen gefunden werden. Durch die assoziative Verknüpfung der Daten ist es möglich, Informationen aus beiden Datenquellen zu erhalten.

4.3.2 Mensch, Diabetes und Cluster-Experiment

Um Dokumente und *gene subgroup mining*-Experimente zu finden, die in Beziehung mit Diabetes und Mensch stehen, wurden in diesem Experiment 769 Zusammenfassungen der PubMed Datenbank mit den Suchtermen „human diabetes cluster“ als Textkorpus verwendet. Das Wort „cluster“ wurde zusätzlich als Suchterm verwendet, um die Ergebnismenge der Dokumente auf diese zu reduzieren, welche Gengruppen bzw. Cluster von Genen behandeln, die im Zusammenhang mit der menschlichen Diabetes stehen. Erneut wurden die menschlichen Genexpressionsdaten aus Abschnitt 4.3.1, die durch *gene subgroup mining* zu Gengruppen zusammengefasst wurden, als weitere Datenquelle verwendet.

Wiederum wurde zuerst ein assoziatives Netz aus den Genen erstellt, anschließend konnten 3600 verschiedene Terme aus dem Textkorpus extrahiert und als Knoten in das Netz eingefügt werden. Die verwendete minimale Termhäufigkeit betrug 2, die maximale Termdokumenthäufigkeit 200, eine minimale Termdokumenthäufigkeit wurde nicht verwendet. Das Netz bestand nach der Erstellung insgesamt aus 3946 Knoten und 294900 Links. In Tabelle 4.5 sind die Gennamen aufgeführt, welche in den Genexpressionsdaten und in den Dokumenten vorkommen. Um die Verknüpfung der Informationen dieser beiden Datenquellen zu analysieren, wurde auch hier nach den gemeinsam auftretenden Gennamen gesucht.

Genname	Häufigkeit
H19	1
SET	38
SAT	12

Tabelle 4.5: Gennamen, die sowohl in den Genexpressionsdaten als auch in den Dokumenten des Textkorpus „human diabetes cluster“ auftreten.

Die Vorverarbeitung in diesem Experiment dauerte 56, 7 Sekunden, das Einfügen der Knoten und Links in das Netz 11501, 2 Sekunden (191, 7 Minuten). Es ist deutlich zu sehen, dass die durchschnittliche Zeit zum Einfügen und Verknüpfen eines Knotens, hier mit 3, 19 Sekunden, im Vergleich zum Experiment aus Abschnitt 4.3.1 mit 0, 88 Sekunden, auf das 3, 62-fache angestiegen ist. Je mehr Knoten im Netz verlinkt sind,

desto länger dauert der Prozess. Das gesamte Netz benötigte 14240368 Bytes (ca. 13, 6 MB) Speicherplatz.

H19

Der erste Genname („H19“) der Tabelle 4.5 wurde als Suchterm verwendet und die Suche im assoziativen Netz mit den Einstellungen aus Tabelle 4.6 gestartet. Die Suche dauerte 0, 49 Sekunden.

Einstellung	Wert
Minimales Linkgewicht	0,7
Maximale Anzahl an Ergebnisdokumenten	30
Maximale Anzahl an Ergebnistermen	11

Tabelle 4.6: Einstellungen der Suche nach „H19“ im „human diabetes cluster“-Experiment.

Die elf am stärksten aktivierten Knoten wurden als verwandte Ergebnisterme zurückgeliefert sowie die Dokumente und *gene subgroup mining*-Experimente, in denen die Terme bzw. Gennamen vorkommen, welche in Tabelle 4.7 aufgelistet sind.

Ergebnisterme
„yolk“, „224997_x_at >“, „H19“, „in“, „imprint“, „sole“, „thymu“, „sac“, „H19, imprinted maternally expressed untranslated mRNA“, „igf2“, „monoallel“

Dokumente und *gene subgroup mining*-Experimente	Terme und Gennamen
Evidence that insulin is imprinted in the human yolk sac.	yolk, H19, in, imprint, sole, thymu, sac, igf2, monoallel
RAD Human U133A	224997_x_at >

Tabelle 4.7: Ergebnisterme, -dokumente und -*gene subgroup mining*-Experimente zur Suche nach „H19“ im „human diabetes cluster“-Experiment.

Der Teilgraph, der durch die Branch-and-Bound-Suche aktiviert wurde, ist in Abbildung 4.2 illustriert. Dargestellt sind erneut alle durch die Verbreitung der Aktivierung angeregten Knoten und deren Verbindungen untereinander. Der Knoten des Suchterms „H19“ ist wieder grün dargestellt und in der Abbildung ganz oben platziert. Er ist sowohl mit der Affymetrix-Id „224997_x_at“ und dem Synonym „H19, imprinted maternally expressed untranslated mRNA“ verknüpft als auch mit Genen, wie „igf2“ und „ins“, die in Zusammenhang mit Diabetes bzw. der Insulinproduktion stehen ([GAAB+01], [GBL+05], [VBC+96]). Das Gen „Ins“ wird durch den Termknoten

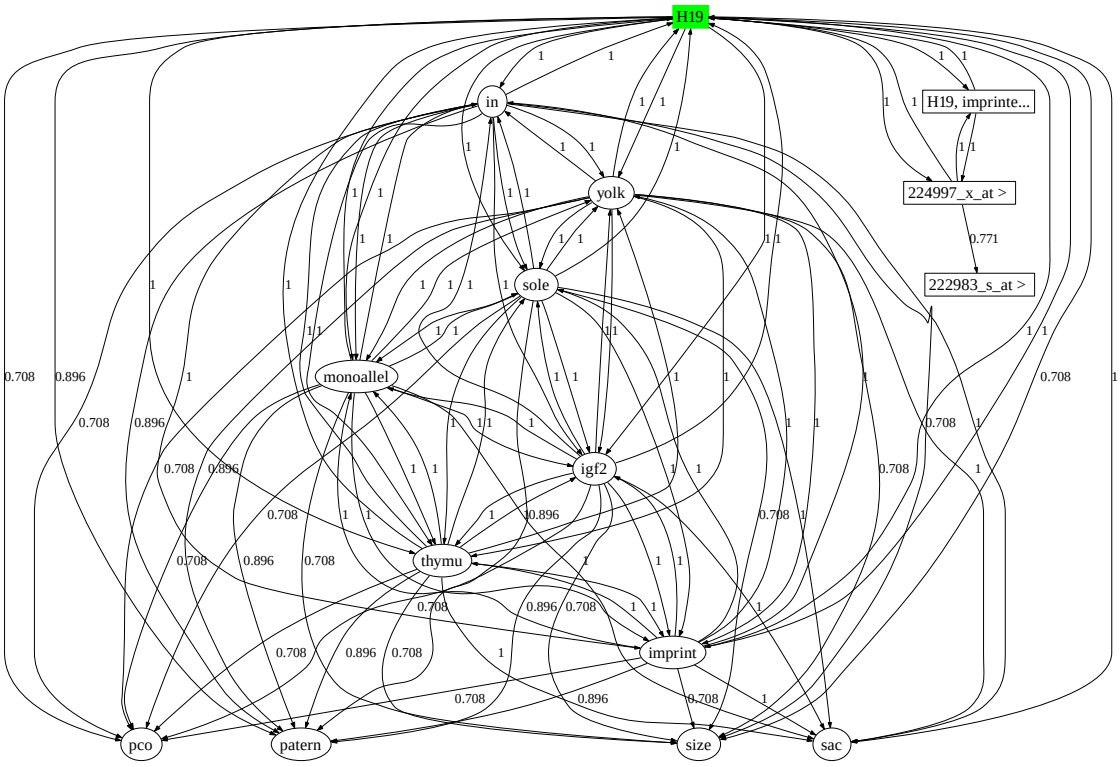


Abbildung 4.2: Der Teilgraph, welcher durch den Branch-and-Bound-Algorithmus, während der Suche nach „H19“ im „human diabetes cluster“-Experiment aktiviert wurde.

„in“ repräsentiert, der durch den Stemmingprozess den Buchstaben s einbüßen musste. Dass es sich hier um das Wort „in“ handeln könnte ist ausgeschlossen, da dies durch den Stopwortfilter entfernt wurde. Hier wird deutlich, dass eine Liste möglichst vieler Gennamen sinnvoll ist, um diese vom Stemmingprozess auszuschließen. Bisher werden nur die Gene ausgeschlossen, die als Genknoten zuvor in das Netz eingefügt wurden.

Weiter besteht von „H19“ und „in“ eine direkte Verbindung zum Termknoten „pco“, was für *polycystic ovary syndrome* (PCOS) steht. PCOS, auch bekannt unter dem Namen Stein-Leventhal-Syndrom, ist eine der häufigsten Stoffwechselstörungen bei geschlechtsreifen Frauen und kann zu Unfruchtbarkeit und Diabetes vom Typ 2 führen ([Hoe03], [SN00], [GAAB+01]). Außer den Synonymen und den verwandten Termen existiert von „H19“ über die Affymetrix-Id „224997_x_at“ eine Verbindung zum Gen „222983_s_at“, das in der gleichen Gengruppe ist.

Zum Term „thymu“, was für Thymus steht, existiert ebenfalls ein Link ausgehend von „H19“. Der Thymus ist ein Organ unseres Lymphsystems und sehr wichtig für

das Immunsystem. Er liefert die Umgebung für die Entwicklung und Reifung der T-Abwehrzellen. Diese Umgebung hilft dabei, selbstreaktive Zellen zu eliminieren oder zu inaktivieren, deren Zerstörungskraft andernfalls gegen das gesunde Körpergewebe verwendet werden würde. Autoimmunerkrankungen wie Diabetes, Rheuma, Multiple Sklerose etc. gehen auf die fehlende Fähigkeit zurück, zwischen körperfremden und körpereigenen Stoffen unterscheiden zu können ([BCR+06], [Boe06], [GBL+05]).

Hier wurde also neben den Synonymen zum einen ein Gen gefunden, was mit „H19“ in derselben Gengruppe ist, zum anderen wurden Gene entdeckt, die mit H19 verknüpft sind und mit Diabetes in Zusammenhang stehen. Biologen und Mediziner können nun genauer untersuchen ob die Gene „Igf2“ und „Ins“ evtl. auch in dieselbe Gengruppe wie „H19“ und „222983_s_at“ einzuordnen sind. Es wurde außerdem durch die Verbindung zu den Termknoten „pco“ und „thymu“ ein weiterer Zusammenhang zu Diabetes hergestellt.

SAT

In einer weiteren Anfrage mit den Einstellungen aus Tabelle 4.8 wurde als Suchterm der Enzymname „SAT“ (*spermidine/spermine N1-acetyltransferase*) verwendet. Die Suche dauerte 0,43 Sekunden. SAT tritt ebenfalls sowohl in den Dokumenten auf als auch in den im *gene subgroup mining* gefundenen Gengruppen.

Einstellung	Wert
Minimales Linkgewicht	0,7
Maximale Anzahl an Ergebnisdokumenten	30
Maximale Anzahl an Ergebnistermen	10

Tabelle 4.8: Einstellungen der Suche nach „SAT“ im „human diabetes cluster“-Experiment.

Als Ergebnisknoten wurden die zehn am stärksten aktivierten Knoten ausgegeben. Diese, die Dokumente und *gene subgroup mining*-Experimente sind in Tabelle 4.9 aufgelistet.

Bei der Suche nach „SAT“ tritt ein Problem mit der Ambiguität von Termen im assoziativen Netz auf. Jeder Term wird im Netz nur durch einen Knoten abgebildet, kann jedoch unterschiedliche Bedeutungen haben, z.B. kann das Wort „Jaguar“ je nach Zusammenhang ein Tier oder ein Auto sein. Dies hat zur Folge, dass Knoten mehrdeutiger Terme Verknüpfungen zu Termen oder Gennamen aus unterschiedlichen Themenwelten haben. Werden diese als Ergebnis zurückgeliefert, so muss der Benutzer selbst entscheiden, ob zwischen diesen ein Zusammenhang besteht oder nicht. Eine Möglichkeit, unerwünschte Terme bzw. Teilgraphen an Termen auszublenden, ist die Verwendung von NOT-Termen in einer weiteren Suchanfrage.

Ergebnisterme
„213988_s_at >“, „subcutan“, „lean“, „android“, „SAT“, „203455_s_at >“, „spermidine/spermine N1-acetyltransferase“, „sat-top“, „leg“, „210592_s_at >“

Dokumente und *gene subgroup mining*-Experimente	Terme und Gennamen
Subcutaneous adipose tissue pattern in lean and obese women with polycystic ovary syndrome.	subcutan, lean, android, SAT, sat-top, leg
Android subcutaneous adipose tissue topography in lean and obese women suffering from PCOS: comparison with type 2 diabetic women.	subcutan, lean, android, SAT, sat-top, leg
Human epicardial adipose tissue is a source of inflammatory mediators.	subcutan
Clustering of dyslipidemia, hyperuricemia, diabetes, and hypertension and its association with fasting insulin and central and overall obesity in a general population. Atherosclerosis Risk in Communities Study Investigators.	lean
RAD Human U133A	213988_s_at >, 203455_s_at >, 210592_s_at >

Tabelle 4.9: Ergebnisterme, -dokumente und -*gene subgroup mining*-Experimente zur Suche nach „SAT“ im „human diabetes cluster“-Experiment.

„SAT“ bezeichnet zum einen das Enzym *spermidine/spermine N1-acetyltransferase*[2] und zum anderen die Gewebeart *subcutaneous adipose tissue*. In Abbildung 4.3, die den durch die Suche aktivierten Teilgraph darstellt, ist zu sehen, dass vom Knoten „SAT“ (grün eingefärbt) Verbindungen zu dessen Affymetrix-Id „213988_s_at >“, Synonym „spermidine/spermine N1-acetyltransferase“ und zu Gengruppen bestehen. In diesem Zusammenhang tritt „SAT“ als Gen- bzw. Enzymname auf. Weiter existieren zusätzlich Links zu den Termen „subcutan“, „leg“, „lean“, „pco“ und „sat-top“ (*SAT topography*). Hier steht „SAT“ für die Gewebeart *subcutaneous adipose tissue* ([TMR+03], [HMR+04]). Die Gene der Gengruppen, in welchen sich „SAT“ befindet, haben demzufolge nichts mit den Termen, die in Bezug zur Gewebeart stehen, zu tun.

4.3.3 Mensch und Diabetes-Experiment

Um weitere Zusammenhänge zum Thema Mensch und Diabetes zu finden, wurde der Datenbestand an wissenschaftlichen Artikeln vergrößert. Aus der PubMed Datenbank wurden 99552 Zusammenfassungen zum Thema „human diabetes“ als Textkorpus

[2] Url der Beschreibung des Gens SAT:
www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene&cmd=Retrieve&dopt=full_report&list_uids=6303

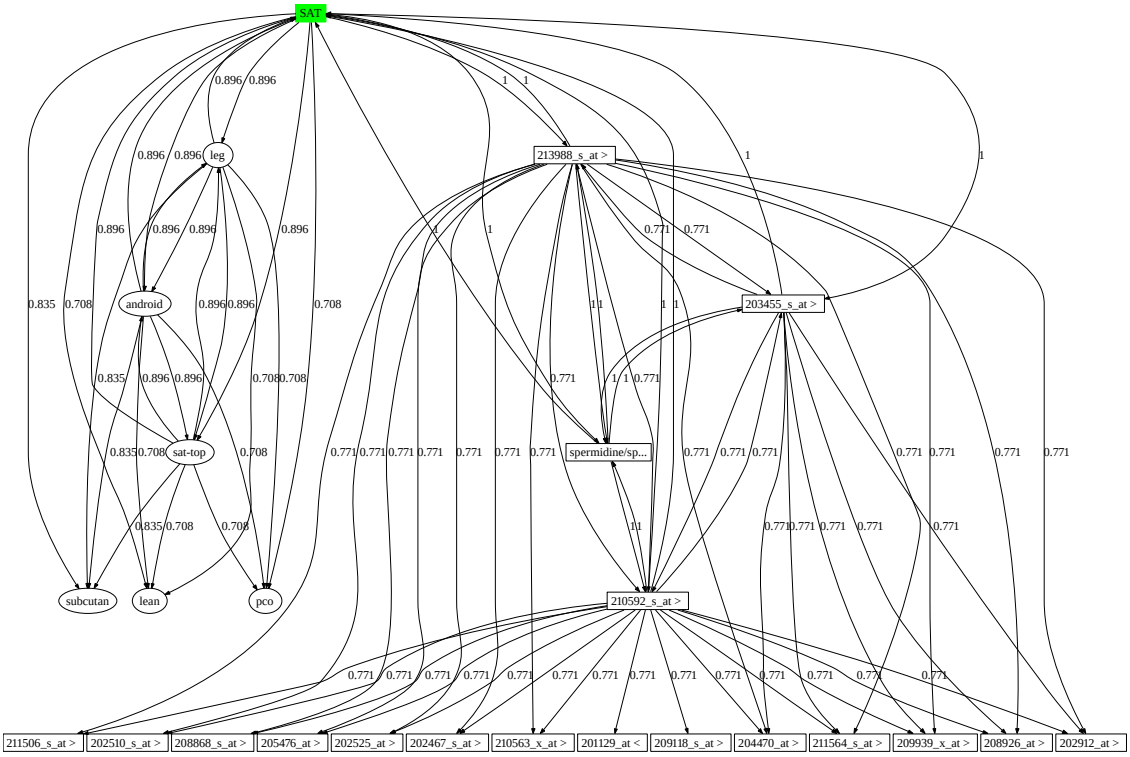


Abbildung 4.3: Der Teilgraph, welcher durch den Branch-and-Bound-Algorithmus, während der Suche nach „SAT“ im „human diabetes cluster“-Experiment aktiviert wurde.

verwendet. Die Datenbanksuche wurde ohne den Anfrageterm „cluster“ durchgeführt, wodurch sich die Anzahl der Ergebnisdokumente von 769 auf 99552 drastisch steigerte. Als zweite Datenquelle wurden dieselben, durch *gene subgroup mining* gruppierten, menschlichen Genexpressionsdaten wie in Abschnitt 4.3.1 benutzt.

Aus dem Textkorpus konnten 11371 verschiedene Terme extrahiert werden. Dabei wurde eine minimale Termhäufigkeit von 3 und eine minimale Termdokumenthäufigkeit von 3 benutzt. Die Verwendung dieser minimalen Termdokumenthäufigkeit hatte zur Folge, dass ein großer Teil der Wörter des Textkorpus nicht als Terme in das Netz eingefügt wurde. Ohne die Verwendung der minimalen Termdokumenthäufigkeit wären 35310 Terme extrahiert worden. Diese Menge an einzufügenden Knoten bzw. die Berechnung der Links zwischen den Knoten, deren Anzahl um ein Vielfaches größer ist, hätte sehr viel Rechenzeit beansprucht. Deshalb wurde die Anzahl der Terme auf etwa ein Drittel reduziert. Insgesamt bestand das Netz aus 11707 Knoten und 934768 Links. Tabelle 4.10 führt alle Gennamen auf, die sowohl in Dokumenten als auch in den Genexpressionsdaten auftreten. Es ist zu erkennen, dass sich deren Anzahl von 24 im

Vergleich zu der Anzahl aus Abschnitt 4.3.2 von 3, um das 8-fache erhöht hat, was auf den umfangreicheren Textkorpus zurückzuführen ist.

Genname	Häufigkeit
IL8	2
RAB1A	1
IL6	31
VCAM1	7
STAT3	14
CFLAR	2
calreticulin	5
SAT	44
RAB18	2
ENSA	21
H19	7
AKR1B1	8
GAPDH	69
adrenomedullin	123
ADM	37
TIMP1	1
APP	27
CROP	2
CYP3A5	6
BMP2	4
SET	1487
calnexin	4
SERPINA1	1
SCOC	7

Tabelle 4.10: Gennamen, die sowohl in den Genexpressionsdaten als auch in den Dokumenten des Textkorpus „human diabetes“ auftreten.

Insgesamt wurden 267792, 2 Sekunden (ca. 74, 39 Stunden) benötigt, um die Vorverarbeitung und die Netzerstellung abzuschließen. Die Vorverarbeitung alleine konnte in 5563, 1 Sekunden (ca. 91, 72 Minuten) abgeschlossen werden. Das Einfügen der Termknoten und dessen Links benötigte 262229, 1 Sekunden (ca. 72, 84 Stunden). Die durchschnittliche Zeit einen Termknoten in das Netz einzufügen und zu verlinken betrug in diesem Experiment 23, 06 Sekunden. Das ist etwa 7, 2 mal soviel wie in dem Experiment aus Abschnitt 4.3.2, mit 3, 19 Sekunden. Die Anzahl der Knoten insgesamt ist in diesem Experiment, mit 11707, 2, 97 mal so groß und die Anzahl der Links, mit 934768, 3, 17 mal so groß wie in Abschnitt 4.3.2. Es ist zu sehen, dass die durchschnittliche Zeit einen Knoten in das Netz einzufügen und mit anderen Knoten zu verknüpfen drastisch steigt, je mehr Knoten und Links existieren. Hier besteht ebenfalls Optimierungsbedarf bezüglich

des Algorithmusses zum Einfügen und Verknüpfen der Knoten und der Datenstruktur, welche das Netz repräsentiert. Das Netz benötigte insgesamt 61258858 Bytes (ca. 58, 42 MB) an Speicherplatz.

Calreticulin

Bei der Suche nach dem Gennamen „calreticulin“, der in Dokumenten und Genexpressionsdaten auftritt, wurde ein weiteres Synonym für dieses Gen gefunden, welches in den Genexpressionsdaten noch nicht vorhanden war. Die Einstellungen, mit denen die Suche im assoziativen Netz durchgeführt wurde, sind in Tabelle 4.11 zu sehen. Die Suche dauerte 0, 62 Sekunden.

Einstellung	Wert
Minimales Linkgewicht	0,1
Maximale Anzahl an Ergebnisdokumenten	30
Maximale Anzahl an Ergebnistermen	5

Tabelle 4.11: Einstellungen der Suche nach „calreticulin“ im „human diabetes“-Experiment.

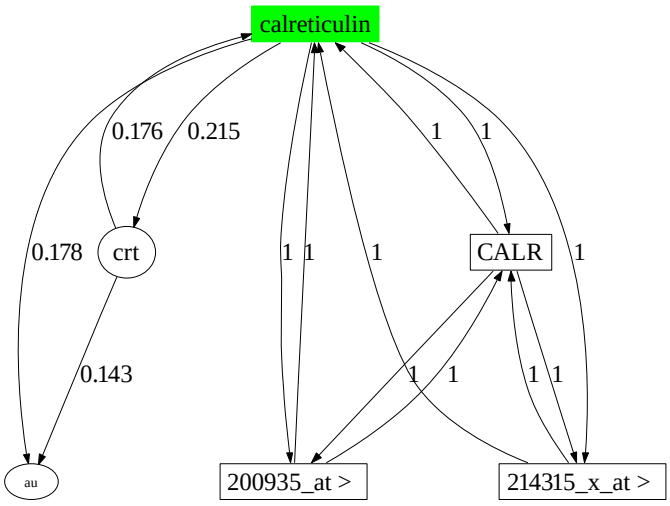


Abbildung 4.4: Der Teilgraph, welcher durch den Branch-and-Bound-Algorithmus, während der Suche nach „calreticulin“ im „human diabetes“-Experiment erstellt wurde.

Die Ergebnisdokumente und *gene subgroup mining*-Experimente, sowie die 5 am stärksten aktivierten Knoten, die als verwandte Terme ausgegeben wurden, sind in

Ergebnisterme
„200935_at >“, „214315_x_at >“, „calreticulin“, „CALR“, „crt“

Dokumente und *gene subgroup mining*-Experimente	Terme und Gennamen
Occurrence of IgA and IgG autoantibodies to calreticulin in coeliac disease and various autoimmune diseases.	calreticulin, crt
Valproate protects cells from ER stress-induced lipid accumulation and apoptosis by inhibiting glycogen synthase kinase-3.	calreticulin
Kidney allograft and patient survival in type I diabetic recipients of cadaveric kidney alone versus simultaneous pancreas kidney transplants: a multivariate analysis of the UNOS database.	crt
Phase II trial of conformal radiation therapy for pediatric patients with craniopharyngioma and correlation of surgical factors and radiation dosimetry with change in cognitive function.	crt
Effect of video display on the grading of diabetic retinopathy.	crt
RAD Human U133A	200935_at >, 214315_x_at >

Tabelle 4.12: Ergebnisterme, -dokumente und -*gene subgroup mining*-Experimente zur Suche nach „calreticulin“ im „human diabetes“-Experiment.

Tabelle 4.12 aufgelistet.

Abbildung 4.4 zeigt den Teilgraph, der durch die Suche nach „calreticulin“ erstellt wurde. Es sind sowohl die Synonyme „CALR“, „200935_at >“ und „214315_x_at >“, aus den Genexpressionsdaten als Knoten auszumachen als auch ein weiteres Synonym namens „crt“ aus dem Textkorpus. „Crt“ ist jedoch nur ein Synonym für das Mäusegen „Calr“[3], nicht für das menschliche Gen.

Durch die Verknüpfung der Daten beider Datenquellen konnten erneut Informationen sinnvoll kombiniert werden. Die Synonymgruppe des Gens „calraticulin“, aus den Genexpressionsdaten, wurde durch ein weiteres Synonym „crt“, aus den Dokumenten, ergänzt.

[3]Url der Beschreibung des Gens Calr bzw. Crt:
www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene&cmd=Retrieve&dopt=full_report&list_uids=12317

H19 Diabesity

Der Genname H19, der in Abschnitt 4.3.2 schon in Zusammenhang mit Diabetes aufgefallen ist, wurde hier gemeinsam mit dem Term „diabesity“ als Suchterm verwendet. Die Suche im assoziativen Netz dauerte 0,56 Sekunden und wurde mit den Einstellungen, welche in Tabelle 4.13 aufgelistet sind, durchgeführt.

EINSTELLUNG	WERT
Minimales Linkgewicht	0,1
Maximale Anzahl an Ergebnisdokumenten	30
Maximale Anzahl an Ergebnistermen	20

Tabelle 4.13: Einstellungen der Suche nach „H19 diabesity“ im „human diabetes“-Experiment.

Die 20 am stärksten aktivierten Knoten, die als Ergebnisterme zurückgeliefert wurden sowie ein Teil der Dokumente und *gene subgroup mining*-Experimente, in denen die Terme bzw. Gennamen vorkommen, sind in Tabelle 4.14 aufgeführt. Das Wort „diabesity“ wurde durch den Stemmingprozess in „diabes“ umgewandelt.

Abbildung 4.5 zeigt, dass eine indirekte Verbindung vom Gennamen „H19“ über die Termknoten „icr“ und „epigenet“ zu „diabes“ besteht. Die Anfrageterme sind wieder grün eingefärbt. Außerdem ist zu sehen, dass die Synonyme des Gens und ein weiteres Gen „222983_s_at >“ angezeigt werden, welches sich mit „H19“ in einer Gengruppe befindet. Zusätzlich bestehen direkte Verbindungen zu den Termen „vntr“ und „in“ und indirekte Links zu „igf2“ und „iddm2“. Die Gene „Igf2“ und „in“ (Ins) wurden auch schon in Abschnitt 4.3.2 in Zusammenhang mit „H19“ und Diabetes gefunden. Überdies wurden, durch die Verwendung des umfangreicheren Textkorpus, die Terme „iddm2“ und „vntr“ entdeckt. Das Gen „Iddm2“ (*insulin-dependent diabetes mellitus 2*) steht wie auch „Igf2“ und „Ins“, in Verbindung mit der Insulinregulierung des Körpers ([VBC+96], [PZF+97]). „Vntr“ (*variable number of tandem repeats*) sind tandemartige Wiederholungen von DNA-Sequenzen, wobei die Anzahl der wiederholten Sequenzen sehr variabel ist. Dies hat zur Folge, dass viele Menschen einer Population an demselben Genlocus[4] heterozygot[5] sind. Die Sequenzen und die Anzahl der Wiederholungen eignen sich daher dafür, verschiedene Individuen voneinander zu differenzieren ([Hen97]). Das Auftreten solcher Sequenzen in bestimmten Wiederholungen an bestimmten Genloci kann sich auf die Insulinproduktion auswirken und somit in Bezug zu Diabetes stehen ([PZF+97], [VBC+96], [OD04]). Neben diesen sind die Genknoten „pparg“ (*peroxisome proliferative activated receptor, gamma*) und „tndm“ (*diabetes mellitus, transient neonatal*) im Teilgraph der Abbildung 4.5 zu erkennen. Sie sind ebenfalls indirekt mit

[4]Die physikalische Position einer DNA-Sequenz im Genom wird als Genlocus bezeichnet.
[5]Die Mischergibigkeit bezüglich eines genetischen Merkmals wird Heterozygotie genannt.

Ergebnisterme
„224997_x_at >“, „sac“, „H19“, „H19, imprinted maternally expressed untranslated mRNA“, „patern“, „vntr“, „yolk“, „in“, „pparg“, „diabes“, „bw“, „postnat“, „icr“, „methylat“, „imprint“, „childhood“, „size“, „epigenet“, „acdc“, „tk“

Dokumente und *gene subgroup mining*-Experimente	Terme und Gennamen
Birth weight, infant growth and insulin resistance.	H19, vntr, in, postnat, childhood, size
Epigenetic alterations of H19 and LIT1 distinguish patients with Beckwith-Wiedemann syndrome with cancer and birth defects.	H19, bw, methylat, epigenet
Structural-tridimensional study of yolk sac in pregnancies complicated by diabetes.	sac, yolk
Imprinted and genotype-specific expression of genes at the IDDM2 locus in pancreas and leucocytes.	vntr, in, imprint
The insulin gene is transcribed in the human thymus and transcription levels correlated with allelic variation at the INS VNTR-IDDM2 susceptibility locus for type 1 diabetes.	vntr, in
RAD Human U133A	224997_x_at >

Tabelle 4.14: Ergebnisterme, -dokumente und -*gene subgroup mining*-Experimente zur Suche nach „H19 diabesity“ im „human diabetes“-Experiment.

„H19“ verknüpft und stehen auch in Beziehung mit Diabetes ([VHL+05], [DWF06], [MBCS+06]).

Auch hier wurde zum einen ein Gen gefunden, welches sich mit „H19“ in derselben Gruppe befindet. Zusätzlich wurden sechs weitere Gennamen als Ergebnis zurückgeliefert, die in enger Beziehung zu „H19“ und Diabetes stehen. Biologen und Mediziner können feststellen, ob die Gengruppe um die sechs zusätzlich gefundenen Gene erweitert werden muss oder ob die vier Gene, die Einfluß auf die Insulinregulierung des Körpers haben, in eine eigene Gruppe einzuordnen sind. Obwohl die Anzahl der Terme durch die hohe minimale Termdokumenthäufigkeit von 10 stark reduziert wurde, konnte doch ein Zusammenhang zwischen dem Gen „H19“ und Diabetes gefunden werden. Die Vermutung liegt nahe, dass eine Erhöhung der Termanzahl, durch die Senkung der minimalen Termdokumenthäufigkeit und Termhäufigkeit, weitere Zusammenhänge zu Tage bringen könnte.

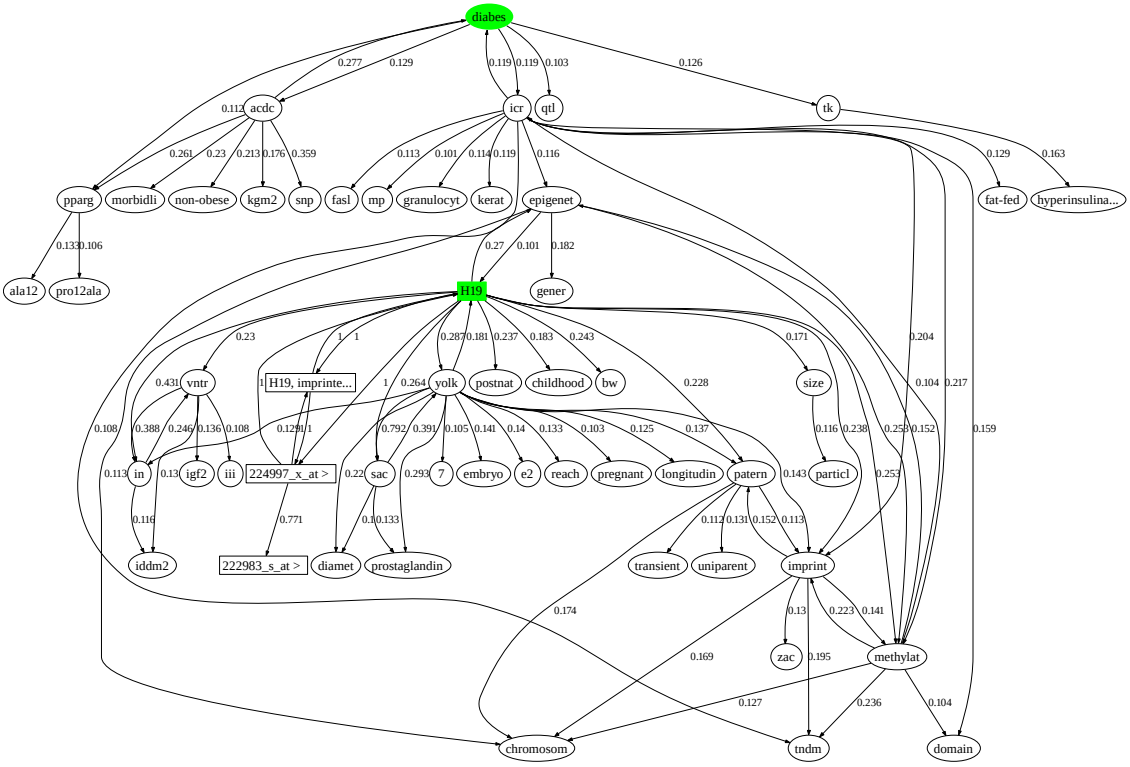


Abbildung 4.5: Der Teilgraph, welcher durch den Branch-and-Bound-Algorithmus, während der Suche nach „H19 diabesity“ im „human diabetes“-Experiment erstellt wurde.

Kapitel 5

Fazit und Ausblick

In dieser Arbeit wurde gezeigt, wie Informationen heterogener Datenräume durch ein assoziatives Netz gemeinsam verknüpft und analysiert werden können. Als Datenquellen wurden Textdokumente, bestehend aus Zusammenfassungen wissenschaftlicher Artikel über Themen der Biologie und der Medizin, aus der PubMed Datenbank und Genexpressionsdaten menschlicher Gene verwendet.

Die experimentellen Ergebnisse zeigen, dass die assoziative Verbindung der Informationen der Datenquellen und deren Exploration durch das Netz möglich ist. So wurde zu einem Gen und dessen Synonymen ein Term gefunden, der ein weiteres Synonym des Gens ist. Es wurden Gene bzw. Gennamen in Dokumenten gefunden, die evtl. eine durch *gene subgroup mining* erstellte Gengruppe erweitern oder selbst eine Gengruppe bilden. Wie genau der Zusammenhang zwischen den gefundenen Genen zu beurteilen ist, müssen Biologen oder Mediziner entscheiden. Es ist jedoch anzunehmen, dass ein Zusammenhang existiert. Dadurch zeigt sich, dass die Informationen beider Datenquellen sinnvoll kombiniert und ausgewertet werden können. Weiter wurde ein Bezug zwischen einem genregulierenden Protein und Genen der Genexpressionsdaten ausgemacht sowie zwischen bestimmten Genen und der Krankheit Diabetes.

Es darf weiterhin auch nicht vergessen werden, dass das in dieser Arbeit erstellte assoziative Netz ein Prototyp ist, um heterogene Datenquellen zu analysieren. Um qualitativ und quantitativ bessere Ergebnisse zu bekommen sind, verschiedene Änderungen und Verfeinerungen denkbar. In der Vorverarbeitung werden zuerst alle Texte tokenisiert. Bisher werden in dieser Arbeit die Dokumente auf Wortebene segmentiert, d.h. Gennamen oder Namen aus Genontologien, die aus mehreren Wörtern bestehen, werden nicht als solche erkannt, sondern Wort für Wort in das Netz eingefügt. Die Erkennung von Namen und Bezeichnungen, die aus mehreren Wörtern bestehen, kann die Qualität der Informationen, welche in das Netz integriert werden, deutlich verbessern. Gerade im Hinblick auf die Verwendung von Genontologien als weitere Datenquelle, ist dies unerlässlich, da die Bezeichnungen der Ontologieeinträge meist aus

mehreren Wörtern bestehen.

Ein weiterer Schritt der Vorverarbeitung, den die Terme bis auf die Gennamen der Genexpressionsdaten, welche bereits in das Netz eingefügt wurden, durchlaufen, ist der Stemmingprozess. Das heißt, dass auch die Namen der Gene, die zuvor nicht ins Netz eingefügt wurden, gestemmt und somit zu teilweise nicht existierenden Gennamen umgeformt werden. Hier kann eine Datenbank mit möglichst vielen Gennamen Abhilfe schaffen, in welcher vor dem Stemmingprozess überprüft wird, ob der zu stemmende Term der Name eines Gens ist.

Außerdem ist die Methode der Termextraktion zu verbessern. So könnten z.B. zusätzlich Struktur- oder semantische Informationen der Texte genutzt werden, um die Einschlägigkeit ihrer Terme festzustellen. Diese Informationen können außerdem mit in die Gewichtsbestimmung der Links zwischen den Termen einfließen. Ein weiteres Problem tritt mit der Ambiguität von Termen auf. Für jeden Term wird nur ein Knoten in das Netz eingefügt. Tritt ein Wort aufgrund seiner verschiedenen Bedeutungen in den Dokumenten in unterschiedlichen Zusammenhängen auf, so werden diese im Netz durch Verbindungen zu Termen dieser Zusammenhänge repräsentiert. Daraus folgt, dass bei einer Suche nach einem mehrdeutigem Term diese Knoten aktiviert werden und somit Terme aus unterschiedlichen Zusammenhängen als verwandte Terme resultieren.

Die in dieser Arbeit verwendeten Textkorpora bestehen aus Zusammenfassungen wissenschaftlicher Artikel. Würden die kompletten Artikel, die mehr Informationen enthalten als deren Zusammenfassungen, als Textkorpus genutzt werden, könnten demnach auch mehr Informationen in das Netz einfließen. Die Erstellung eines Textkorpus unter Verwendung von kompletten Artikeln ist ein weiterer Punkt, der dazu beiträgt, qualitativ bessere Ergebnisse zu erhalten. Aus einem großen Textkorpus können jedoch auch viele Terme extrahiert werden. Mit der in dieser Arbeit verwendeten Datenstruktur, zur Speicherung des Netzes und dem Algorithmus zum Einfügen der Knoten und Links, würde die Erstellung eines Netzes mit sehr vielen Termen bzw. Genen äußerst viel Zeit benötigen, wie die Experimente deutlich machen. Es ist deshalb notwendig, die Datenstruktur und den Algorithmus soweit zu verbessern, dass ein großes Netz aus einigen 10000 Knoten in moderater Zeit erstellt werden kann.

Alles in allem wurden die Daten der verwendeten heterogenen Datenquellen durch den Prototyp des assoziativen Netzes sinnvoll verbunden und konnten, wie die Ergebnisse der Experimente zeigen, erfolgreich exploriert und analysiert werden.

Anhang A

XML-DTDs

A.1 Anfrage-XML-DTD

Listing A.1: Anfrage-XML-DTD

```
<!ELEMENT Query (MaxNoRelatedTerms, MaxNoDocuments, MinimumLinkWeight,
Strict, WeightPolicy, Terms, ExcludedTerms)>

<!ELEMENT MaxNoRelatedTerms (#PCDATA) >
<!ELEMENT MaxNoDocuments    (#PCDATA) >
<!ELEMENT MinimumLinkWeight (#PCDATA) >
<!ELEMENT Strict            (#PCDATA) >
<!ELEMENT WeightPolicy      (#PCDATA) >
<!ELEMENT Terms             (Term)* >
<!ELEMENT ExcludedTerms     (Term)* >
<!ELEMENT Term              (#PCDATA) >
```

A.2 Antwort-XML-DTD

Listing A.2: Antwort-XML-DTD

```
<!ELEMENT ResultSet  (RelatedTerms*,Document*,GeneExpressionSubgroup*)>

<!ELEMENT RelatedTerms   (Term)* >
<!ELEMENT Term           (#PCDATA) >
<!ATTLIST Term           weight CDATA #IMPLIED >

<!ELEMENT Document (Rank, Title, Authors, File, ContainedTerms) >
<!ELEMENT Rank           (#PCDATA) >
<!ELEMENT Title          (#PCDATA) >
<!ELEMENT File           (#PCDATA) >
```

```
11
12 <!ELEMENT Authors          (Author)* >
13 <!ELEMENT Author           (Firtsname,Lastname) >
14 <!ELEMENT Firstname        (#PCDATA) >
15 <!ELEMENT Lastname         (#PCDATA) >
16
17 <!ELEMENT ContainedTerms (Term)* >
18
19 <!ELEMENT GeneExpressionSubgroup (Organism, Platform, ThresholdType,
20 UnderExpressionValue, OverExpressionValue, Support, Remarks) >
21
22 <!ELEMENT Organism                (#PCDATA) >
23 <!ELEMENT Platform                (#PCDATA) >
24 <!ELEMENT ThresholdType          (#PCDATA) >
25 <!ELEMENT UnderExpressionValue (#PCDATA) >
26 <!ELEMENT OverExpressionValue   (#PCDATA) >
27 <!ELEMENT Support               (#PCDATA) >
28 <!ELEMENT Remarks               (#PCDATA) >
```

Literaturverzeichnis

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[BCR+06] Conrad C. Bleul, Tatiana Corbeaux, Alexander Reuter, Paul Fisch, Joachim Schulte-Mönting, and Thomas Boehm. Formation of a functional thymus initiated by a postnatal epithelial progenitor cell. *Nature*, 33:106–119, 2006.

[Bel86] R K Belew. *Adaptive information retrieval: machine learning in associative networks*. PhD thesis, Univ. Michigan, CS Department, Ann Arbor, MI, 1986.

[Bel89] R. K. Belew. 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*, pages 11–20, New York, NY, USA, 1989. ACM Press.

[Bel00] Richard K. Belew, editor. *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambrigde University Press, 1 edition, 2000.

[Boe06] T. Boehm. Quality control strategies in self/non-self discrimination systems. *Cell*, 125:845–858, 2006.

[CBN95] H. Chen, K. Basu, and T. Ng. An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): Symbolic branch-and-bound search vs. connectionist hopfield net activiation. *Journal of the American Society for Information Science*, 46(5):348–369, 1995.

[Che95] Hsinchun Chen. Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society of Information Science*, 46(3):194–216, 1995.

[CHL+97] S. Cunningham, G. Holmes, J. Littin, R. Beale, and I. Witten. Applying connectionist models to information retrieval. In *S. Amari, and N. Kasobov (eds.) Brain-Like Computing and Intelligent Information Systems*, pages 435–457. Springer-Verlag, 1997.

[CPS98] Yi-Ming Chung, William M. Pottenger, and Bruce R. Schatz. Automatic subject indexing using an associative neural network. In *ACM DL*, pages 59–68. ACM Press, 1998.

[DD] J. Dalton and A. Deshmane. Artificial neural networks. *IEEE Potentials*, 10.

[Dil06] Fabian Dill. Subgroup mining of heterogeneous gene expression data. Master's thesis, University of Konstanz, 2006.

[DWF06] K. Delaval, A. Wagschal, and R. Feil. Epigenetic deregulation of imprinting in congenital diseases of aberrant growth. *BioEssays : news and reviews in molecular, cellular and developmental biology*, 28(5), 2006.

[Fer03] Reginald Ferber, editor. *Information Retrieval Suchmodelle und Data-Mining Verfahren für Textdammlungen und das Web*. dpunkt.verlag, 1 edition, 2003.

[GAAB+01] GE. Moore GE, SN. Abu-Amero, G. Bell, EL. Wakeling, A. Kingsnorth, P. Stanier, E. Jauniaux, and ST. Bennett. Evidence that insulin is imprinted in the human yolk sac. *Diabetes*, 50(1):199–203, 2001.

[GBL+05] V. Geenen, F. Brilot, C. Louis, I. Hansenne, Ch. Renard, and H. Martens. Importance of a thymus dysfunction in the pathophysiology of type 1 diabetes. *Revue médicale de Liège*, 60(5-6), 2005.

[Hau98] Matthias Haun, editor. *Simulation Neuronaler Netze*. expert Verlag, 1 edition, 1998.

[Heb49] D.O. Hebb. *The organization of behaviour*. John Wiley & Sons, 1949.

[Hen97] S. Henikoff. *Encyclopedia of Life Sciences*. Nature Publishing Press, 1997.

[HMR+04] R. Horejsi, R. Moller, S. Rackl, A. Giuliani, U. Freytag, K. Crailsheim, K. Sudi, and E. Tafeit. Android subcutaneous adipose tissue topography in lean and obese women suffering from pcos: comparison with type 2 diabetic women. *American journal of physical anthropology*, 124(3), 2004.

[Hoe03] KM. Hoeger. Role of lifestyle modification in the management of polycystic ovary syndrome. *Baillière's best practice & research. Clinical endocrinology & metabolism*, 2003.

[Jen01] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer-Verlag, New York, 2001.

[KK88] K. L. Kwok and William Kuan. Experiments with document components for indexing and retrieval. *Inf. Process. Manage.*, 24(4):405–417, 1988.

[KKTA+05] K. Ishii K, M. Kurita-Taniguchi, M. Aoki, T. Kimura, Y. Kashiwazaki, M. Matsumoto, and T. Seya. Gene-inducing program of human dendritic cells in response to bcg cell-wall skeleton (cws), which reflects adjuvancy required for tumor immunotherapy. *Immunology letters*, 2005.

[Kni90] Kevin Knight. Connectionist ideas and algorithms. *Commun. ACM*, 33(11):58–74, 1990.

[Koh89] T. Kohonen. *Self-organizing and associative memory*. Springer-Verlag, 1989.

[Koh95] T. Kohonen. *Self-organizing Map*. Springer-Verlag, 1995.

[Kor97] Robert R. Korfhage, editor. *Information Storage and Retrieval*. Wiley Computer Publishing, 1 edition, 1997.

[Kuh77] Rainer Kuhlen. *Experimentelle Morphologie in der Informationswissenschaft*. Verlag: Dokumentation, München, 1977.

[Kwo85] K L Kwok. A probabilistic theory of indexing and similarity measure based on cited and citing documents. *J. Am. Soc. Inf. Sci.*, 36(5):342–351, 1985.

[Kwo86] K. L. Kwok. An interpretation of index term weighting schemes based on document components. In *SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–283, New York, NY, USA, 1986. ACM Press.

[Kwo89] K.L. Kwok. A neural network for probabilistic information retrieval. In *SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–30, New York, NY, USA, 1989. ACM Press.

[Lew05] Dirk Lewandowski, editor. *Web Information Retrieval. Technologien zur Informationssuche im Internet*. Deutsche Gesellschaft für Informationswissenschaft und Informationspraxis e.V., 1 edition, 2005.

[MBCS+06] DJ. Mackay, SE. Boonen, J. Clayton-Smith, J. Goodship, JM. Hahnemann, SG. Kant, PR. Njolstad, NH. Robin, DO. Robinson, R. Siebert, JP. Shield, HE. White, and IK. Temple. A maternal hypomethylation syndrome presenting as transient neonatal diabetes mellitus. *Advances in human genetics*, 120(2), 2006.

[OD04] KK. Ong and DB. Dunger. Birth weight, infant growth and insulin resistance. *European journal of endocrinology / European Federation of Endocrine Societies*, 151, 2004.

[Por97] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[pub] Pubmed. http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed.

[PZF+97] A. Pugliese, M. Zeller, A. Jr. Fernandez, LJ. Zalcberg, RJ. Bartlett, C. Ricordi, M. Pietropaolo, GS. Eisenbarth, ST. Bennett, and DD. Patel. The insulin gene is transcribed in the human thymus and transcription levels correlated with allelic variation at the ins vntr-iddm2 susceptibility locus for type 1 diabetes. *Nature genetics*, 15(3), 1997.

[RJ88] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. pages 143–160, 1988.

[Roj93] Raul Rojas, editor. *Theorie der neuronalen Netze: Eine systematische Einführung*. Springer Verlag, 1 edition, 1993.

[Sac] Saccharomyces genome database. http://www.yeastgenome.org/.

[Sch97] Andreas Scherer, editor. *Neuronale Netze: Grundlagen und Anwendungen*. Vieweg, 1 edition, 1997.

[Sha76] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[SN00] ST. Sharma ST and JE. Nestler. Prevention of diabetes and cardiovascular disease in women with pcos: treatment with insulin sensitizers. *Baillière's best practice & research. Clinical endocrinology & metabolism*, 2000.

[TC90] H. Turtle and W. B. Croft. Inference networks for document retrieval. In *SIGIR '90: Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–24, New York, NY, USA, 1990. ACM Press.

[TH87] David W. Tank and John J. Hopfield. Collective computation in neuronlike circuits. *Scientific American*, 6(257), 1987.

[TMR+03] E. Tafeit, R. Moller, S. Rackl, A. Giuliani, W. Urdl, U. Freytag, K. Crailsheim, K. Sudi, and R. Horejsi. Subcutaneous adipose tissue pattern in lean and obese women with polycystic ovary syndrome. *Experimental biology and medicine (Maywood, N.J.)*, 228(6), 2003.

[VBC+96] P. Vafiadis, ST. Bennett, E. Colle, R. Grabs, CG. Goodyer, and C. Polychronakos. Imprinted and genotype-specific expression of genes at the iddm2 locus in pancreas and leucocytes. *Journal of autoimmunity*, 9(3):397–403, 1996.

[VHL+05] F. Vasseur, N. Helbecque, S. Lobbens, V. Vasseur-Delannoy, C. Dina, K. Clement, P. Boutin, T. Kadowaki, and PE. Scherer Pand P. Froguel. Hypoadiponectinaemia and high risk of type 2 diabetes are associated with adiponectin-encoding (acdc) gene promoter variants in morbid obesity: evidence for a role of acdc in diabesity. *Diabetologia*, 48(5), 2005.

[vR77] C.J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *J. of DOC*, 33:106–119, 1977.

[ZPOL97] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, and Wei Li. New algorithms for fast discovery of association rules. Technical Report TR651, 1997.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und nur mit erlaubten Hilfsmitteln angefertigt habe.

Konstanz, den 12. September 2006

Kilian Thiel