Manfred Stede

Textverstehen in der Computerlinguistik am Beispiel der Automatischen Textzusammenfassung

Abstract

Der Beitrag gibt zunächst einen Überblick über die Herangehensweisen der Computerlinguistik an das Automatische Textverstehen seit den Siebziger Jahren. Nach einer psycholinguistisch geprägten Frühphase rückten zunächst die wissensbasierten Ansätze der Künstlichen Intelligenz in den Mittelpunkt des Interesses, die allerdings nicht über den Status kleiner experimenteller Systeme hinaus gelangen konnten. Es folgte eine Hinwendung zur Linguistik mit syntaktischer und semantischer Analyse, bevor um 1990 die „statistische Wende" der Computerlinguistik begann; seither stehen quantitative Verfahren im Vordergrund. Am Beispiel der Aufgabe der Automatischen Textzusammenfassung plädiert der Beitrag dafür, intelligente Verknüpfungen von symbolischen linguistisch-fundierten und statistischen Verfahren zu suchen, um die Robustheit statistischer Verfahren auch durch eine insgesamt höhere Qualität zu verbessern.

1. Textverstehen als Teildisziplin der Künstlichen Intelligenz

Im Jahr 1969 stellten Quilian und Collins ihr Modell der semantischen Netze vor, mit dem sie Unterschiede in der Reaktionszeit von Probanden bei der Präsentation einfacher Sätze dazu nutzten, ein hierarchisches Modell der Wissensspeicherung im menschlichen Gehirn zu skizzieren (Collins/Quilian 1969). Weil beispielsweise der Satz „A canary has skin" von den Probanden langsamer verarbeitet wurde als „A canary can sing", schlossen sie auf die Existenz verschiedener Ebenen im konzeptuellen Wissen, die über *is-a*-Relationen miteinander verbunden sind. Die Attribute der Konzepte sind jeweils auf der allgemeinsten Ebene gespeichert, so dass beispielsweise *has-skin* am Konzept *Animal* notiert ist, das durch *is-a* unter anderem zu *Bird* spezialisiert wird (*can-fly* etc.) und dies wiederum u. a. zu *Canary* – erst hier ist die Information *can-sing* abgelegt, da es andere Vögel gibt, die bekanntlich nicht singen. Um die Aussage „A canary has skin" zu überprüfen, müssen wir also von *Canary* ausgehend zwei Ebenen der konzeptuellen Repräsentation überspringen, um qua Vererbung das Attribut *has-skin* zu finden. Auf diese Weise sei unser gesamtes konzeptuelles Wissen in Form von Vererbungshierarchien organisiert.

Diese Gedächtnismodelle haben in den Siebziger Jahren die Künstliche Intelligenz und mithin die Arbeiten zum Automatischen Textverstehen stark beeinflusst. *Verstehen* wurde aufgefasst als ein Prozess, der in erster Linie

vom bereits gespeicherten Vorwissen beim Textrezipienten gesteuert wird. Die Modelle des *Frame* und des *Script* wurden entwickelt, um einerseits statisches Wissen (strukturierte Konzepte) und andererseits episodisches Wissen (stereotype Abläufe von Handlungen) im Rechner darzustellen. Ein Skript für einen Restaurantbesuch beispielsweise stellt den üblichen Verlauf aus Sicht des Besuchers dar. Das Verstehen eines Textes, der einen konkreten Restaurantbesuch schildert, entspricht dem Aktivieren des entsprechenden Skripts und dem Abgleichen der einzelnen Sätze mit den Einträgen des Skripts. Dabei werden die Platzhalter im allgemeinen Skript durch die im Text genannten Personen und Gegenstände ersetzt, so dass am Ende die Wissensbasis ein instanziiertes Skript enthält – eine Repräsentation der Geschichte, die mit dem Vorwissen verbunden ist und mit der weitere Schlussfolgerungsprozesse angestoßen werden können.

Zur Repräsentation einzelner Sätze folgte man vielfach dem Vorschlag der *conceptual dependency theory* (CD; Schank/Abelson 1977), die eine Dekomposition der Verbbedeutung in ca. 20 semantische Primitiva vorschlug. Die Repräsentation eines kurzen Satzes kann durch die Explizierung enthaltener Kausalität oder Bewegungsverläufe zu einer recht komplexen Graphstruktur anwachsen. Ziel des Ansatzes war eine Repräsentation, die von der sprachlichen Formulierung so stark abstrahiert, dass möglichst viele Paraphrasen eines Satzes auf dieselbe CD-Struktur zurückgeführt werden können – und ebenso Paraphrasen in anderen Sprachen, denn CD wurde als sprachunabhängige Darstellung postuliert.

Modelle dieser Art lassen sich kaum auf ihre kognitive Adäquatheit hin evaluieren, müssen somit spekulativ bleiben. Zudem zeigte sich rasch, dass ihre Implementierungen nicht über das Stadium kleiner experimenteller Systeme hinaus gelangen konnten. Das Programm FRUMP von DeJong (1982) implementierte etwa 40 Skripts und konnte damit einige Arten von Zeitungsmeldungen verarbeiten, machte aber gleichzeitig (wohl unbeabsichtigt) die zentrale Schwäche der Skript-Ansätze deutlich: Um weitere Arten von Meldungen (oder gar völlig andere Textsorten) verarbeiten zu können, müssten immer mehr Skripts von Hand modelliert werden – eine (Teil-)Automatisierung war für diese Aufgabe nicht denkbar. Wie aber soll bei einer großen Menge von Skripts in der Wissensbasis beim Textverstehen noch die Aktivierung des jeweils „richtigen“ Skripts vonstatten gehen?

2. Linguistische Analyse für Textverstehen

Der Gedanke, Automatisches Textverstehen durch Abgleich mit explizitem Vorwissen zu realisieren, wurde um 1980 herum zusehends als Sackgasse erkannt, und man wandte sich stattdessen Modellen zu, die unabhängiger vom Inhalt der zu verstehenden Texte operieren sollten. Im Zentrum stand nicht mehr modelliertes Welt-Wissen, sondern linguistisches Wissen über Wörter und Satzbau: Es wurde eine systematische syntaktische Analyse der einzelnen Sätze

angestrebt, an die sich eine semantische Auswertung anschließen sollte. In einem letzten, pragmatisch motivierten Analyseschritt sollte aus den Satz-Repräsentationen dann eine Textbedeutung hergeleitet werden. Während in der oben dargestellten KI-Phase wichtige Anstöße wie die Idee der Semantischen Netze aus der Psychologie aufgenommen wurden, bildete nun die formale, kompositionale Semantik nach Montague (1974) eine wesentliche Inspiration, auch für solche Ansätze, die ihr nicht im Detail folgten. Montagues Interpretationsverfahren regte eine elegante und systematische Ableitung von Satzsemantik an, ohne auf gespeichertes Vorwissen angewiesen zu sein. Es handelt sich um eine eindeutig nicht-dekompositionelle Theorie: Im Fokus steht die Errechnung der Satzbedeutung aus einzelnen atomaren Wortbedeutungen, und nicht länger das Hinterfragen und Zerlegen der Wortbedeutungen.

Als Beispiel sei hier das bundesdeutsche Projekt LILOG (Linguistische und logische Methoden zum maschinellen Verstehen des Deutschen) genannt, das von IBM und einer Reihe von Universitäten durchgeführt wurde (Herzog/Rollinger 1991). Es betonte die Notwendigkeit der Formalisierung der zugrundeliegenden Wissensrepräsentations- und -verarbeitungsmechanismen; während zuvor in der Frame/Skript-Phase eher hemdsärmelig mit den Repräsentationen verfahren wurde, entwickelte man nun gründliche logische Fundierungen. LILOG verarbeitete touristische Texte (von nicht zu komplizierter Gestalt) und konnte am Ende einige Fragen über die im Text beschriebenen Sachverhalte beantworten. Bevor die wissensverarbeitende Komponente in Aktion tritt (und zum Beispiel versucht, implizite Zusammenhänge zwischen Handlungen zu explizieren, ähnlich der Idee des „Auffüllens" der Story-Repräsentation durch die früheren Skripts), wird eine syntaktische und semantische Analyse vorgenommen. Diese Analyse-Module sollten im Prinzip unabhängig von der Domäne „Touristik" sein, somit wieder verwendbar in anderen Anwendungen.

Im Zuge von LILOG entstand eine Vielzahl gründlicher Untersuchungen von Einzelphänomenen (etwa zur Semantik räumlicher Beschreibungen), gemäß der Annahme, dass qualitativ hochwertiges linguistisches Wissen erforderlich ist, um Textverstehen erfolgreich und portabel zu modellieren. Jedoch stellte sich heraus, – und dies gilt nicht nur für LILOG – dass die seinerzeitigen Analysegrammatiken weder auf Seiten der Syntax noch der Semantik eine hinreichende Abdeckungsbreite besaßen, um „reale" Texte zu verarbeiten. Es ließ sich etwas mehr anfangen als in der Frame/Skript-Phase, und es wurde gründlicher und systematischer gearbeitet, doch wiederum taten sich Grenzen auf, die anscheinend prinzipieller Natur waren.

3. Die „statistische Wende" der Computerlinguistik

In den späten 1980er Jahren entstand in der Computerlinguistik ein neuer „Trend", der sehr schnell zur beherrschenden Strömung der gesamten Disziplin wurde: eine Hinwendung zu Korpus-basierten, statistischen Methoden.

An die Stelle handgeschriebener Regelwerke (wie sie etwa in LILOG mit großem Aufwand entwickelt worden waren) trat die automatische Auswertung von Daten und die Umsetzung der Ergebnisse in probabilistische Modelle. Ausgelöst wurde diese Entwicklung einerseits durch die allgegenwärtige Unzufriedenheit mit den bis dato voherrschenden symbolischen Ansätzen und ihren Skalierungsproblemen; gleichzeitig war man beeindruckt von den enormen Fortschritten, die seinerzeit die automatische Erkennung gesprochener Sprache zu verzeichnen hatte. Diese Programme – etwa die frühen Diktiersysteme – waren frei von handcodierten Regeln und funktionierten allein auf der Grundlage probabilistischer Modelle des Aufeinanderfolgens sprachlicher Zeichen. Angesichts ihres überraschenden Erfolges begann man diese Ansätze zunächst auf syntaktische Analyse, später auch auf weitere Aufgaben zu übertragen. Selbst in der automatischen Übersetzung machte ein vollständig „Linguistik-freies" und einzig auf einer statistischen Auswertung eines zweisprachigen Korpus basierendes Übersetzungsprogramm Furore (Brown et al. 1990).

Das neue Paradigma lässt sich gut an der Aufgabe der *information extraction* illustrieren, die in den USA über viele Jahre Gegenstand eines regelmäßigen Wettstreits partizipierender Computerlinguistik-Arbeitsgruppen war. Diese Wettbewerbe (seinerzeit „message understanding conference" (MUC), später „document understanding conference" u. a.) werden von der DARPA (Defense Advanced Research Projects Agency) organisiert und bilden einen wichtigen Rahmen für die computerlinguistischen Forschungsaktivitäten in den Vereinigten Staaten. Das Ziel der *information extraction* besteht darin, aus Texten eine vorgegebene Menge von Informationen zu entnehmen; Art und Domäne der Texte werden im Vorfeld bekannt gegeben, doch die dem eigentlichen Wettbewerb zugrunde liegenden Texte bleiben bis zuletzt geheim. „Skalierung" und Robustheit sind jetzt also unausweichlich, weil die zu entwickelnde Software letztlich im Wettbewerb auf unbekannten Daten evaluiert wird. Wir betrachten folgenden Beispieltext aus einer frühen MUC-Runde (Grishman 1997):

> „19 March. A bomb went off this morning near a power tower in San Salvador leaving a large part of the city without energy, but no casualties have been reported. According to unofficial sources, the bomb allegedly detonated by urban guerrilla commandos blew up a power tower in the northwestern part of San Salvador at 0650."

Die Aufgabe besteht darin, eine vorgegebene Schablone mit Feldern für „incident type", „date", „location", „perpetrator", „physical target" u. a. aufzufüllen anhand der im Text gefundenen Informationen, also: incident type = bombing, date = March 19, location = San Salvador, perpetrator = urban guerilla commandos, physical target = power tower.

Weil sowohl die Art der Texte (hier: Terror-Nachrichten) als auch die zu füllende Schablone im Voraus bekannt gegeben werden, können die Wettbewerbsteilnehmer für jedes Feld der Schablone ein spezifisches Programm

entwerfen, das nach genau dieser Information im Text sucht. Diese Unterscheidung ist wichtig: Es wird kein „allgemeines Textverstehen" angestrebt, sondern lediglich die gezielte Suche nach konkreten Informationen eines bekannten Typs in einer bekannten Textsorte. Für *incident type* beispielsweise würde man in solchen Zeitungsmeldungen erwarten, dass es genau der Sachverhalt ist, der im ersten Satz der Meldung berichtet wird. Da die Menge möglicher *incident types* zudem recht begrenzt ist, kann ein Programm sich zielgenau auf die Identifikation einstellen. Die Gesamtaufgabe des *message understanding* wurde dann im Zuge der MUC-Runden zusehends aufgespalten in verschiedene Teilaufgaben wie die sog. *named-entity recognition*, also die Erkennung von Eigennamen von Personen oder Institutionen. Jede Teilaufgabe konnte einzeln genau quantitativ evaluiert werden, und der Wettbewerb besteht jedes Mal darin, die Erkennungswerte zu verbessern, wenn auch nur minimal.

Die Betonung des Evaluationsgedankens war ein wesentlicher positiver Aspekt der statistischen Wende: Es reichte nicht mehr aus, allein per Introspektion Repräsentationen und Verarbeitungsprozesse linguistischer Information vorzuschlagen, sondern es entwickelte sich ein Zwang zur quantitativen Auswertung der eigenen Arbeit und zum Nachweis einer Verbesserung gegenüber dem bisherigen Stand der Kunst. Dies wurde und wird bisweilen ein wenig auf die Spitze getrieben, doch insgesamt hat die Disziplin von der gewachsenen Rigorosität fraglos profitiert.

4. Automatische Textzusammenfassung

Wenden wir uns nun einer der Anwendungen der Computerlinguistik etwas detaillierter zu: der Automatischen Textzusammenfassung. Hier muss man – um gar nicht erste falsche Vorstellungen zu wecken – grundsätzlich unterscheiden zwischen den Verfahren *extracting* und *abstracting*: Ersteres bezeichnet das Auswählen von relevant erscheinenden Sätzen des Originaltextes, die dann einfach hintereinander gereiht werden; *abstracting* umfasst eine tiefere Analyse des Textes und die aktive Generierung der Zusammenfassung, die also zumindest teilweise aus den „eigenen Worten" des Programms entsteht. Die heute kommerziell erhältlichen Systeme betreiben ausnahmslos *extracting*, während *abstracting* in einigen Forschungsprototypen realisiert ist.[1]

Die Grundidee des *extracting* geht auf Luhn (1958) zurück: Man bildet zunächst eine Liste der im Text auftretenden Inhaltswörter (mit Hilfe einer Stoppwortliste der Funktionswörter) und zählt deren Vorkommen, was zu einer Rangliste führt. Sodann bestimmt man für jeden Satz, wie viele Wörter aus hohen Positionen der Rangliste er enthält und weist ihm ein entsprechen-

[1] Einen Überblick über die Aufgaben der Automatischen Zusammenfassung gibt beispielsweise Endres-Niggemeyer (2004).

des Gewicht zu. Nachdem jeder Satz gewichtet ist, werden die am höchsten bewerteten Sätze ausgewählt und aneinandergereiht – wie viele, hängt davon ab, wie lang die Zusammenfassung werden soll, was die Benutzerin des Programms vorab einstellen kann.

Dieses Verfahren bildet bis heute die Grundlage von *extraction*-basierter Zusammenfassung, wenngleich die Gewichtung der Wörter und Sätze durchaus etwas ausgefeilter verlaufen kann. So kann man mittels sogenannter TF/IDF-Verfahren (*term frequency / inverted document frequency*) bestimmen, wie häufig ein Wort im Text relativ zu einer Menge von Vergleichstexten ist. Nehmen wir an, der zusammenzufassende Text sei eine Bundestagsrede und es liege ein Korpus von solchen Reden vor. Dann ergibt die Auszählung der Wörter der fraglichen Rede womöglich, dass *Bundeskanzler* zu den häufigsten Wörtern zählt – allerdings wird es in den meisten anderen Reden ebenso häufig sein und ist insofern nicht sonderlich charakteristisch für die fragliche Rede. Ist jedoch das Wort *Gentechnik* unter den häufigsten, wird sich herausstellen, dass es gemittelt über alle anderen Reden eher selten ist – und insofern als besonders relevant für die fragliche Rede gelten darf. Sätze, die *Gentechnik* enthalten, erhalten dann eine höhere Extraktionswahrscheinlichkeit als solche, die *Bundeskanzler* enthalten.

Ein grundsätzliches Problem der *extraction*, unabhängig von der Ausgefeiltheit der Relevanz-Maße, besteht in potenziellen Kohärenzproblemen der Zusammenfassung. Kurz gesagt stellt jeder anaphorische Ausdruck, der auf den Ko-Text verweist, eine mögliche Gefahr dar, denn es kann ja sein, dass der Satz, der das Antezedens enthält, nicht zu den für relevant befundenen zählt und somit nicht Teil der Zusammenfassung ist. Dann entstehen im günstigen Fall sinnlose Sätze (kein Antezedens erkennbar), im ungünstigen Fall irreführende (es ist für den Leser ein Antezedens in der Zusammenfassung erkennbar, jedoch ist es falsch). Dies kann nicht nur bei Pronomina, sondern beispielsweise auch bei Komparativen geschehen. Beispiel: „In Deutschland sind fünf Millionen Menschen arbeitslos. Noch höher war der Anstieg in Ostdeutschland." Zwischen diesen beiden Sätzen befand sich im Originaltext das Attribut, auf das sich *höher* bezieht, doch der es enthaltende Satz wurde leider nicht extrahiert. Manche Programme antworten auf diese Probleme damit, dass Sätze, die Pronomina enthalten, aus Zusammenfassungen einfach wieder entfernt werden – dies fördert zwar die Kohärenz der Zusammenfassung, kann aber natürlich der Leserin wichtige inhaltliche Information vorenthalten.

Der unbestreitbare Vorteil der Extraktionsverfahren liegt in ihrer hohen Robustheit: Sie funktionieren auf jedwedem Text, völlig unabhängig von der Textsorte und der inhaltlichen Domäne. Die Qualität allerdings stößt an prinzipielle Grenzen; Relevanz oder Wichtigkeit von Textabschnitten bemessen sich nicht allein an der Frequenz von verwendeten Wörtern.

5. *SUMMaR:* Ein „hybrides" Verfahren zur Zusammenfassung

An der Universität Potsdam wird seit Ende 2004 im Projekt „SUMMaR" ein Verfahren zur Automatischen Textzusammenfassung entwickelt, das die Vorzüge der eben skizzierten statistischen Verfahren mit denen einer partiellen linguistischen Analyse zu verbinden versucht.[2] An zwei Stellen soll versucht werden, über die Vorgehensweise der gängigen statistischen Verfahren hinaus zu gehen: Zum einen tritt neben die TF/IDF-basierte Ermittlung der Relevanz von Sätzen ein systematisch repräsentiertes und eingesetztes *Textsortenwissen*, das dem Programm ebenfalls Hinweise auf potenziell wichtige Textstellen liefert. Zum anderen soll die Kohärenz der Zusammenfassung verbessert werden, indem anaphorische Verweise nicht einfach getilgt, sondern nach Möglichkeit resolviert werden und der Text entsprechend angepasst wird. Das System soll am Ende eine *multi-document summarization* leisten, also aus einer Menge von Texten zum selben Thema eine gemeinsame, vergleichende Zusammenfassung erstellen, wir befassen uns in diesem Beitrag allerdings nur mit der Bearbeitung einzelner Dokumente. Zur Illustration dienen Texte aus dem WWW, in denen Kunden ihre Meinung zu Produkten/Dienstleistungen kundtun oder auch ausführliche Besprechungen liefern. Beispielsweise gibt es eine Website, die Kundenmeinungen zu Hotels sammelt. Wenn ein Internet-Surfer sich für ein bestimmtes Hotel interessiert, kann er oder sie die Meinungen derjenigen einsehen, die bereits dort waren; sie werden zunächst als Liste aller Beiträge, jeweils mit den ersten 2–3 Sätzen präsentiert, und man kann dann per Mausklick jeweils den kompletten Text ansehen. Leider sind aber die ersten 2–3 Sätze einer solchen Besprechung nicht immer sehr aussagekräftig, insbesondere muss darin die letztendliche Beurteilung keineswegs erkennbar sein. Wesentlich illustrativer wäre statt der Anfangssätze eine knappe Zusammenfassung, die die wesentlichen Aussagen und das Gesamturteil auf den Punkt bringt. (Eine ganz ähnliche Situation findet sich bei Online-Buchhändlern, die Rezensionen von Käufern desselben Buches anbieten, oder auch allgemein bei Internet-Suchmaschinen, die ebenfalls zu den gefundenen Seiten einen knappen, möglichst charakteristischen Ausschnitt anzubieten versuchen.)

Um die Rolle von TF/IDF Verfahren ein wenig zu verdeutlichen, sei hier ein kleines Experiment geschildert. Dazu wurden 250 Hotel-Besprechungen von einer Internet-Website kopiert und zu einer einzelnen Textdatei gebündelt. Gemäß dem im vorigen Abschnitt geschilderten Verfahren soll versucht werden, in diesem Korpus diejenigen Wörter zu finden, die charakteristisch für Hotel-Besprechungen bzw. -beschreibungen sind. Dazu genügt es nicht, ein-

[2] Siehe http://www.ling.uni-potsdam.de/~acl-lab. Gefördert vom Bundesministerium für Bildung und Forschung im Rahmen des Programms „Innovative regionale Wachstumskerne".

fach die häufigsten Wörter zu zählen, sonst wäre vermutlich auch ein relativ unspezifisches Wort wie *Zimmer* unter den hochfrequenten. Steht jedoch ein großes, hinsichtlich der Textsorte möglichst breit gestreutes Referenzkorpus zur Verfügung (das wir für unseren Zweck als repräsentatives Abbild des deutschen Sprachgebrauchs auffassen), so kann die *relative* Häufigkeit von Wörtern ermittelt werden – also diejenigen Wörter gefunden werden, deren Anteil an der Wortmenge in den Hotel-Texten erheblich größer ist als ihr Anteil am Referenzkorpus. Wir haben als Referenzkorpus das DWDS-Korpus[3] eingesetzt und mit einer einfachen Berechnung eine geordnete Liste erstellt, deren Anfang hier genannt sei:

> *Minibar, Fazit, Supermarkt, Aerobic, Sauna, Tischtennis, Albatros, Gartenanlage, Salate, Luxor, Bungalows, Strand, Balkon, Boccia …*

Dies soll lediglich illustrieren, wie bei Verfügbarkeit großer Datenmengen mit sehr einfachen Mitteln ein textsortenspezifisches Vokabular gewonnen werden kann. SUMMaR setzt die TF/IDF-Berechnung ebenfalls zur Bestimmung relevanter Textstellen ein, ergänzt dies jedoch mit Wissen über *Textsorten*, was im Folgenden kurz erläutert wird.

Eine Analyse des Korpus von Hotel-Besprechungen zeigt, dass die große Mehrzahl der Texte ein „Gesamturteil" enthält, das typischerweise am Ende, mitunter ganz am Anfang, jedoch so gut wie nie in der Textmitte auftaucht. Es ist in der Regel durch Formulierungen wie „Insgesamt lässt sich sagen, dass …", „Zusammenfassend finde ich …" etc. explizit gekennzeichnet. Neben dem Gesamturteil sind Einzelaspekte von Interesse, die dem Verfasser des Beitrags besonders wichtig erschienen. Diese können überall im Text genannt sein und sind wiederum lexikalisch markiert: „Hervorzuheben ist …", „Besonders unerfreulich war …" etc. Diese Beobachtungen gelten unabhängig von der Domäne „Hotel" für Texte der Sorte „Produktbesprechung" – insbesondere für solche aus dem Internet, denn in Print-Publikationen gelten in der Regel engere Konventionen für Textaufbau und Formulierung.

Anliegen von SUMMaR ist es, exemplarisch für eine Reihe von Textsorten dieses Wissen über typische Dokumentstruktur und über Schlüssel-Phrasen formal zu repräsentieren und bei der Analyse mit dem zusammenzufassenden Text abzugleichen. Neben der TF/IDF-Berechnung soll also auch das Wissen darüber, wo sich üblicherweise wichtige Textstellen befinden und wie sie markiert sind, in die Relevanz-Bewertung der Sätze einfließen. Wir versprechen uns davon einen deutlichen Qualitätsgewinn, sind es doch gerade die kurzen, aber bedeutungsvollen Sätze wie „Insgesamt bin ich sehr zufrieden", die keine Chance haben, bei einem standardisierten TF/IDF-Verfahren als „relevant" eingestuft zu werden – der vorstehende Beispielsatz enthält keinen Hotel-

[3] http://www.dwds.de. Das Korpus umfasst 100 Millionen Wörter aus verschiedenen, breit gestreuten Textquellen des 20. Jahrhunderts. Herzlichen Dank an Dr. Alexander Geyken für die Bereitstellung der Zählergebnisse.

spezifischen Begriff, ist für die Textsorte Produkt*besprechung* jedoch von enormer Wichtigkeit. Das Wissen über Textsorten wird innerhalb von SUMMaR als XML-Schemata dargestellt, die die Reihenfolge von erforderlichen und optionalen Textabschnitten (und übliche Varianten) beschreiben und in den jeweiligen Abschnitten darüber hinaus eine möglichst umfassende Menge von Schlüsselphrasen (in einer morphologisch reduzierten Form als reguläre Ausdrücke, die mit dem zusammenzufassenden Text abgeglichen werden können).

Die zweite wesentliche Verbesserung, die wir in SUMMaR anstreben, betrifft die partielle Re-Generierung von Sätzen oder Satzteilen, somit soll über reine *extraction* hinaus ein kleiner Schritt in Richtung *abstracting* versucht werden. Unser Augenmerk gilt dabei in der ersten Phase den anaphorischen Pronomen. Wie oben geschildert, kann ein Satz wie „Er hat mir insgesamt nicht gut gefallen", der vom System womöglich als relevant und somit als Teil der Zusammenfassung eingestuft wird, großen Schaden anrichten, wenn das Antezedens nicht in der Zusammenfassung enthalten ist. Im Rahmen einer robusten linguistischen Analyse unterzieht SUMMaR den zusammenzufassenden Text einem *part-of-speech tagging* und einer partiellen syntaktischen Analyse (sog. *chunk parsing*), bei der vor allem Nominalphrasen identifiziert werden. Unter Ausnutzung heuristischer Algorithmen wird dann versucht, – ohne jedes inhaltliche Textverständnis – den auftretenden Pronomen jeweils ein Antezedens zuzuordnen, mithin die „referentiellen Ketten" des Textes zu bestimmen, bzw. diejenigen Teile, die pronominal gebildet werden. (Für definite Nominalphrasen, insbesondere sog. *bridging anaphors*, ist eine heuristische Resolution deutlich schwieriger.) Steht diese Hintergrundinformation bereit, kann für ein Pronomen in der Zusammenfassung zunächst festgestellt werden, ob sein wahrscheinliches Antezedens bereits genannt und vermutlich auch als solches erkennbar ist – die referentiellen Ketten werden für die Zusammenfassung auf der Basis der Analyse des Ausgangstexts separat erstellt. Fehlt ein Antezedens und ist es im Ausgangstext bekannt, so kann es in vielen Fällen in den entsprechenden Satz eingesetzt werden; im obigen Beispiel wäre das Resultat vielleicht „Der Urlaub hat mir insgesamt nicht gefallen." Dieses Verfahren wird keineswegs perfekt sein, da nur eine oberflächennahe Analyse des Textes stattfindet, doch wenn die einfacheren Probleme von *extractions* auf diesem Weg gelöst werden können, versprechen wir uns davon bereits einen erkennbaren Qualitätsgewinn für die Zusammenfassungen.

Neben der Koreferenz-Analyse versucht SUMMaR, – insbesondere für vorwiegend argumentative Texte – durch eine Untersuchung der im Text verwendeten Konnektoren die *rhetorische Struktur* (im Sinne von Mann/Thompson 1988) partiell zu rekonstruieren. Auch sie kann wichtige Hinweise auf unterschiedliche Relevanz von Textabschnitten liefern. Wiederum ein kurzes Beispiel: „Obwohl das Wasser etwas kalt war und die Anlage reichlich besucht war, habe ich den Pool sehr genossen." Die Kombination aus dem Konnektor *obwohl* und der Satzstruktur lässt darauf schließen, dass „habe ich den Pool

sehr genossen“ einen wichtigen Abschnitt darstellt, da der Konnektor eine Diskursrelation *Concession* signalisiert, deren *Nukleus* stets das größere kommunikative Gewicht trägt. Im Falle von *obwohl* entspricht der Nukleus dem Matrixsatz, der somit bei der Relevanzberechnung einen entsprechenden Bonus bekommt. Dieses „rhetorische Parsing“, wiederum allein auf der Grundlage oberflächennaher Informationen, ohne semantische Analyse, hat in den letzten Jahren große Aufmerksamkeit erfahren, es liegen interessante Arbeiten vor allem für englische Texte vor (z. B. Marcu 2000). Für das Deutsche scheint uns eine gründliche Analyse der Gebrauchsbedingungen von Konnektoren unabdingbar, insbesondere eine Klassifikation der Oberflächenmerkmale ihrer Kontexte. Dies ist oftmals bereits für die Desambiguierung erforderlich, etwa um festzustellen, ob *da* als unterordnende Konjunktion (und damit als Signal einer kausalen Diskursrelation) oder als Partikel verwendet wird, oder ob *schließlich* eine temporale oder eine begründende Verwendung findet (vgl. dazu auch den Beitrag von Breindl/Waßner in diesem Band). Dies lässt sich durch Oberflächenanalyse keineswegs immer entscheiden, in vielen Fällen aber doch, und dann ist die Information für die weitere Analyse und schließlich für die Konstruktion der Zusammenfassung möglicherweise sehr wertvoll.

6. Fazit

In diesem Beitrag habe ich versucht, einen kurzen und damit zwangsläufig auch verkürzenden Überblick über die Ansätze zum Textverstehen in der Computerlinguistik der letzten 35 Jahre zu geben – es ging mir darum, die aus meiner Sicht vorherrschenden Strömungen zu identifizieren. In der ersten Phase (grob: die 70er) wurde die Rolle des nicht-linguistischen Vorwissens (also des „Welt-Wissens“) für die Textverarbeitung betont und es wurde versucht, dieses explizit im Computer zu modellieren; mehr als kleine Experimente lassen sich auf diesem Weg allerdings nicht realisieren, weil das dazu notwendige Wissen nicht automatisch gewonnen werden kann, sondern „von Hand“ aufgebaut werden muss. Für praktische Anwendungen ist ein solches Vorgehen freilich indiskutabel. Die anschließende Phase (grob: die 80er) war vom Bemühen geprägt, mit systematischer linguistischer Analyse die Abhängigkeit von modelliertem Hintergrundwissen zu beseitigen oder abzumildern und gleichzeitig „portable“ Lösungen zu entwickeln – ein syntaktischer Parser sollte sich ja eigentlich für ganz verschiedene Texte gleichermaßen verwenden lassen. Allerdings erwiesen sich formale Grammatiken seinerzeit als nicht mächtig genug, um mit „realen“ Texten umzugehen, und die semantische Analyse stieß ebenfalls auf grundlegende Skalierungsprobleme, denen auch die wissensbasierten Ansätze bereits begegnet waren. Es folgte ein Wechsel zur statistischen Phase (grob: seit 1990), die mit zunehmender Hinwendung zu automatischen Lernverfahren bis heute anhält. Es stehen oberflächennahe Analyseverfahren im Vordergrund, die oftmals durch annotierte

Daten trainiert sind und probabilistische Ergebnisse, etwa für die Syntax-Analyse, liefern. Dies lässt sich lesen als eine fortschreitende Bewegung weg von Semantik und Wissen hin zu reiner Mustererkennung – getrieben von den oftmals im Vordergrund stehenden Anwendungserfordernissen in der Computerlinguistik, die eben Robustheit und Abdeckungsbreite zum zentralen Faktor machen.

In jüngster Zeit mehren sich die Anzeichen dafür, dass das Pendel ein wenig zurückschlägt. Beispielsweise werden unter dem Stichwort *Semantic Web* verstärkt Fragen des Einsatzes, der Gewinnung und der Pflege von Ontologien (grob: Formalisierung von Domänenwissen) diskutiert, auch in Verbindung mit sprachverstehenden Anwendungen. Des weiteren ist bei den Verarbeitungsstrategien eine Hinwendung zu „hybriden" Verfahren erkennbar, die (möglicherweise von Hand geschriebene) symbolische Regeln mit statistisch erworbenem Wissen zu verbinden versuchen. Eine Ausprägung eines solchen Ansatzes habe ich mit dem Potsdamer SUMMaR-Projekt für die Anwendung der Automatischen Textzusammenfassung vorgestellt. Auch dieses Projekt ist letztlich primär durch Anwendungserfordernisse charakterisiert, bemüht sich jedoch um die Entwicklung von Lösungen, die auch für andere Anwendungen im Prinzip einsetzbar sind. Plakativ gesagt soll beispielsweise unser modelliertes Textsortenwissen so allgemein sein, dass es nicht nur für Zusammenfassungen einsetzbar ist. Das allgemeine Anliegen ist, eine Grenze zu ziehen zwischen linguistischer Information, die sich dem Text bzw. den Sätzen entnehmen lässt, und kontextabhängiger Information, die von der Interpretation oder von der Domäne abhängt. Wenn die oberflächennahe Auswertung anwendungsneutral gestaltet werden kann, ist ein großer Schritt in Richtung Portabilität getan. Ein zentrales technisches Erfordernis dabei ist die *Unterspezifikation* von Information: Die Analyse wird häufig kein eindeutiges Ergebnis erzielen, sondern vielleicht eine Reihe von Möglichkeiten offen lassen wollen (beispielsweise alternative PP-Anbindungen in der Syntax, alternative Anapher-Antezedenten in der Diskurs-Interpretation). Wenn diese systematisch repräsentiert werden, kann eine unterspezifizierte Struktur die Schnittstelle zu weiteren Analysemodulen bilden, die möglicherweise über Domänenwissen, über pragmatische Heuristiken und dergleichen verfügen, um letztlich die Entscheidung für *eine* PP-Anbindung, für *ein* Antezedens herbeizuführen (sofern das für die gestellte Aufgabe überhaupt erforderlich ist). Textverstehen wäre dann ein zweistufiger Prozess: Robuste und portable linguistische Analyse erzeugt unterspezifizierte Zwischenrepräsentation, domänen- oder anwendungsspezifische Komponenten können darauf aufbauend weitere Auswertungsschritte vornehmen. Aus anwendungsnaher Sicht scheint dies eine sinnvolle Perspektive – ob es freilich in irgendeiner Form auch ein *kognitiv* adäquates Modell sein könnte, steht auf einem ganz anderen Blatt.

Literatur

Breindl, Eva/Waßner, Ulrich H. (2005): Syndese vs. Asyndese. Konnektoren und andere Wegweiser für die Interpretation semantischer Relationen in Texten. In diesem Band.

Brown, Peter F. et al. (1990): A Statistical Approach to Machine Translation. In: Computational Linguistics 16(2), S. 79–85.

Collins, Allan M./Quilian, M. Ross (1969): Retrieval time from semantic memory. In: Journal of Verbal Learning and Verbal Behavior 8, S. 240–247.

De Jong, Gerald (1982): An over view of the FRUMP system. In: Lehnert, Wendy G./ Ringle, Mark (eds): Strategies for natural language processing. Hillsdale, NJ: Erlbaum. S. 149–172.

Endres-Niggemeyer, Brigitte (2004): Automatisches Textzusammenfassen. In: Lobin, Henning/Lemnitzer, Lothar (Hg.): Texttechnologie – Perspektiven und Anwendungen. Tübingen: Stauffenburg. S. 407–432.

Grishman, Ralph (1997): Information extraction: Techniques and Challenges. In: Pozienza, Maria Teresa (ed.): Information Extraction, a multidisciplinary approach to an emerging information technology. Berlin/Heidelberg: Springer. S. 10–27.

Herzog, Otthein/Rollinger, Claus-Rainer (eds.) (1991): Text Understanding in LILOG. Integrating Computational Linguistics and Artificial Intelligence. Berlin/Heidelberg: Springer.

Luhn, Hans Peter (1958): The automatic creation of literature abstracts. In: IBM Journal of Research and Development 2(2), S. 159–165.

Mann, William C./Thompson, Sandra A. (1988): Rhetorical Structure Theory: Toward a Functional Theory of text organization. In: Text 8, S. 243–281.

Marcu, Daniel (2000): The theory and practice of dicourse parsing and summarization. Cambridge/MA: MIT Press.

Montague, Richard (1974): The Proper Treatment of Quantification in Ordinary English. In: Thomason, Richmond (ed.): Formal Philosophy: Selected Papers of Richard Montague. Yale University Press.

Schank, Roger C./Abelson, Robert P. (1977): Scripts, Plans, Goals and Understanding. Hillsdale, NJ: Erlbaum.

Anhang:

Aktuelle Projekte in Textlinguistik und Textverstehensforschung