Gerold Schneider, Heinrich Zimmermann

Text-Mining-Methoden im Semantic Web

Aufbau, Pflege und Nutzung großer Wissensdatenbanken erfordern den kombinierten Einsatz menschlicher und maschineller Informationsverarbeitung. Da große Teile des menschlichen Wissens in Textform vorliegen, bieten sich Methoden des Text Mining zur Extraktion von Wissensinhalten an. Dieser Artikel behandelt Grundlagen des Text Mining im Kontext des Semantic Web. Methoden des Text Mining werden besprochen, die für die halbautomatische Annotierung von Texten und Textteilen eingesetzt werden, insbesondere Eigennamenerkennung (Named-Entity Recognition), automatische Schlüsselworterkennung (Keyword Recognition), automatische Dokumentenklassifikation, teilautomatisches Erstellen von Ontologien und halbautomatische Faktenerkennung (Fact Recognition, Event Recognition). Es werden auch kritische Hintergrundfragen aufgegriffen. Das Problem der zu hohen Fehlerrate und der zu geringen Performanz automatischer Verfahren wird diskutiert. Zwei Beispiele aus der Praxis werden vorgestellt: Erstens das Forschungsprojekt OntoGene der Universität Zürich, in dem Protein-Protein-Interaktionen als Relationstripel aus der Fachliteratur extrahiert werden, und zweitens ein ontologiebasierter Tag-Recommender, der die manuelle Vergabe von Schlüsselwörtern an Wissensressourcen unterstützt.

Inhaltsübersicht

1 Annotierungsaufwand für das Semantic Web

Das Semantic Web hilft den Usern, Inhalte besser zu finden, zu organisieren und zu bearbeiten. Das Anreichern der Dokumente mit semantischer Information soll eine automatisierte Weiterverarbeitung, z.B. durch Softwareagenten, unterstützen. Während die Semantic-Web-Sprachen wie RDF(S) (Resource Definition Framework (Schema)) und OWL (Web Ontology Language) gut erforscht und standardisiert sind, gibt es viel weniger Forschung zur Frage, wie die enormen Mengen an Webdaten semantisch annotiert werden sollen, also die Transformation von konventionellen Webseiten zu reich annotierten Semantic-Web-Ressourcen.

Außer für Experimente und Demonstrationen ist die manuelle Eingabe von realistischen RDF- und OWL-Ontologien und -Ressourcen kaum machbar. Es gibt auch vielversprechende Ansätze, Untermengen der natürlichen Sprache direkt und ohne Mehrdeutigkeiten in Semantic-Web-Sprachen zu übersetzen [Kaljurand 2008]. Die klassische Antwort, um den Schwierigkei-

ten der Syntax von Semantic-Web-Sprachen ausweichen zu können, ist die Verwendung von Ontologie-Editoren wie z.B. Protégé oder OntoEdit. Die Tatsache, dass die Annotierung großer Textmengen zu aufwendig ist, bleibt aber bestehen, sodass umfassendere Aufgaben kaum realistisch machbar sind. Schon seit einigen Jahren wird deshalb vorgeschlagen (z.B. [Rinaldi et al. 2003]), Computerlinguistik und Sprachtechnologie (Natural Language Processing, NLP) zu verwenden. Wir erklären Basismethoden der Sprachtechnologie und des Text Mining in Abschnitt 2 und erläutern zwei konkrete Anwendungen in Abschnitt 4.

Nach den Erfahrungen mit Technologien der künstlichen Intelligenz (KI), bei der viele Ansätze zu große Fehlerraten aufwiesen oder nicht skalierten, ist Vorschlägen zur Verwendung von Sprachtechnologie einerseits mit Skepsis entgegenzutreten, wie wir in Abschnitt 3 berichten. Andererseits haben sich die Umstände geändert. Heutige Systeme sind stark statistisch basiert, Evaluierung und Skalierung stehen im Zentrum. Die Fehlerraten sind für einige Anwendungen tolerierbar klein geworden, für andere rücken halbautomatische Systeme, bei denen ein maschineller Klassifikator und der menschliche Annotator eng zusammenarbeiten, in den Fokus der Forschung.

2 Methoden des Text Mining für das Semantic Web

2.1 Eigennamenerkennung (Named-Entity Recognition and Grounding)

Das Erkennen von Instanzen von Eigennamen war schon lange eine weitverbreitete Anwendung der Sprachtechnologie. In einfachen Ausprägungen der Eigennamenerkennung werden Eigennamen und Ketten von Eigennamen gesucht. Meist verwendet man einen sogenannten *Tagger*, ein automatisches Tool, das für alle Wörter im Lauftext die Wortklasse (z.B. Substantiv, Eigenname, Verb, Adjektiv) angibt. Tagger haben meist Fehlerraten unter 5 Prozent, gerade im Englischen ist die Erkennung der Wortklasse Eigenname meist einfach, da sie im Gegensatz zu Substantiven groß geschrieben werden. Um verschiedene Schreibweisen desselben Begriffes aufeinander abzubilden, kommen Fuzzy-Match-Methoden (ähnlich wie bei Korrekturvorschlägen von Spell-Checkern) und Synonymlisten zum Einsatz.

Ein wichtiges Teilgebiet der Eigennamenerkennung ist die Terminologieerkennung, bei der Fachbegriffe gesucht werden, und diese müssen nicht unbedingt Eigennamen sein. Fachbegriffe kann man idealerweise daran erkennen, dass sie in Fachwörterbüchern vorkommen. Oft sind diese aber unvollständig. Sie können eventuell auch daran erkennbar sein, dass sie in allgemeinen Wörterbüchern nicht vorkommen, z.B. von einem allgemeinen Spell-Checker zurückgewiesen werden. Manchmal erkennt man sie an typischen Nominalisierungsendungen (z.B. *-ion*, *-ung*), manchmal an ihrem Kontext (steht z.B. nach dem Wort *sogenannt(er)* oder in Kursivschrift). Diese Kriterien sind aber lückenhaft. Wörter, die in Fachdokumenten häufig, allgemein aber seltener vorkommen, sind gute Termkandidaten (dazu kann man den TF-IDF-Algorithmus, den wir im nächsten Abschnitt besprechen, verwenden).

Mehrwortterme erkennt man recht gut daran, dass sie statistisch auffallend häufig in Kombination erscheinen, sogenannte Kollokationen (siehe z.B. [Evert 2005] für Kollokationsforschung).

Der Tatsache, dass richtig erkannte Instanzen ohne eine Zuordnung zu einer universellen Beschreibung nur beschränkte Anwendungen haben, wurde man sich erst später bewusst. Die Zuordnung (*is_a*) zu einem universell eindeutigen Bezeichner, z.B. zu einem Uniform Resource Identifier (URI), wird als *Grounding* bezeichnet. Grounding muss oft Mehrdeutigkeiten auflösen. So muss z.B. das Erkennen des Personennamens *Helmut Schmidt* als Lauftext noch nicht zwingend bedeuten, dass es sich um den Ex-Bundeskanzler handelt – es leben über ein Dut-

zend Helmut Schmidts in Deutschland. Terme oder Eigennamen, die nicht in Fachwörterbüchern oder anderen Ressourcen beschrieben sind, lassen sich schwierig grounden. Allenfalls kann man einen Annotator auffordern, ein Wörterbuch oder eine Ontologie zu ergänzen. Dessen Arbeit wird allerdings wesentlich erleichtert durch ein System, das in der Mehrzahl der Fälle richtige Vorschläge macht. [Weeds et al. 2005] stellen ein Verfahren vor, das in bis zu 75 Prozent der Vorkommen für Proteine das richtige Grounding vorschlägt. Das Verfahren basiert auf dem syntaktischen Kontext und geht im Prinzip davon aus, dass der syntaktische Kontext semantisch ähnlicher Wörter identisch ist. So sind z.B. fast alle syntaktischen Objekte des Verbs *essen* Lebensmittel.

Fachbegriffe – das Wort *Grounding* selbst ist ein gutes Beispiel – sind häufig mehrdeutig. Durch Kontext und Thema des Dokumentes lässt sich oft die Mehrdeutigkeit auflösen: Im Kontext von Flugzeugen trifft man typischerweise eine Lesart, im Kontext von Texten über Terminologieerkennung eine andere.

Das Erkennen von Instanzen und Zuordnen zu einer Klasse ist nur eine von unzähligen Relationen, die man im Semantic Web ausdrücken will. Die hier besprochenen Relationen mit Verweisen auf die Struktur dieses Abschnitts sind unten in der Übersicht erkennbar.

2.2 Automatische Schlüsselworterkennung (Keyword Recognition)

Viele der in einem Text gemachten Aussagen sind für den Text nicht zentral, sondern Hintergrundinformation, ein Nebenschauplatz, und dienen dem Aufbau eines Argumentes oder als Vorbereitung oder Reflexion einer Handlung oder These. Niemand würde behaupten, dass Shakespeares Macbeth ein Buch übers Händewaschen ist (obwohl Lady Macbeth das tut), dass Darwins wichtige Aussage sei, dass Tiere leben, dass eine Haupteigenschaft des Mikrowellenherdes sei, dass er am Strom angeschlossen werden muss (steht in der Betriebsanleitung), dass man darin keine Hunde trocknen soll (steht da manchmal auch).

Ein frühes auf künstlicher Intelligenz basierendes System, das aus Nachschlagewerken Faktenwissen über die Welt lernen sollte, habe angeblich als eine der zentralen Schlussfolgerungen folgenden Satz ausgespuckt: »Most people are famous.« Diese Aussage ist zwar absolut folgerichtig, da fast alle Leute, die im Großen Brockhaus eingetragen sind, berühmt sind, aber sie ist irrelevant.

Für die Keyword Recognition geht es gerade um die Erkennung relevanter Themen in einem Dokument (*Aboutness*). Ein simpler und weitverbreiteter Algorithmus ist *Term Frequency, Inverse Document Frequency (TF-IDF)*, der wie folgt funktioniert: Für jedes Wort in einem Dokument wird untersucht, wie häufig es in diesem Dokument vorkommt, geteilt durch die Anzahl der Dokumente in einem Referenzkorpus, in dem das Wort enthalten ist. Die Intention ist klar: Begriffe, die in einem Dokument oft erwähnt werden, sind wichtig (Term Frequency). Falls es aber Begriffe sind, die ganz allgemein oft vorkommen, so bleiben sie relativ unwichtig (Inverse Document Frequency). Jedes Wort erhält einen Score, die Wörter mit den höchsten Scores können als Tag dem Dokument zugewiesen werden, ähnlich wie Aboutness Tags aus dem Social Tagging. Wird ein allgemeines Referenzkorpus verwendet, so dominieren Fachbegriffe der Domäne des Dokumentes. Durch Wahl einer Textsammlung aus der Domäne

Erkennen von Instanzen und Zuordnen zu einer Klasse (*is_a*)	Abschnitt 2.1
Thema eines gegebenen Textes (*Aboutness*-Relation)	Abschnitt 2.2
Relationen zwischen einzelnen Texten	Abschnitt 2.3
Relationen zwischen im Text vorkommenden Instanzen	Abschnitt 2.4

kristallisieren sich die zentralen Begriffe, Terme und Themen des gegebenen Textes für einen Experten heraus.

Die Performanz dieses Verfahrens für Keyword Recognition ist für viele Anwendungen genügend gut. Aber wie Social Tagging ist das Verfahren mit dem Mangel behaftet, dass das Grounding oft unklar ist. Durch die Verwendung bekannter Ressourcen zur Erkennung von Synonymen und Mehrdeutigkeit, z.B. WordNet, lässt sich die Situation aber etwas verbessern.

Ein klassisches Einsatzgebiet von Keyword Recognition ist die Suche von Dokumenten zu einem gegebenen Thema, ein weiteres die automatische Dokumentenklassifikation, die wir im Folgenden erläutern.

2.3 Automatische Dokumentenklassifikation

Die Zuordnung von Dokumenten zu einer Klasse ist Aufgabe der automatischen Dokumentenklassifikation. Diese erlaubt es einem beispielsweise, verwandte Dokumente zu finden, eingehende Informationen dem geeigneten Experten zuzuspielen oder aufgrund von Produktbeschreibungen potenziell interessierte Kunden darauf aufmerksam zu machen.

Ein einfacher Ansatz zur Dokumentenklassifikation ist der Vergleich der Schlüsselwörter (siehe Abschnitt 2.2) der zu klassifizierenden Dokumente, sei es untereinander oder zu einer vorgegebenen universellen Semantic-Web-Klasse. Bei großer Überlappung der Schlüsselwörtermenge ist davon auszugehen, dass die Dokumente zur gleichen Klasse gehören. Je feiner granuliert die Unterteilung sein soll, desto mehr Schlüsselwörter sind jedoch nötig, und das Scoring der Schlüsselwörter sollte mit berücksichtigt werden. Vektormodelle können dies leisten. Jedem Schlüsselwort, oder gar jedem Wort im Dokument, wird dazu eine Dimension in einem Vektorraum zugeordnet mit der Häufigkeit des Wortes oder besser dem TF-IDF-Wert des Wortes. Stellen wir uns als einfaches Beispiel für ein Vektormodell vor, dass drei Dokumente, in denen nur die drei Schlüsselwörter *market*, *computer* und *government* vorkommen, durch ein Vektormodell verglichen werden. Jedes Schlüsselwort entspricht einer Dimension im Vektorraum. Die Worthäufigkeiten aus der Tabelle 1 ergeben dabei das Vektormodell in Abbildung 1.

Falls die Klassen, denen die Dokumente zugeordnet werden sollen, nicht schon vorgegeben sind, so kann man Clustering-Algorithmen verwenden zur automatischen Klassenbildung. Der Vergleich der Winkel zwischen verschiedenen Dokumenten (Kosinus als Ähnlichkeitsmaß) lässt Klassen bilden. In unserem Beispiel ist der Winkel δ zwischen den Vektoren für dok2 und dok3 besonders klein. Gruppen von Dokumenten, deren Vektoren sehr kleine Winkel bilden, formen eine Klasse. Die Klassenbeschreibung ergibt sich aus den häufigen Schlüsselwörtern, aber das Grounding der Klasse ist dann nicht eindeutig. Falls die Klassen vorgegeben und gegroundet sind, so kann man mit einer Schlüsselwortliste oder besser einer kleinen Sammlung prototypischer Dokumente den zentralen Vektor einer Klasse bestimmen und die Dokumente dann zuordnen.

Vektorbasierte Systeme sind im Information Retrieval weit verbreitet und haben viele Anwendungen. Eine Anwendung zur Auflösung von Wortsinnambiguitäten (z.B. Grounding in verschiedenen Kontexten) ist z.B. in [Schütze 1998] beschrieben.

Worthäufigkeit	market	computer	government
dok1	2	8	1
dok2	4	2	6
dok3	5	1	7

Tab. 1: Zählungen von drei Schlüsselwörtern in drei fiktiven Dokumenten

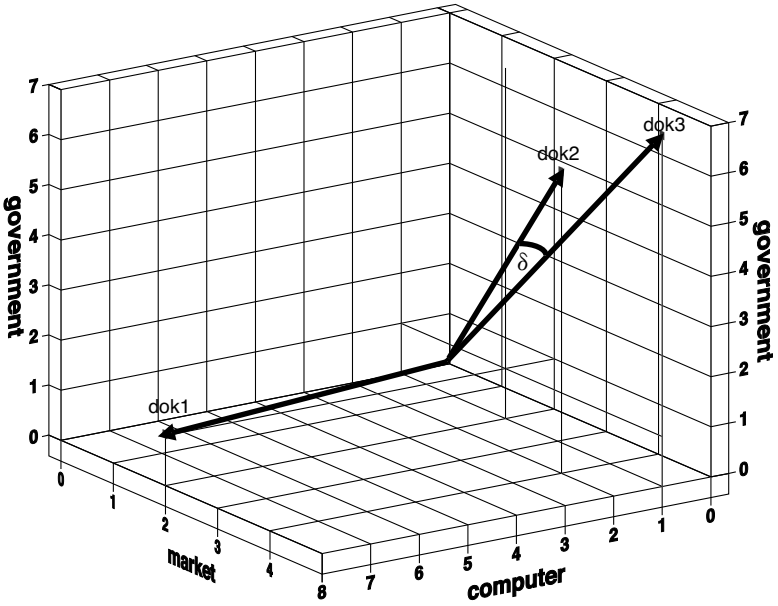

Abb. 1: Das aus der Tabelle 1 resultierende Vektormodell

Die bisher diskutierten Basistechnologien vermögen zwar die zentralen Themen eines Textes auszudrücken (Aboutness) und Ähnlichkeitsbeziehungen zwischen verschiedenen Dokumenten zu beschreiben, aber vieles, was man vom Semantic Web erwartet, können sie nicht erfüllen. Insbesondere sind sie nicht fähig, (1) Relationen zwischen im Dokument erwähnten Instanzen oder Klassen zu beschreiben und (2) beim Aufbau von Ontologien eine Hilfe zu bieten. Wir beschäftigen uns mit Punkt 1 im folgenden Abschnitt sowie in Abschnitt 3 und mit Punkt 2 in Abschnitt 2.5.

2.4 Automatische Faktenerkennung (Fact Recognition, Event Recognition)

Die meisten Ontologien verknüpfen definierte Konzepte mittels der semantischen Relationen *is-a* (Hyponymie), *part-of* (Meronymie) und *has-property*. Fachtexte enthalten aber auch Informationen, die weit über diese Relationen hinausgehen, aber für die Domäne zentral sind. In der Universitätsontologie in [Antoniou & Harmelen 2008] ist es zentral, wer welche Kurse unterrichtet. In einer chemischen Ontologie ist es z.B. sehr wichtig, zu wissen, welche Substanz mit welchen anderen Substanzen reagiert, eine Wissensform, die in unzähligen Publikationen in Form von Aussagen enthalten ist, z.B. in der Form *A has been shown to react with B,* in der die Domänenrelation *react* zwischen A und B, also das Faktum *react(A,B)*, ausgedrückt ist.

A has been shown to **react** with **B**. => react(A,B)

Um dieses Wissen zu extrahieren, werden entweder einfache Oberflächenmuster (»Surface Patterns«) oder aufwendige syntaktische Analysen eingesetzt. In Oberflächenverfahren wird z.B. von einem Domänenwort aus eine bestimmte Anzahl Wörter nach links und rechts geschaut, und dort stehende Terme werden als Argumente der Relation aufgefasst. Der Vorteil von syntaktischen Analyseverfahren besteht darin, dass die Fehlerraten niedriger sind. Zwar machen automatische Syntaxanalysetools (sog. *Parser*) auch Fehler, aber die Ableitung folgender falscher Fakten ist bei oberflächenbasierten Systemen viel wahrscheinlicher:

Experts fear the **virus** will spread.
=> fear(expert,virus)

C and **A react** with each other, but **B** with D.
=> react(A,B)

X, if processed with **A, reacts** with **B.**
=> react(A,B)

Wir stellen ein Beispiel in Abschnitt 4.1 vor.

2.5 Teilautomatisches Erstellen von Ontologien

Die bisher beschriebenen Zugänge erlauben es bestenfalls, in bestehenden Ontologien Instanzen Klassen zuzuordnen. Auf die Frage, wie Sprachtechnologie helfen kann, Begriffssysteme aufzubauen, gibt es erst Teilantworten.

Die Erstellung einer Ontologie bedarf ohnehin sorgfältiger Planung und Diskussionen mit allen Betroffenen, vom Anwender bis zum Experten. Die Zuordnung von Instanzen zu Klassen lässt sich leichter automatisieren. Trotzdem können sprachtechnologische Zugänge bei der Erstellung von Begriffssystemen Hilfe leisten, indem sie erstens Vorschläge liefern und zweitens mögliche Lücken aufdecken. Ein klassisches Vorgehen besteht darin, eine Reihe von linguistischen Mustern in Domänentexten zu suchen, die typischerweise nahe Verwandtschaft (Geschwister in der Ontologie) und Hyponymie ausdrücken. Es gibt wenige explizite Varianten, Synonymie und Hyponymie auszudrücken, Beispiele sind:

X und Y sind gleich
X ist eine Art von Y

Aber gerade uns selbstverständliche oder Experten bekannte Verwandtschaften werden kaum explizit ausgedrückt. Zum Glück sind implizite Nennungen viel häufiger. Deutsche Muster für implizite Geschwisterbeziehungen zwischen X und Y (und evtl. Z) sind z.B.:

X und/oder Y
X, Y und/oder Z
X, aber nicht Y

Analog gilt dies in vielen anderen Sprachen. Deutsche Muster für Hyponymiebeziehungen zwischen X und Y sind beispielsweise:

X, (wie) z.B. Y,
X, insbesondere Y,
X, vor allem Y,
alle X außer Y
Y und/oder andere/alle X

Oder im Englischen:

X, such as Y
X, including Y
X and in particular Y
Y and/or other/all X

Diese Muster werden auch nach ihrer Erfinderin als Hearst-Patterns bezeichnet [Hearst 1992]. Juristische Texte sind z.B. reich an Formulierungen, die solche Muster enthalten. Damit das Verfahren funktionieren kann, sind aber große Textmengen nötig.

Ein weiteres Verfahren zur Erkennung verwandter oder synonymer Begriffe besteht darin, Wörter zu erkennen, deren Kontext identisch oder sehr ähnlich ist. Beispielsweise treten die Wörter »Astronaut« und »Kosmonaut« selten im selben Text auf, aber die sie umgebenden Worte sind oft identisch. Der Kontext eines Wortes kann als Vektormodell dargestellt werden (siehe Abschnitt 2.3), Verfahren wie in [Schütze 1998] beschrieben können so zur Erkennung von Geschwistern in Ontologien verwendet werden und Vorschläge liefern, die Experten dann annehmen oder ablehnen. Anders als wenn man sich Begriffe aus den Fingern saugt, hat dieses Vorgehen den Vorteil, dass kaum Synonyme vergessen werden. Wenn man also einen tiefen Schwellenwert im Kosinusmaß setzt und bereit ist, viele Vorschläge zu verwerfen, wird die Ausbeute entsprechend groß.

Eine erfolgreiche Methode des Kontextverfahrens für das teilautomatische Erstellen von Ontologien ist, wie bei [Weeds et al. 2005] (siehe Abschnitt 2.1), den syntaktischen Kontext zu verwenden. In Kombination mit Clustering-

Methoden erreichen [Cimiano et al. 2005] eine Präzision von 29 Prozent bei einer Ausbeute von bis zu 65 Prozent.

Einen guten Überblick über den Stand der Forschung der hier in Abschnitt 2 vorgestellten Methoden geben [Buitelaar et al. 2005].

3 Hintergrundfragen

Wir haben wiederholt von Fehlerraten gesprochen. Fehlerrate ist 1 minus Erfolgsrate. Die klassischen Erfolgsratenmaße sind Präzision und Ausbeute sowie Kombinationen davon (z.B. f-Measure). Präzision misst, wie viele der von einem automatischen System vorgeschlagenen Vorkommen vom menschlichen Annotator als richtig eingestuft werden. Ausbeute misst, wie viele der von einem menschlichen Annotator gefundenen Vorkommen auch von einem automatischen System gefunden werden.

Ein allgemein bekanntes, ungeschriebenes Gesetz der Informationsverarbeitung, der Sprachtechnologie, der künstlichen Intelligenz, des Text Mining usw. ist, dass je derivierter oder indirekter oder vom Text abstrahierter eine Metainformation ist, desto größer ist die Fehlerrate. Während z.B. Tagging eine Fehlerrate von unter 5 Prozent aufweist, hat syntaktische Analyse, die typischerweise auf dem Tagging aufbaut, eine grob gesagt doppelt so hohe Fehlerrate. Semantische Analysen bauen klassischerweise auf der Syntax auf und haben wiederum wesentlich höhere Fehlerraten. Bezogen auf die Anwendung von Text Mining bedeutet das, dass man entsprechend der zunehmenden Schwierigkeit und aufgrund des teilweise aufbauenden Charakters der vorgestellten Technologien folgende Fehlerratenhierarchie erwarten muss:

Abschnitt 2.1 < Abschnitt 2.2 < Abschnitt 2.3 etc. ...

Obwohl die Sprachtechnologie ständig Fortschritte macht, gilt das Gesetz des abnehmenden Grenznutzens (»Ceiling Effect«): Für jede kleine Verbesserung erhöht sich der Aufwand exponentiell, und die Fehlerrate nähert sich asymptotisch einem gewissen Grenzwert (»Ceiling«), den man kaum mehr –höchstens mit einem prinzipiell anderen Verfahren – je unterschreiten kann.

Neben der zunehmenden Abstrahiertheit und Abhängigkeit von früheren Ergebnissen ist eine weitere Teilerklärung, dass die Divergenzen zwischen Annotatoren (»Inter-Annotator Disagreement«) für komplexe Entscheidungen auch höher sind. Ludwig Wittgensteins sprachphilosophische – und auf den ersten Blick abgehobene – Erkenntnisse erlangen hier klare, täglich deutlich spürbare Bedeutung: Man kann alles (ganz besonders betroffen sind Abstrakta) auf ganz verschiedene Weisen modellieren, je nach Sichtweise. Bei genauerer Betrachtung sind auch fast alle Wörter mehrdeutig. Wortsinn-Disambiguierung ist heute eine der großen Forschungsrichtungen der Computerlinguistik.

Ein Korollar, das als positiver Nebeneffekt aus dem Grenzwertnutzen folgt, ist, dass man oft mit einfachen Zugängen schon recht gute Ergebnisse erzielen kann. Insbesondere erreicht man oft schnell eine gute Präzision, während die Ausbeute bei einfachen Verfahren noch schlecht bleibt. Für schlussfolgernde Systeme gilt zum Glück: Präzisionsfehler (also etwas Falsches als richtig annehmen) sind viel schlimmer als ein Ausbeutefehler (also ein Fakt verpasst zu haben), somit kann man Systeme mit guter Präzision, aber schlechter Ausbeute doch gut einsetzen.

Ebenso ist es ein Glücksfall, dass natürliche Sprache stark redundant ist. Wichtige Nachrichten erscheinen meist in verschiedenen Zeitungen gleichzeitig in verschiedenen Formulierungen. Ein Autor, der die zentrale Aussage seines wissenschaftlichen Artikels nur einmal macht und sie nicht im Abstract und in den Schlussfolgerungen leicht anders formuliert wiederholt, ist ohnehin schlecht beraten: Das Risiko, dass auch menschliche Leser den springenden Punkt verpassen, wäre zu hoch. Als Folge der sprachlichen Redundanz wächst die Ausbeute automatischer Zugänge von alleine.

Obwohl auch heutige Systeme noch Fehler machen, haben sich viele Umstände seit den nur mäßigen Erfolgen im KI-Zeitalter geändert, wie wir nun zusammenfassen.

Erstens sind heutige Systeme stark statistisch basiert. Fakten, auf denen Schlussfolgerungen beruhen, liegen jeweils statistisch gewichtet vor. Im Falle von sich widersprechenden Schlussfolgerungen bricht das System nicht mehr zusammen, sondern die als wahrscheinlicher gewichtete Folgerung obsiegt. Fakten mit hoher statistischer Gewichtung sind meist zuverlässiger, haben aber geringe Ausbeute; somit kann durch die Wahl verschiedener Schwellenwerte oft ein geeigneter Kompromiss zwischen Präzision und Ausbeute gefunden werden. Eine enorme Effizienzsteigerung lässt sich bei statistischen Verfahren oft dadurch erreichen, dass wenig Erfolg versprechende Möglichkeiten durch Stutzen von Suchbäumen (»Pruning«) gar nicht in Betracht gezogen werden.

Zweitens stehen Evaluierung und Skalierung heute im Zentrum. Schon in der Evaluierungsphase werden realistische, große Datenmengen verwendet, sodass Skalierungsprobleme in der Zielanwendung wesentlich seltener auftreten.

Drittens sind die Fehlerraten aufgrund der Fortschritte für einige Anwendungen tolerierbar klein geworden. Für andere, meist komplexe Anwendungen rücken halbautomatische Systeme, bei denen ein maschineller Klassifikator und der menschliche Annotator eng zusammenarbeiten, in den Fokus der Forschung, wie wir im folgenden Abschnitt erläutern.

4 Beispiele aus der Praxis

In der Praxis gibt es schon viele Programme und Webservices, die einige der oben beschriebenen Sprachtechnologien verwenden, um Dokumente semantisch anzureichern. Ein beliebter Service ist z.B. OpenCalais. Der rohe Text wird in einer Webform eingereicht, der semantisch annotierte Text wird grafisch wie in Abbildung 2 illustriert, aber auch als RDF-File zurückgeliefert.

Die Eigennamenerkennung verschiedener Kategorien liefert gute Ergebnisse, die automatische Schlüsselworterkennung (als »Social Tags« bezeichnet) scheint intuitiv richtig. Die Faktenerkennung ist vermutlich oberflächenbasiert und scheint ziemlich partiell: Einige gefundene Ereignisse sind wenig relevant, andere relevante oder komplexere Relationen werden nicht gefunden, Letzteres deutet auf eine eher geringe Ausbeute. Keine der acht vorgeschlagenen generischen Relationen ist aber falsch, was auf eine gute Präzision hindeutet. Die hier gezeigte Annotierung ist nicht unbedingt repräsentativ, denn oft stehen registrierten und zahlenden Usern auch fortgeschrittenere Methoden zur Verfügung.

Wir betrachten im Folgenden zwei Beispiele von fortgeschrittenen Anwendungen etwas detaillierter.

4.1 Protein-Protein-Interaktionen: OntoGene

Die Extraktion von Interaktionen zwischen Proteinen (z.B. Genen) aus Fachzeitschriften, Patenten und anderen Publikationen ist eine wichtige Applikation von Information Retrieval, da sie für Systembiologie und Life Sciences zentral ist. Da biologisches Wissen gut strukturiert ist, bildet es auch eine Semantic-Web-Applikation. Proteine sind in großen systematischen Ontologien erfasst, z.B. UniProt (*www.uniprot.org*), die möglichen Interaktionen bilden eine geschlossene, gut dokumentierte Klasse (typische Relationswörter sind *bind, block, interact, react, activate, co-activate* etc.). Die Aufgabenstellung umfasst zwei Teile: 1. das Erkennen von Proteinen und deren Rückführung auf eine universelle Identität (Grounding, siehe Abschnitt 2.1) und 2. für Proteine, die im gleichen Satz vorkommen, zu entscheiden, ob sie in einer syntaktischen Beziehung, die eine biomedizinische Relation ausdrückt, stehen (siehe Abschnitt 2.4). Wir stellen dazu das Forschungsprojekt OntoGene (*www.ontogene.org*) an der Universität Zürich vor [Rinaldi et al. 2008; Kaljurand et al. 2009; Schneider et al. 2009].

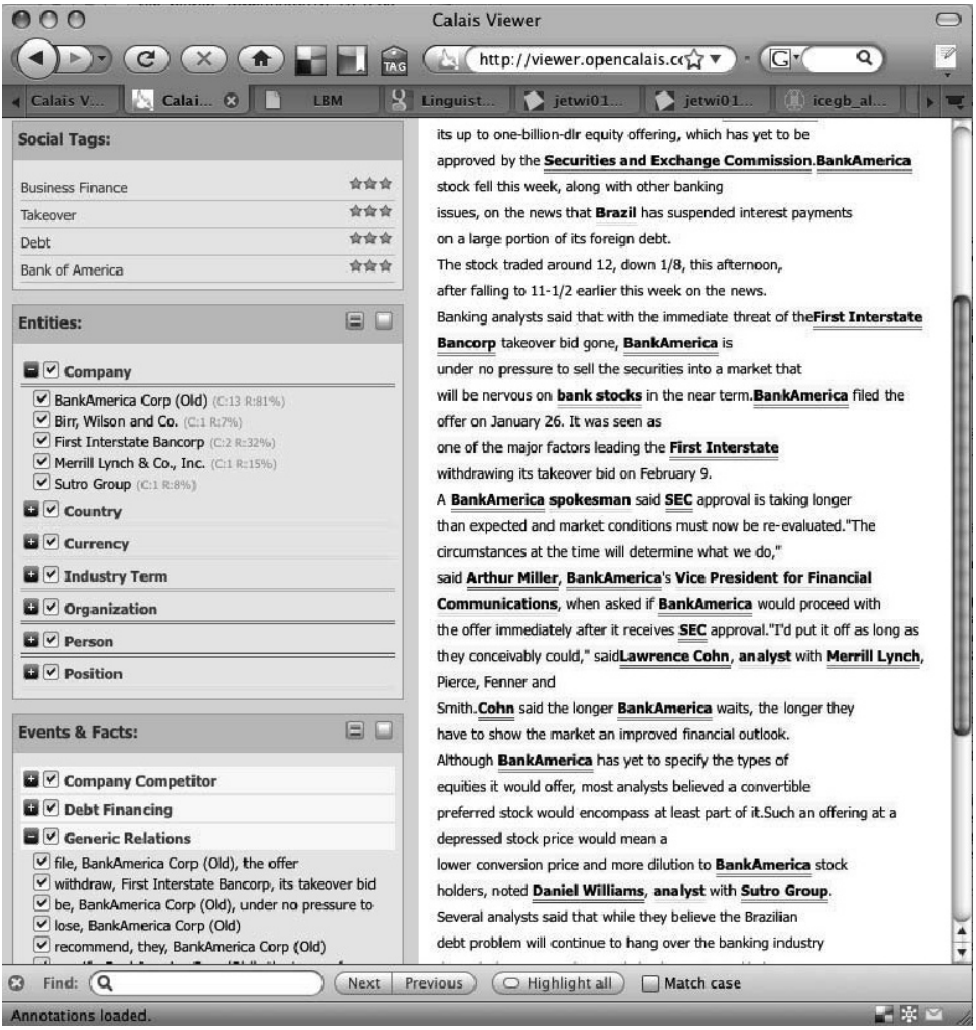

Abb. 2: Screenshot aus dem freien OpenCalais-Service

Im 1. Teil, Erkennen von Proteinen und Grounding, werden alle aus Ontologien wie UniProt bekannten Schreibweisen sowie deren typische Variationen (Leerfelder, Groß-/Kleinschreibung, arabische oder römische Zahlen, Bindestriche) in den vorliegenden Dokumenten gesucht. Die so erkannten Terme können nicht immer eindeutig einer Position in der Ontologie zugeordnet werden. Oft stehen sie, dies gilt vor allem für Proteine, die in verschiedenen Lebewesen eine wichtige Rolle spielen, an mehreren Orten. Basierend auf den im Dokument vorkommenden Organismen werden die Proteine so weit wie möglich disambiguiert. Das Verfahren ist im Detail beschrieben in [Kaljurand et al. 2009].

Im 2. Teil, Faktenerkennung, muss für alle Proteine, die im gleichen Satz enthalten sind, entschieden werden, ob im Text eine Interaktion zwischen ihnen beschrieben ist. Dazu vergleicht der Algorithmus alle syntaktischen Teil-

strukturen, in denen zwei erkannte Proteine sowie ein Relationswort vorkommen, mit einer vorgängig erstellten Sammlung relevanter syntaktischer Muster. Die syntaktischen Muster sind flexibel, um die Abdeckung und damit die Ausbeute zu erhöhen. Beispielsweise kürzen sich die Muster automatisch um semantisch weniger relevante Worte, wie z.B. *a group of* in *A binds to a group of B.*

A binds to a group of B => A binds to B => bind(A,B)

Damit werden die syntaktischen Muster einfacher und genereller. Das genaue Verfahren ist in [Schneider et al. 2009] beschrieben. Wir haben mit verschiedenen Varianten unseres Systems an mehreren öffentlichen Wettbewerben (»Shared Tasks«) teilgenommen und überdurchschnittliche Ergebnisse erzielt, insbesondere eine relativ gute Ausbeute. Für Protein-Protein-Interaktionen (ohne komplexe Interaktion-Interaktion-Interaktionen) erreichten wir z.B. im BioNLP Shared Task[1] 57 Prozent Präzision und 40 Prozent Ausbeute, und im BioCreative II.5 Shared Task (*www.biocreative.org*) wurden wir als eines der drei besten Systeme klassiert.

Die erkannten Fakten werden als Tripel abgelegt. Da die Fehlerraten noch zu hoch sind für eine vollautomatische Anwendung, fügen wir nun einen halbautomatischen Zwischenschritt ein. Wir entwickeln interaktive grafische Tools, die gefundene Relationen darstellen und es Annotatoren erlauben, vorgeschlagene Relationen per Mausklick anzunehmen oder zu verwerfen. Der Screenshot eines solchen Tools ist in Abbildung 3 ersichtlich.

1. *http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/*

Abb. 3: Screenshot eines OntoGene-Annotator-Interfaces

Der Annotierungsaufwand wird viel kleiner, wenn ein Annotator einen Vorschlag annehmen oder verwerfen kann, statt einen ganzen Artikel lesen zu müssen: »For biologists, an automated system with high recall and even moderate precision [...] confers a great advantage over skimming text by eye« [Müller et al. 2004].

4.2 Ontologiebasierter Tag-Recommender

Große Social-Bookmarking-Plattformen, wie »del.icio.us«, repräsentieren Spuren von Milliarden von Tag-Events. In einem Tag-Event ordnet ein User zu einem bestimmten Zeitpunkt einer Webressource ein Tag zu. Ein einfacher Tag-Recommender schlägt ihm dabei Tags vor, die andere User für diese Ressource schon früher vergeben haben.

In anspruchsvolleren Anwendungsfällen ist der Benutzerkreis kleiner und die Tags müssen mit einem kontrollierten Vokabular abgeglichen werden. Auch automatische Annotationsservices, wie »OpenCalais«, können in solchen Szenarien höchstens Teilaufgaben erfüllen. Aufbau, Pflege und Nutzung von Fachontologien und zugeordneten Webressourcen gelingen am besten, wenn die Vorteile der menschlichen und der maschinellen Informationsverarbeitung geeignet kombiniert werden.

Der Einsatz eines ontologiebasierten Tag-Recommenders [Blumauer & Hochmeister 2009] kann wie folgt skizziert werden:

1. Ein Annotationsservice (z.B. »OpenCalais«) bestimmt charakteristische Terme für die zu taggenden Ressourcen.
2. Diese Terme werden mit Fachontologien verglichen und den Usern entsprechend angezeigt.
3. User taggen Ressourcen, indem sie angezeigte Terme auswählen oder eigene Terme eingeben.
4. Fachexperten pflegen Ontologien und nutzen dabei die Spuren der Tag-Events.

5 Schlussfolgerungen und Ausblick

Der Umgang mit Bedeutungsaspekten von Information ist eine genuin menschliche Tätigkeit. Die unüberschaubare Informationsmenge in globalen Netzen erfordert aber den Einsatz von maschinellen Verfahren zur Bedeutungsverarbeitung. Von semantischen Technologien können folgende Aktivitäten wirkungsvoll unterstützt werden:

1. Bestimmen von Schlüsselwörtern zu Dokumenten
2. Pflegen von fachspezifischen Begriffssystemen, die ein Grounding von Schlüsselwörtern und Termen ermöglichen
3. Extrahieren von Fakten aus Dokumenten
4. Suchen von Dokumenten zu einer gegebenen Fragestellung

Zu 1): Bei der Schlüsselwortbestimmung lassen sich heute mit vollautomatischen Verfahren schon recht gute Ergebnisse erzielen. Die Suche nach relevanten Dokumenten für menschliche Leser wird damit um einiges effizienter.

Zu 2): Fachspezifische Begriffssysteme (Ontologien) sind Modelle von Wissensbereichen und als solche Hauptresultate wissenschaftlicher Erkenntnis. Maschinelle Informationsverarbeitung dient zum Beispiel der Konsistenzprüfung.

Zu 3): Fakten, die halbautomatisch aus Dokumenten extrahiert werden, können dem Aufbau von Wissensbasen von Expertensystemen dienen. Die Unterstützung durch Softwareagenten erfolgt vorzugsweise so, dass automatisch Vorschläge generiert werden, aus denen der User auswählen kann. Zusätzlich zu dieser Auswahl soll dem User wo sinnvoll immer auch eine freie Eingabe ermöglicht werden.

Zu 4): In Zukunft werden Softwareagenten die Suche nach Dokumenten zu einer gegebenen Fragestellung im Hintergrund und weitgehend autonom durchführen können.

Zwar handelt es sich beim Semantic Web nach wie vor um eine Vision. Mit Text Mining und anderen Sprachtechnologien rückt die Realisierung dieser Vision aber in greifbare Nähe.

6 Literatur

[Antoniou & Harmelen 2008] *Antoniou, G.; Harmelen, F. van:* A Semantic Web Primer. 2nd ed., MIT Press, Cambridge, MA, 2008.

[Blumauer & Hochmeister 2009] *Blumauer, A.; Hochmeister, M.:* Tag-Recommender gestützte Annotation von Web-Dokumenten. In: Blumauer, A.; Pellegrini, T. (Hrsg.): Social Semantic Web. Springer-Verlag, Berlin, 2009, S. 227-243.

[Buitelaar et al. 2005] *Buitelaar, P.; Cimiano, P.; Magnini, B. (Hrsg.):* Ontology Learning from Text: Methods, Evaluation and Applications. IOS Press, 2009.

[Cimiano et al. 2005] *Cimiano, P.; Hotho, A.; Staab, S.:* Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. In: Journal of Artificial Intelligence Research 24, 2005, S. 305-339.

[Evert 2005] *Evert, S.:* The Statistics of Word Cooccurrences: Word Pairs and Collocations. Dissertation, Institut für maschinelle Sprachverarbeitung, University of Stuttgart, 2005.

[Hearst 1992] *Hearst, M.:* Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992, S. 539-545.

[Kaljurand 2008] *Kaljurand, K.:* Attempto Controlled English as a Semantic Web Language. Dissertation. University of Tartu, Estonia, Faculty of Mathematics and Computer Science, Institute of Computer Science, 2008.

[Kaljurand et al. 2009] *Kaljurand, K.; Rinaldi, F.; Kappeler, T.; Schneider, G.:* Using Existing Biomedical Resources to Detect and Ground Terms in Biomedical Literature. In: Proceedings of AIME 2009, Verona, Italy, 2009, S. 225-234.

[Müller et al. 2004] *Müller, H.; Kenny, E.; Sternberg, P.:* Textpresso: An ontology-based information retrieval and extraction system for biological literature. PLoS Biology, 2(11):e309, 09, 2004.

[Rinaldi et al. 2003] *Rinaldi, F.; Kaljurand, K.; Dowdall, J.; Hess, M.:* Breaking the Deadlock. In: Proceedings of ODBASE, 2003 (International Conference on Ontologies, Databases and Applications of SEmantics), Catania, Italy, Springer-Verlag, 2003, S. 876-888.

[Rinaldi et al. 2008] *Rinaldi, F.; Kappeler, T.; Kaljurand, K.; Schneider, G.; Klenner, M.; Clematide, S.; Hess, M.; Allmen, J.; Parisot, P.; Romacker, M.; Vachon, T.:* OntoGene in BioCreative II. Genome Biology, 2008, 9, S. 13.

[Schneider et al. 2009] *Schneider, G.; Kaljurand, K.; Rinaldi, F.:* Detecting Protein-Protein Interactions in Biomedical Texts using a Parser and Linguistic Resources. Best Paper Award (2nd place). In: Proceedings of CICLing 2009, Mexico City. Springer-Verlag, LNC 5449, S. 406-417.

[Schütze 1998] *Schütze, H.:* Automatic Word Sense Discrimination. Computational Linguistics, 24 (1), 1998, S. 97-124.

[Weeds et al. 2005] *Weeds, J.; Dowdall, J.; Schneider, G.; Keller, B.; Weir, D.:* Using Distributional Similarity to Organise BioMedical Terminology. Terminology, 11(1), 2005, S. 3-4.

Dr. Gerold Schneider
Universität Zürich
Institut für Computerlinguistik
Binzmühlestr. 14
CH-8050 Zürich
gschneid@ifi.uzh.ch
www.ifi.uzh.ch

Dr. Heinrich Zimmermann
Fernfachhochschule Schweiz
Pestalozzistr. 33
CH-3600 Thun
heinrich@zimmermann.com
www.fernfachhochschule.ch