

**Universität Leipzig**  
Fakultät für Mathematik und Informatik  
Institut für Informatik

Wortbedeutungsdisambiguierung mittels  
sprachunabhängiger, konkurrenzbasierter Algorithmen

**Bachelorarbeit**

Leipzig, November 2007

vorgelegt von  
Reuter, Sven  
geb. am 09.02.1981  
Studiengang Informatik

**Betreuender Hochschullehrer:** Prof. Dr. habil. Uwe Quasthoff

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b> .....	<b>iv</b>
<b>Tabellenverzeichnis</b> .....	<b>iv</b>
<b>1. Einleitung</b> .....	<b>1</b>
1.1 Begriffsklärung.....	1
1.1.1 Ambiguität.....	1
1.1.2 Disambiguierung .....	3
1.1.3 Kookkurrenzen.....	3
1.2 Einschränkungen .....	4
1.3 Zusammenhänge in der Sprache .....	4
1.4 Ansätze zur Disambiguierung .....	7
<b>2. Algorithmus</b> .....	<b>8</b>
2.1 Voraussetzungen .....	8
2.1.1 WSI-Algorithmus.....	8
2.1.2 WSI-Evaluation.....	9
2.2 Verfahren.....	9
2.3 Schwellwert.....	14
2.4 Fazit.....	14
<b>3. Evaluation</b> .....	<b>16</b>
3.1 Evaluationsverfahren.....	17
3.1.1 Evaluation anhand des line-hard-serve Korpus .....	17
3.1.2 Senseval-Evaluation .....	17
3.1.3 Evaluation mittels Pseudowörtern.....	18
3.1.4 Durchführung der Evaluation .....	19
3.2 Obere und untere Grenze.....	20
3.3 Einflussnahmen und Schwellwerte .....	22
3.3.1 Beeinflussung durch Kookkurrenzen .....	22
3.3.2 Beeinflussung durch den Satz .....	24

3.3.3	Beeinflussung durch Wort- und Frequenzklasse.....	27
3.3.4	Beeinflussung durch den Schwellwert .....	28
3.4	Vergleich zu bestehenden Verfahren .....	28
3.5	Fazit.....	29
<b>4.</b>	<b>Ansätze zur Optimierung .....</b>	<b>31</b>
4.1	Automatische Optimierung und Kombination von Konfigurationen .....	31
4.2	Linke und rechte Nachbarn .....	31
4.3	Sprach- und syntaxabhängige Optimierungen .....	32
<b>5.</b>	<b>Fazit .....</b>	<b>33</b>
	<b>Literaturverzeichnis.....</b>	<b>34</b>

## Abbildungsverzeichnis

Abbildung 1.1	Der Zusammenhang zwischen Rang $r$ , Frequenz $f$ und deren Produkt $r * f$ im BNC. ....	5
Abbildung 1.2	Zusammenhang zwischen Rang $r$ und Anzahl der unterschiedenen Bedeutungen $m$ im BNC. ....	6
Abbildung 1.3	Zusammenhang zwischen Rang $r$ und Größe der Kookkurrenzliste $c$ im BNC. ....	7
Abbildung 2.1	Disambiguierung am Beispiel: Bedeutungen und Kookkurrenzen. In der ersten Zeile befindet sich der Originalsatz (ohne Satzendezeichen), in der zweiten die Bedeutungen bzw. das Wort selbst noch einmal und in der dritten Zeile die fünf stärksten Kookkurrenzen. ....	10
Abbildung 3.1	Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit von der maximalen Größe der Kookkurrenzliste. ....	23
Abbildung 3.2	Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit von der Position des Pseudoworts im Satz. ....	25
Abbildung 3.3	Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit von der Satzlänge. ....	26
Abbildung 3.4	Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit vom Stoppwortanteil im Satz. ....	26
Abbildung 3.5	Die <i>precision</i> in Abhängigkeit vom prozentualen Unterschied der Bewertungen. ....	28

## Tabellenverzeichnis

Tabelle 2.1	Exemplarische Analyse von <i>space_0</i> . ....	13
Tabelle 2.2	Exemplarische Analyse von <i>space_1</i> . ....	13
Tabelle 3.1	Evaluationsergebnisse für die deutsche und englische Sprache. ....	20
Tabelle 3.2	<i>Baselines</i> für die Evaluation des WSD-Algorithmus. <i>Recall</i> beträgt in allen Fällen 100%. ....	21
Tabelle 3.3	Die <i>precision</i> in Abhängigkeit von der Frequenzklasse des Quellworts (QW) und Mergeworts (MW) in englischer (links) und deutscher Sprache (rechts). 27	
Tabelle 3.4	Die <i>precision</i> in Abhängigkeit von der Wortklasse des Quellwortes (QW) und Mergewortes (MW) in englischer (links) und deutscher Sprache (rechts). ....	27
Tabelle 3.5	Gegenüberstellung von <i>Senseval</i> -Ergebnissen und Evaluationsergebnissen dieses Algorithmus. ....	29

## 1. Einleitung

Disambiguierung ist noch immer ein schwieriges und wichtiges Thema in der Verarbeitung natürlicher Sprache. Seit dem wissenschaftlichen Interesse an maschineller Übersetzung und künstlicher Intelligenz beschäftigen sich Wissenschaftler mit diesem Gebiet (Agirre & Edmonds, 2006) mit unterschiedlichsten Ansätzen und Ergebnissen. Neben den genannten sind automatische Thesauri, Extraktion von Informationen und Information Retrieval weitere Einsatzgebiete von Disambiguierungsergebnissen.<sup>1</sup>

Diese Arbeit soll einen Ansatz präsentieren, der auf einer neuartigen Wortbedeutungsunterscheidung (Bordag, 2006) basiert. Dabei werden verschiedene Einflüsse analysiert, wie beispielsweise Satzlänge, Stoppwörter und Wortklasse. Als Grundlage dient der British National Corpus (BNC), eine Textsammlung, die mit 100 Mio. laufenden Wörtern einen repräsentativen Querschnitt des britischen Englischs des späten 20. Jahrhunderts darstellt und ein breites Themenspektrum abdeckt. (University of Oxford, [bnc] About the British National Corpus: What is the BNC?, 2007) Die englische Sprache wurde gewählt, um einen besseren Vergleich zu bestehenden und vorausgesetzten Verfahren zu ermöglichen.

Für die Vergleichssprache Deutsch werden Daten verwendet, die von der Abteilung Automatische Sprachverarbeitung des Instituts für Informatik der Universität Leipzig zur Verfügung gestellt wurden. Diese sind etwa doppelt so groß<sup>2</sup>, ebenfalls breit gefächert, enthalten jedoch keine gesprochene Sprache wie der BNC.

### 1.1 Begriffsklärung

#### 1.1.1 Ambiguität

In Helmut Glücks Lexikon wird Ambiguität wie folgt definiert:

„**Ambiguität** (lat. *ambiguitās* ›Doppelsinn‹ ... Auch: Ambivalenz, Amphibolie, Mehrdeutigkeit, Vieldeutigkeit, semant. Unbestimmtheit) Typ semant. Unbestimmtheit eines Zeichens, der in Abgrenzung zur Vagheit<sup>3</sup> dadurch charakterisiert werden kann, daß für ein und dieselbe Zeichenform mehrere miteinander konkurrierende Interpretationen feststellbar sind.“ (Glück, 2005, S. 35)

---

<sup>1</sup> In Anlehnung an (Bordag, 2006) und (Edmonds, Lexical disambiguation, 2006).

<sup>2</sup> Der BNC umfasst 6 Mio., die deutschen Daten 10 Mio. Sätze.

<sup>3</sup> „pragmatische Unbestimmtheit“ (Bußmann, 2002)

Es werden dort weiter drei Untergruppen<sup>4</sup> beschrieben:

- Lexikalische Ambiguität
- Syntaktische Ambiguität
- Skopusambiguität

Für die Wortbedeutungsdisambiguierung ist jedoch nur die erste Gruppe, die lexikalische Ambiguität, von Interesse.<sup>5</sup> Zu ihr zählen die Homonymie und die Polysemie.<sup>6</sup> Polyseme Ausdrücke haben im Gegensatz zu homonymen eine gemeinsame Wurzel; eine strikte Trennung ist jedoch nicht immer möglich (Glück, 2005). Beispiele für Polysemie sind: *Brücke* als Bauwerk, rhetorische Überleitung oder Zahnersatz; *Geist* als Intellekt oder als übernatürliches Wesen. Die Unterschiede in den Bedeutungen sind jedoch bei Homonymen durch ihre unterschiedliche Herkunft oft stärker und zeichnen sich häufig durch verschiedene Genera (*der/die Kiefer*, *das/die Mark*, *der/das Tau*, aber: *Das Tau* als Seil und als griechischer Buchstabe ist auch eine Homonymie.) oder Pluralbildung (*die Mütter/Muttern*) aus. Eine Spezialform der Homonymie ist die Homografie, bei der die Aussprache bei gleicher Schreibweise unterschiedlich ist (*modern* als fortschrittlich oder als verwesend, *Montage* von Montieren oder als Plural von Montag). Sie werden im Lexikon unter verschiedenen Einträgen aufgeführt und haben meist auch unterschiedliche Übersetzungen. Die Homophonie (gleiche Aussprache bei unterschiedlicher Schreibweise: *mehr/Meer*, *heute/Häute*) als weiterer Spezialfall ist allerdings eher ein Problem bei der Spracherkennung als bei der automatischen Disambiguierung. Carstensen et al. (2004) ergänzt Mehrdeutigkeiten bei Wörtern noch um Metonymie („Verschiebung der begrifflichen Interpretation“; Beispiel: „Die Firma rief an.“) und Metaphern („nicht-wörtliche Rede“).

Edmonds ordnet die Ambiguitäten in eine Hierarchie von grobkörnig bis feinkörnig ein (Edmonds, *Lexical disambiguation*, 2006):

- Part-of-speech (Wortklasse)
- Homografie
- Polysemie
- Reguläre Polysemie
- Wortgebrauch
- Fester Ausdruck

---

<sup>4</sup> Teilweise wird auch eine vierte Gruppe benannt: relationale Ambiguität (Bußmann, 2002).

<sup>5</sup> Für die Definitionen der anderen Gruppen vgl. vertiefend (Bußmann, 2002) oder (Glück, 2005).

<sup>6</sup> Die folgenden Definitionen und Beispiele in diesem Abschnitt stützen sich weitgehend auf Bußmann, 2002. Agirre und Edmonds beschreiben Polysemie als eine Eigenschaft von Wörtern und Ambiguität von Texten. (Agirre & Edmonds, 2006)

### 1.1.2 Disambiguierung

Disambiguierung ist die „Beseitigung lexikal. oder struktureller Mehrdeutigkeit (Ambiguität, ...) durch den außersprachl. und sprachl. Kontext“ (Glück, 2005). Strukturelle Mehrdeutigkeiten werden „durch explizite Ausformulierung der zugrunde liegenden Strukturen“ (Bußmann, 2002) disambiguiert:

„So sind die beiden Lesearten des Satzes *Die Wahl der Vorsitzenden fand Zustimmung* zu disambiguieren durch die Paraphrasen P<sub>1</sub> *Dass die Vorsitzende gewählt wurde, fand Zustimmung* bzw. P<sub>2</sub> *Die Wahl, die die Vorsitzende getroffen hat, fand Zustimmung.*“ (Bußmann, 2002, S. 169)

Außersprachlicher Kontext ist dort weiter beschrieben als Sprechsituation, Vorwissen, Erwartungshaltung, Einstellung, aber auch Gestik und Mimik.

Die Wortbedeutungsdisambiguierung<sup>7</sup> beschränkt sich auf die lexikalische Mehrdeutigkeit im sprachlichen Kontext (Manning & Schütze, 1999), obwohl unter Umständen auch Vorwissen, zum Beispiel das Wissen über das Themengebiet oder über vorherige Disambiguierungen im Text, genutzt werden kann.

### 1.1.3 Kookkurrenzen

Kookkurrenz und Kollokation sind Begriffe mit ähnlichen Bedeutungen, was dazu führt, dass Autoren diese mitunter gleichbedeutend verwenden (Manning & Schütze, 1999). Beide Begriffe sind ein Mittel für die Annahme des gemeinsamen Vorkommens von zum Beispiel Wörtern. Mit Kollokationen sind allerdings konkret Wortverbindungen gemeint, die primär eine semantische Beziehung haben (*Hund : bellen, dunkel : Nacht*) und in bestimmter, naher Anordnung (lat. *collocatio* ›Anordnung‹) vorkommen, wie es zum Beispiel bei linken und rechten Nachbarn der Fall ist. Kookkurrenz dagegen bezeichnet das gemeinsame Vorkommen und damit eine gewisse Abhängigkeit in einem größeren Kontext, wie zum Beispiel Sätzen oder Dokumenten. (Manning & Schütze, 1999; Bußmann, 2002) Weiterhin wird bei Satzkookkurrenzen<sup>8</sup> die Struktur und Ordnung im Satz ignoriert (*bag-of-words model*). (Bordag, 2006; Manning & Schütze, 1999)

Mit statistischen Tests kann die Signifikanz (ein Maß für die Abhängigkeit) bestimmt werden, wobei das Miteinandervorkommen mit dem alleinigen Auftreten der beiden Wörter ins Verhältnis

<sup>7</sup> In der englischsprachigen Literatur *Word Sense Disambiguation (WSD)* (Bordag, 2006; Agirre & Edmonds, 2006) oder auch kurz *disambiguation* (Manning & Schütze, 1999). Im Deutschen auch: *Leseartendisambiguierung* (Carstensen, Ebert, Endriss, Jekat, Klabunde, & Langer, 2004). Im Folgenden wird *Disambiguierung* gleichbedeutend mit *Wortbedeutungsdisambiguierung* benutzt.

<sup>8</sup> Im Folgenden gleichbedeutend mit *Kookkurrenzen* verwendet.

gesetzt wird. Für die verwendeten Daten wurde das log-likelihood-Maß verwendet (Dunning, 1993); Alternativen sind zum Beispiel Mutual Information oder der t-Test.<sup>9</sup> In dem Disambiguierungsverfahren wird die Signifikanz selbst nicht verwendet, sondern lediglich zur Sortierung der Kookkurrenzen genutzt.

## 1.2 Einschränkungen

Neben dem Vernachlässigen von Ordnung und Struktur bei den zugrunde liegenden Daten und den Kookkurrenzen werden im gesamten Verfahren des Disambiguierungsalgorithmus auch die Wortklasse (*Part-of-Speech (PoS)*), der Sprachbau, die Valenz (Wertigkeit von Verben) und das Wissen um die Wort-Wortform-Beziehungen (Flexion, Komposition, Derivation) ignoriert. So wird zum Beispiel das Paar *Zwerg : Zwerge* so unterschiedlich aufgefasst wie *Zwerg : Computer*.

Dies ermöglicht einen sprach- und syntaxunabhängigen und daher universellen Einsatz sowie eine vollkommen automatische Verarbeitung, aber unter Umständen gehen damit Qualitätseinbußen einher. Ein weiterer Vorteil ist, dass die betrachteten Einheiten (Wortformen) ebenso komplexere Ausdrücke sein können, wie zum Beispiel Wortgruppen.

## 1.3 Zusammenhänge in der Sprache

George K. Zipf (Zipf, 1965) beschrieb Zusammenhänge in der Sprache bezüglich des Ranges eines Wortes. So zeigte er, dass die Frequenz  $f$  eines Wortes in einem Text umgekehrt proportional zu seinem Rang<sup>10</sup>  $r$  ist:

$$f \sim \frac{1}{r} \quad (1.1)$$

$$f * r = k_f \quad (1.2)$$

Dabei charakterisiert das Produkt  $k_f$  aus Rang und Frequenz eine textabhängige Konstante, welche beim BNC im Gesamtdurchschnitt bei etwa 1,0 Mio. liegt, jedoch zwischen 12 Mio. und 150 Tsd. schwankt. Besonders markant zeigt sich das an dem Abfallen zwischen Rang 10.000 und 100.000 (Abbildung 1.1). Der Grund für diese Schwankungen ist, dass im BNC mehr als 4000, mitunter thematisch sehr verschiedene, Texte vereint sind (University of Oxford, [bnc] About the British National Corpus: The BNC in numbers). Dies führt dazu, dass vor allem die Gruppe der niederfrequenten Wörter überrepräsentiert ist.

<sup>9</sup> Siehe dazu vertiefend (Manning & Schütze, 1999).

<sup>10</sup> Der Rang ergibt sich aus der fortlaufenden Nummerierung der nach Frequenz absteigend sortierten Wortliste.



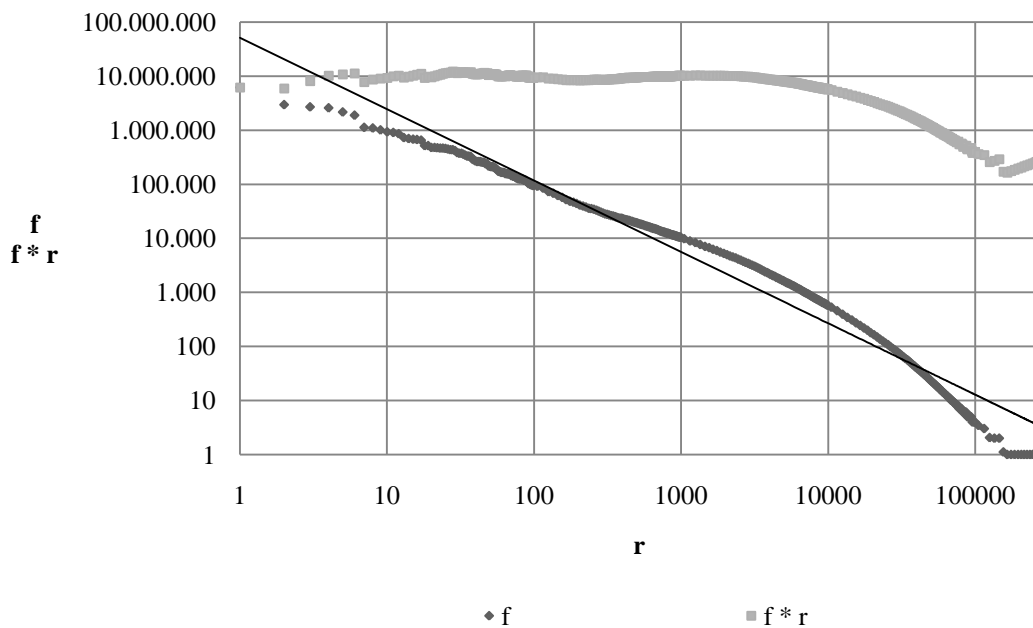


Abbildung 1.1 Der Zusammenhang zwischen Rang  $r$ , Frequenz  $f$  und deren Produkt  $r * f$  im BNC.

Wie in Abbildung 1.1 zu sehen ist, kommen etwa 75% der betrachteten Wortformen<sup>11</sup> höchstens zehn Mal im 100 Mio. Wörter umfassenden BNC vor. 42% aller Wortformen sind im BNC einmalig.

Eine andere Gesetzmäßigkeit, die Zipf (Zipf, 1965) erläuterte, ist die des Zusammenhanges zwischen dem Rang und Bedeutungen in einem Text. Demnach ist das Produkt aus der Anzahl der Bedeutungen eines Wortes  $m$  und der Wurzel aus seinem Rang  $r$  konstant ( $k_m$ ):

$$m \sim \frac{1}{\sqrt{r}} \tag{1.3}$$

$$m * \sqrt{r} = k_m \tag{1.4}$$

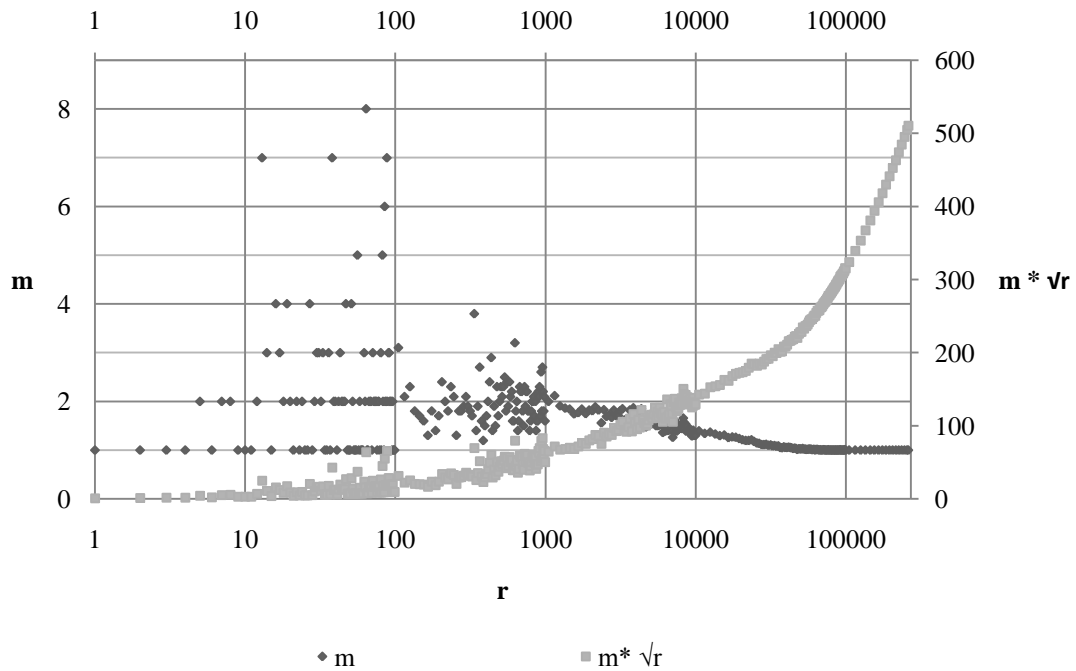
Verallgemeinernd lässt sich Formel 1.4 mit einer weiteren Konstanten  $l$  wie folgt darstellen (entspricht der Formel 1.4 mit  $l = 0,5$ ):

$$m * r^l = k_m \tag{1.5}$$

In Abbildung 1.2 ist die Verteilung der Bedeutungen in der auf den BNC basierenden Wortliste, die im Algorithmus verwendet wird, veranschaulicht. Das Produkt  $k_m$  ist hier nicht konstant, sondern monoton steigend. Das ist darin begründet, dass sich Zipfs Annahme auf einen einzelnen

<sup>11</sup> Dargestellt werden hier Wortformen, die sowohl in der Frequenzliste (Kilgarriff, 1996), als auch in der verwendeten Wortliste vorhanden sind (insgesamt etwa 260 Tsd.).

Text bezieht und nicht ohne Weiteres auf ganze Korpora übertragbar ist<sup>12</sup> und dass der WSI-Algorithmus nicht alle Bedeutungen der Wörter identifiziert<sup>13</sup>. Zu erwarten wären gerade im hochfrequenten Bereich mehr Bedeutungen: Zipf fand in seiner Studie für Wörter um den Rang 2000 im Schnitt 4,6 Bedeutungen anhand eines Wörterbuches (Zipf, 1965); hier sind es etwa 1,8.

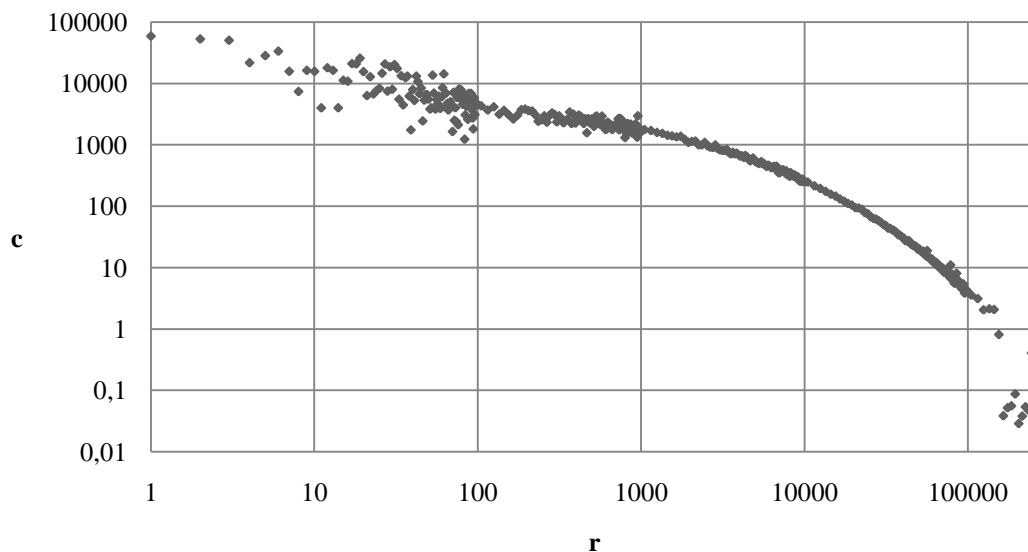


**Abbildung 1.2** Zusammenhang zwischen Rang  $r$  und Anzahl der unterschiedenen Bedeutungen  $m$  im BNC.

Wichtig ist neben der Verteilung der Bedeutungen in der verwendeten Wortliste auch die Größe der Kookkurrenzlisten, die quasi das Wissen für die Disambiguierung bereitstellen:

<sup>12</sup> Philip Edmonds bestätigte diese Annahme jedoch für den BNC für  $l = 0,404$  (Formel 1.5) anhand der WordNet 2.0 Bedeutungen (<http://wordnet.princeton.edu/>) von Wortgrundformen mit Abweichungen in den oberen und unteren Rängen. (Edmonds, Lexical disambiguation, 2006) Die Verteilung der Bedeutungen aus dem WSI-Algorithmus ist für  $l = 0,2$  etwa konstant, jedoch ebenfalls mit Abweichungen in den Randbereichen.

<sup>13</sup> Siehe dazu auch Abschnitt 2.1.2.



**Abbildung 1.3** Zusammenhang zwischen Rang  $r$  und Größe der Kookkurrenzliste  $c$  im BNC.

Wie in Abbildung 1.3 zu sehen, ist auch die Größe der Kookkurrenzliste mit steigendem Rang abfallend. Auffällig ist, dass etwa 60% der Wortformen weniger als fünf Kookkurrenzen haben.

Die Auswirkungen der betrachteten Verteilungen und Zusammenhänge auf den Algorithmus wird in Kapitel drei erläutert.

#### 1.4 Ansätze zur Disambiguierung

Disambiguierungsmethoden werden meist anhand ihrer verwendeten Quellen für die Wortbedeutungsunterscheidung klassifiziert. Wörterbuchbasierende Ansätze fassen Algorithmen zusammen, die auf Wörterbüchern (auch Übersetzungswörterbücher), Thesauri oder Lexika basieren, dafür aber keine Korpusdaten benutzen. Bei den automatischen („unsupervised“) Methoden, zu denen auch der vorgestellte Algorithmus gehört, ist es umgekehrt: Sie nutzen keine externen Quellen, sondern arbeiten nur mit den Rohdaten aus dem Korpus, zum Beispiel per Clustering oder anhand eines gleichen Korpus in einer anderen Sprache. Als dritte Möglichkeit gibt es überwachte („supervised“) Algorithmen, bei der annotierte Trainingsdaten verwendet werden. Algorithmen, die mit einer sehr kleinen, handverlesenen Trainingsmenge auskommen und selbstlernend sind, werden halb überwacht („semi-supervised“) genannt. Die Ansätze werden in manchen Algorithmen auch kombiniert. (Agirre & Edmonds, 2006; Manning & Schütze, 1999)

Evaluationen, wie Senseval-2 und -3, haben gezeigt, dass die überwachten Methoden die beste Performanz aufweisen. (Agirre & Edmonds, 2006)

## 2. Algorithmus

### 2.1 Voraussetzungen

Um Mehrdeutigkeiten aufzulösen, muss primär das Wissen über mögliche Bedeutungen vorhanden sein. Dieses Wissen kann zum Beispiel aus Wörterbüchern oder Trainingsmengen entnommen werden (Manning & Schütze, 1999).

In diesem Fall werden Wort- und Kookkurrenzlisten verwendet, die der WSI<sup>14</sup>-Algorithmus von Stefan Bordag produziert. Dieser Algorithmus und die Evaluation sind in seinem Dokument „Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation“ (Bordag, 2006) beschrieben und werden in den folgenden zwei Abschnitten zusammenfassend erklärt.

#### 2.1.1 WSI-Algorithmus

Stefan Bordags Algorithmus ist automatisch („unsupervised“) und basiert auf der „one sense per collocation“<sup>15</sup>-Beobachtung, arbeitet aber mit Worttripeln statt Wortpaaren. Für ein Zielwort wird solch ein Tripel aus dem Wort selbst und jeder möglichen Zweierkombination der 200 stärksten Satzkoookkurrenzen (nach Signifikanz; in Dreißigergruppen) gebildet. Es wird dann die Schnittmenge aus den Kookkurrenzen der Tripelwörter gebildet, welche als Eigenschaft („feature“) des Tripels für die folgende Clusteranalyse benutzt wird. Es wird davon ausgegangen, dass eine große Schnittmenge ein Anzeichen dafür ist, dass dieses Tripel ein eindeutiges Thema beschreibt. Sowohl in den Kookkurrenzen, aus denen die Tripel gebildet werden, als auch aus den Kookkurrenzen für die Schnittmengen werden die Stoppwörter entfernt, da diese trotz großer Schnittmenge meist kein eindeutiges Thema beschreiben.

In der Clusteranalyse werden Tripel eines Zielworts zusammengefasst, dessen Eigenschaften zueinander hinreichend ähnlich sind (über 80% gleicher Elemente). Durch einen weiteren Clusterschritt werden Cluster zusammengeführt, die ähnliche Schlüsselmengeten (die aus den Tripelwörtern beim Clustern entstehen) haben. Dies eliminiert überflüssige Bedeutungsunterscheidungen.

Als letzte Schritte werden Wörter den bestehenden Clustern zugeordnet, die in keinem geclusterten Tripel vorkamen und Cluster mit weniger als acht Wörtern entfernt. Die verbleibenden Cluster repräsentieren die unterschiedenen Bedeutungen des Zielworts, welche dann, im Falle der Mehrdeutigkeit, mit fortlaufenden Suffixen anstelle des Zielworts abgelegt werden.

---

<sup>14</sup> Word Sense Induction: Wortbedeutungsunterscheidung

<sup>15</sup> engl. für ›Eine Bedeutung pro Kollokation‹

So wurden beispielsweise für das Wort *space* zwei Bedeutungen gefunden (eine im Sinne von Weltraum, eine für Platz), welche unter *space\_0* und *space\_1* in der Wortliste und mit den entsprechenden Kookkurrenzen in der Kookkurrenzdatei abgelegt wurden. Das Wort *space* kommt in der resultierenden Wortliste nicht mehr vor.

### 2.1.2 WSI-Evaluation

Schütze (Manning & Schütze, 1999)<sup>16</sup> erläuterte eine Evaluierungsmethode mithilfe von Pseudowörtern. Diese Evaluationsmethode kommt sowohl für den WSD-Algorithmus zum Einsatz, als auch, in ähnlicher Form, beim WSI-Algorithmus.<sup>17</sup>

Der Algorithmus erreichte, angewandt auf den BNC, eine Präzision von 85,42% bei einer Vollständigkeit von 72,90%. Dies entspricht einem F-Maß ( $\alpha = 0,5$ ) von 78,66% und liegt damit etwa 6% über dem F-Maß von einem Durchlauf mit Wortpaaren statt Worttripeln.<sup>18</sup>

In einem weiteren Durchgang wurden als Clusterfeatures lediglich direkte Nachbarwörter benutzt, dessen Evaluationsergebnis mit einem F-Maß von 47,10% unter dem mit Satz-kookkurrenzen liegt.

Bordag zeigte an Beispielen, dass zwar nicht immer alle Bedeutungen eines Wortes gefunden werden, die gefundenen jedoch intuitiv sind. Dies muss kein Nachteil sein, denn es kommt auf den Einsatzzweck an, für den der Algorithmus verwendet wird. Zum Beispiel die Mehrdeutigkeit des englischen Wortes *mouse* (mit seinen Bedeutungen als Tier und Eingabegerät): Diese Mehrdeutigkeit ist für eine maschinelle Übersetzung ins Deutsche unerheblich, denn die Übersetzung ist in beiden Fällen *Maus*. Dagegen ist dieser Unterschied im Gebiet von Information Retrieval von Relevanz. (Agirre & Edmonds, 2006)

## 2.2 Verfahren

Die vom Disambiguierungsverfahren betrachtete Einheit ist ein Satz, in wenigen Fällen auch ein kurzer Abschnitt. Diese Einheit wird am Leerzeichen in die einzelnen Wörter gesplittet. Existiert ein Wort in der Wortliste, so ist es nicht ambig. Kommt es mit dem Suffix *\_0* vor, dann ist es mehrdeutig, und es können iterativ alle Bedeutungen entnommen werden (mit Suffixen *\_0*, *\_1*, *\_2*, usw.) bis die Wort-Suffix-Kombination nicht mehr in der Wortliste vorkommt. Enthält die

<sup>16</sup> Ursprünglich beschrieben in: Schütze, H. (1992). Context space. (R. Goldman, P. Norvig, E. Charniak, & B. Gale, Hrsg.) *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, 113–120.

<sup>17</sup> Siehe dazu auch Abschnitt 3.1.

<sup>18</sup> Zu Präzision (auch: *precision*), Vollständigkeit (auch: *recall*) und F-Maß (auch *F-measure*) siehe auch Kapitel drei.

Wortliste weder das Wort selbst noch mit dem Suffix *\_0*, dann ist es der Wortliste unbekannt.<sup>19</sup>

Im darauffolgenden Schritt werden zu den gefundenen Wörtern und Bedeutungen die 900 signifikantesten Kookkurrenzen als geordnete Liste aus der Kookkurrenzdatei geholt. Am Beispiel stellt sich das dann mit realen Daten und maximal fünf Kookkurrenzen pro Wort wie folgt dar:

<i>An</i>	<i>astronaut</i>	<i>flew</i>	<i>the</i>	<i>space</i>		<i>shuttle</i>
<i>An</i>	<i>astronaut</i>	<i>flew</i>	<i>the</i>	<i>space_0</i>	<i>space_1</i>	<i>shuttle</i>
( <i>example</i> )	( <i>space</i> )	( <i>aircraft</i> )	( <i>between</i> )	( <i>time</i> )	( <i>storage</i> )	( <i>space</i> )
( <i>%N%</i> )	( <i>hole</i> )	( <i>back</i> )	( <i>same</i> )	( <i>shuttle</i> )	( <i>into</i> )	( <i>mission</i> )
( <i>alternative</i> )	( <i>black</i> )	( <i>into</i> )	( <i>er</i> )	( <i>which</i> )	( <i>parking</i> )	( <i>NASA</i> )
( <i>by</i> )	( <i>capsule</i> )	( <i>plane</i> )	( <i>end</i> )	( <i>between</i> )	( <i>outer</i> )	( <i>Atlantis</i> )
( <i>important</i> )	( <i>singularity</i> )	( <i>over</i> )	( <i>most</i> )	( <i>search</i> )	( <i>floor</i> )	( <i>Challenger</i> )

**Abbildung 2.1** Disambiguierung am Beispiel: Bedeutungen und Kookkurrenzen. In der ersten Zeile befindet sich der Originalsatz (ohne Satzzeichen), in der zweiten die Bedeutungen bzw. das Wort selbst noch einmal und in der dritten Zeile die fünf stärksten Kookkurrenzen.

Das *%N%* in den Kookkurrenzen von *An* ist ein Repräsentant für alle Zahlen. Weiter ist zu sehen, dass in den Kookkurrenzen mehrdeutige Wörter nur in ihrer „Grundform“ (ohne Suffix) vorkommen (siehe Kookkurrenz *space* für *astronaut*).

Nachdem die benötigten Informationen verfügbar sind, wird mit der Analyse begonnen. Dabei wird für jedes mehrdeutige Wort die passendste Bedeutung anhand der besten Bewertung bestimmt. Die Bewertung kann mit mehreren Faktoren gewichtet werden. Jeder Bedeutung des Zielworts  $w_i$  (im Beispiel ist das Zielwort *space* mit den Bedeutungen *space\_0* und *space\_1*) wird nacheinander jedes andere Satzwort  $w_k$  ins Verhältnis gesetzt und anfangs überprüft, ob das Wort selbst in den Kookkurrenzen der betrachteten Bedeutung auftritt. Ist das der Fall, dann wird der inverse Rang<sup>20</sup> von  $w_k$  in der Kookkurrenzliste der betrachteten Bedeutung mit einem Ranggewicht und einem Faktor für die Gewichtung für das Vorkommen von Satzwörtern in den Kookkurrenzen multipliziert. Ist das Ranggewicht zum Beispiel eins, so entspricht die Bewertung für einen Treffer dem inversen Rang (für Position null: fünf, Position eins: vier, usw.). Ist das Ranggewicht zwei, dann ist die Bewertung das Doppelte des inversen Ranges (für Position null: zehn, Position eins: acht, usw.). Das Ranggewicht beträgt sowohl bei dem realen Beispiel als auch bei der späteren Evaluation 50, die Gewichtung von Vorkommen von Satzwörtern in den Kookkurrenzen zehn.

Im nächsten Schritt wird geprüft, wie viele gemeinsame Elemente die Kookkurrenzen von dem Satzwort und der betrachteten Bedeutung besitzen und die Anzahl mit einem Gewicht (das eins beträgt) für gemeinsame Kookkurrenzen multipliziert. Das heißt, dass Vorkommen von Satz-

<sup>19</sup> Intern werden die Wortstrings aus Performanzgründen als eindeutige Nummern weiterverarbeitet, die aus der Wortliste entnommen werden.

<sup>20</sup> Der inverse Rang ergibt sich aus der maximalen Anzahl der Kookkurrenzen minus der Position in der Kookkurrenzliste (bei null beginnend).

wörtern in den Kookkurrenzen zehn Mal stärker bewertet werden (zuzüglich der Bewertung für den Rang) als gemeinsame Kookkurrenzen.

Im letzten Schritt wird die bisherige Bewertung der betrachteten Bedeutung bezüglich des aktuellen Satzworts mit der Distanz zwischen den beiden Elementen im Satz gewichtet. Dafür wird der Abstand zwischen den Elementen im Satz berechnet und von der Satzlänge subtrahiert (der direkte Nachbar bekommt also die höchste Bewertung, die weiter entfernten Wörter eine niedrigere). Diese Differenz wird mit einem Faktor für die Distanzgewichtung multipliziert, und das Produkt wird potenziert mit einem zweiten Gewicht für die Distanz. Das Ergebnis daraus wird letztendlich mit der bisherigen Bewertung der betrachteten Bedeutung bezüglich des aktuellen Satzworts multipliziert. Grund für diesen Faktor und Exponent ist die Annahme, dass näherstehende Wörter die Bedeutung mehr beeinflussen als entferntere. Mit dem Exponenten lässt sich statt eines linearen Verlaufs ebenfalls ein quadratischer, kubischer, usw. darstellen. Der Faktor beträgt im realen Beispiel fünf und der Exponent vier. Die Teilbewertung wird dann der Gesamtbewertung der Bedeutung hinzugefügt.

Zusammenfassend lässt sich der Disambiguierungsalgorithmus wie folgt mit Pseudocode beschreiben:

```

Für jedes Satzwort
  Wenn Satzwort bekannt dann
    Hole Kookkurrenzen zu Satzwort
  Sonst
    Wenn (Satzwort + „_0“) bekannt dann
      Hole Kookkurrenzen zu (Satzwort + „_0“)
      Zähler = 1
      Solange (Satzwort + „_“ + Zähler) bekannt
        Hole Kookkurrenzen für (Satzwort + „_“ + Zähler)
        inkrementiere Zähler

Für jedes bekannte Satzwort  $w_i$ 
  Wenn  $w_i$  mehrdeutig dann
    Für jede Bedeutung  $b_j$  von  $w_i$ 
       $p_j = 0$  // Bewertung für Bedeutung  $b_j$ 
      Für jedes bekannte Satzwort  $w_k$  außer  $w_i$ 
         $p_{jk} = 0$  // Bewertung für Bedeutung  $b_j$  bzgl.  $w_k$ 
        Wenn  $w_k$  in den Kookkurrenzen  $k_j$  von  $b_j$  vorkommt dann
           $f_{rg} =$  Ranggewicht (50)
           $f_{skg} =$  Satz-Kookkurrenz-Gewicht (10)
          Wenn  $f_{rg} > 0$  dann
             $p_{jk} +=$  inverser Rang von  $w_k$  in  $k_j$  *  $f_{rg}$  *  $f_{skg}$ 
          Sonst
             $p_{jk} += f_{skg}$ 
        Für jede Bedeutung  $b_1$  von  $w_k$ 
           $m = |k_j \cap k_1|$  // Größe der Schnittmenge,  $k_1 \dots$  Kookk. v.  $b_1$ 
           $f_{kkg} =$  Kookkurrenz-Schnittmengen-Gewicht (1)
           $p_{jk} += m * f_{kkg}$ 
           $f_{dg} =$  Distanz-Gewicht (5)
           $f_{dpg} =$  Distanz-Potenz-Gewicht (4)
          Wenn  $f_{dg} > 0$  dann
             $n =$  Satzlänge - (Abstand zwischen  $w_i$  und  $w_k$ )
             $p_{jk} += p_{jk} * (f_{dg} * n)^{f_{dpg}}$ 
           $p_j += p_{jk}$ 
      Bestimme passendste Bedeutung anhand der höchsten Bewertung und
      ↪ ersetze sie im Output-Satz

```

Zur Veranschaulichung wird der Beispielsatz aus Abbildung 2.1 analysiert: Das einzige mehrdeutige Wort in diesem Satz ist *space*, und es wird mit der Bewertung von *space\_0* begonnen:



Aktuelles Wort	Beschreibung	Algorithmus	
		Bewertung	Gesamt-bewertung
<i>An</i>	- kommt selbst nicht in Kookkurrenzen von <i>space_0</i>	0	0
<i>astronaut</i>	vor	0	0
<i>flew</i>	- hat keine gemeinsamen Kookkurrenzen mit <i>space_0</i>	0	0
<i>the</i>	- kommt selbst nicht in Kookkurrenzen von <i>space_0</i> vor - hat eine gemeinsame Kookkurrenz mit <i>space_0</i> ( <i>between</i> ) – Bewertung mit eins - Satzlänge sechs minus Abstand eins = fünf - bisherige Bewertung bzgl. <i>the</i> (eins) wird mit $(5*5)^4 = 390625$ multipliziert	390625	390625
<i>shuttle</i>	- kommt selbst auf Rang eins (inverser Rang vier = Faktor 200 ( $4 * 50$ )) in Kookkurrenzen von <i>space_0</i> vor - diese Bewertung wird mit zehn gewichtet (= 2000) - hat keine gemeinsamen Kookkurrenzen mit <i>space_0</i> - Satzlänge sechs minus Abstand eins = fünf - bisherige Bewertung bzgl. <i>shuttle</i> (2000) wird mit $(5*5)^4 = 390625$ multipliziert	781250000	781640625

Tabelle 2.1 Exemplarische Analyse von *space\_0*.

Danach wird die Bedeutung *space\_1* analysiert:

Aktuelles Wort	Beschreibung	Algorithmus	
		Bewertung	Gesamt-bewertung
<i>An</i>	- kommt selbst nicht in Kookkurrenzen von <i>space_1</i>	0	0
<i>astronaut</i>	vor	0	0
<i>flew</i>	- hat keine gemeinsamen Kookkurrenzen mit <i>space_1</i>		
<i>flew</i>	- kommt selbst nicht in Kookkurrenzen von <i>space_1</i> vor - hat eine gemeinsame Kookkurrenz mit <i>space_1</i> ( <i>into</i> ) – Bewertung mit eins - Satzlänge sechs minus Abstand zwei = vier - bisherige Bewertung bzgl. <i>flew</i> (eins) wird mit $(4*5)^4 = 160000$ multipliziert	160000	160000
<i>the</i>	- kommt selbst nicht in Kookkurrenzen von <i>space_1</i>	0	160000
<i>shuttle</i>	vor	0	160000
<i>shuttle</i>	- hat keine gemeinsamen Kookkurrenzen mit <i>space_1</i>		

Tabelle 2.2 Exemplarische Analyse von *space\_1*.

Die Bedeutung *space\_0* erhält die höhere Gesamtbewertung von 781.640.625, *space\_1* lediglich 160.000; im Output-Satz wird also die Bedeutung *space\_0* gesetzt. Diese Entscheidung ist

richtig und kann mit einem prozentualen Unterschied der Bewertungen von 99,98% als sehr sicher angesehen werden.

### 2.3 Schwellwert

Es ist bei Disambiguierungsalgorithmen nicht üblich, Schwellwerte anzugeben oder zu implementieren, da es zum Beispiel im Bereich von Information Retrieval und maschineller Übersetzung keinen Sinn macht, weil für gewöhnlich eine Entscheidung für die vermeintlich richtige Bedeutung getroffen werden muss. Geht es jedoch beispielsweise um das Finden von Beispielsätzen für ein Wörterbuch, um den Gebrauch der verschiedenen Bedeutungen zu verdeutlichen, ist es sinnvoll, dort nur Sätze zu verwenden, in denen die Entscheidung über die richtige Bedeutung möglichst zuverlässig war. Dies kann erreicht werden, indem ein Mindestmaß für den prozentualen Unterschied der besten zur zweitbesten Bewertung implementiert wird und die Entscheidung nur dann getroffen wird, wenn das Mindestmaß überschritten ist, also als sicher angesehen werden kann. Ebenso können die zu verarbeitenden Sätze nach Satzlänge und Stoppwortanteil gefiltert werden. Inwiefern die Leistung dadurch verbessert werden kann, wird in Kapitel drei betrachtet.

### 2.4 Fazit

Der Disambiguierungsalgorithmus wurde inkrementell in Java implementiert. In der ersten Stufe wurden nur die Vorkommen der Satz Wörter in den Kookkurrenzen der Bedeutung beachtet, in der zweiten dann auch die Schnittmengen der Kookkurrenzen. Zu diesem Zeitpunkt wurde ebenfalls die automatische Evaluation eingebunden, die auch nach und nach verbessert wurde, um detailliertere Ergebnisse zu liefern. In Erwartung höherer Qualität wurden daraufhin Filter beziehungsweise Parameter eingeführt. So ist es beispielsweise möglich, die Stoppwörter aus den Kookkurrenzen vor der Analyse zu entfernen oder die Kookkurrenzlisten auf eine einheitliche Länge im Satz zu kürzen.

Ein ausschlaggebender Einfluss bei der Entwicklung war neben der Qualität auch der Zeitfaktor: Bei der Analyse sollen etwa 50 Sätze pro Minute bearbeitet werden können. Um in diesen Bereich zu gelangen, hat es bei der Implementierung einige Optimierungen von Datentypen und -strukturen erfordert, um die Verarbeitung zu beschleunigen. Weiterhin wurde die objektorientierte Struktur wieder etwas abstrahiert; die spezialisierteste instanziierte Klasse ist der Satz, in der die Klassen für Wörter und Bedeutungen weitgehend zusammengeführt wurden.

Ein alternativer Ansatz zur Analyse prüft Satzvariationen. Die Variationen werden mittels der möglichen Bedeutungen der ambigen Wörter gebildet, und statt einzelner Bedeutungen wird die komplette Variation bewertet. So werden für den Beispielsatz aus Abbildung 2.1 zwei Sätze

analysiert und bewertet:

- *An astronaut flew the space\_0 shuttle*
- *An astronaut flew the space\_1 shuttle*

Die Variation mit der höheren Bewertung wird als die richtige erkannt. Dieser Ansatz hat den Vorteil, dass Bedeutungen auch direkt aufeinander wirken können. Der Nachteil und zugleich der Grund, dass dieser Ansatz nicht weiterverfolgt wurde, ist der Zeitbedarf. Eine Analyse für eine Satzvariation ist etwa annähernd aufwendig wie für einen kompletten Satz beim verwendeten Algorithmus, und die Anzahl an Variationen kann sich bei langen Sätzen (etwa 30 Wörter) auf über 1 Mio. belaufen.

### 3. Evaluation

Der Zweck der Evaluation ist, die Qualität des Algorithmus zu testen und dabei möglicherweise Erweiterungs- und Verbesserungsmöglichkeiten zu finden. Während eine manuelle Evaluation, also die menschlich-intuitive Beurteilung der Korrektheit von Ergebnissen, eine einfache und schnelle Lösung darstellt, kann eine automatische Evaluation dagegen reproduzierbare Ergebnisse liefern (Bordag, 2006).

Allen Evaluationsmethoden ist gemein, dass sie die gleichen Grundgrößen benutzen, um die Qualität zu errechnen<sup>21</sup>:

- *Anzahl (n)*: die Gesamtzahl der Evaluationsobjekte
- *True positive (tp)* oder *richtig positiv*: Das Ergebnis wurde ausgewählt und wurde richtig ersetzt.
- *False positive (fp)* oder *falsch positiv*: Das Ergebnis wurde ausgewählt, aber falsch ersetzt.
- *True negative (tn)* oder *richtig negativ*: Das Ergebnis wurde falsch ersetzt und nicht ausgewählt.
- *False negative (fn)* oder *falsch negativ*: Das Ergebnis wurde nicht ausgewählt, aber richtig ersetzt.

Die Qualität von Disambiguierungsalgorithmen wird häufig mit *accuracy* (Genauigkeit, Treffgenauigkeit) angegeben<sup>22</sup> und ergibt sich aus dem prozentualen Anteil der richtigen Entscheidungen:

$$\mathit{accuracy} = \frac{tp+tn}{n} \quad (3.1)$$

Eine weitere Möglichkeit, das Maß an Qualität auszudrücken, kommt aus dem Bereich des Information Retrieval und ermöglicht es, Schwellwerte zu berücksichtigen: *precision* und *recall*. *Precision* (Genauigkeit, Präzision) ist ein Maß dafür, dass ein ausgewähltes Element vom System richtig disambiguiert wurde:

$$\mathit{precision} = \frac{tp}{tp+fp} \quad (3.2)$$

---

<sup>21</sup> Folgende Definitionen und Formeln zu *accuracy*, *precision*, *recall* und *F-measure* stützen sich auf (Manning & Schütze, 1999).

<sup>22</sup> Siehe (Edmonds & Kilgarriff, Introduction to the Special Issue on Evaluating Word Sense Disambiguation Systems, 2002), (Patwardhan, Banerjee, & Pedersen, 2007) und (Brody, Navigli, & Lapata, 2006).

*Recall* (Abruf, Wiederaufruf) beschreibt den Anteil der richtig disambiguierten Elemente, die ausgewählt wurden:

$$(3.3) \quad recall = \frac{tp}{tp+fn}$$

Werden alle Elemente ausgewählt (kein Schwellwert), dann beträgt *recall* 100% und *precision* entspricht *accuracy*.

Das *F-measure* ist eine Kombination aus *recall* und *precision* und wie folgt definiert:

$$F = \frac{1}{\alpha \frac{1}{precision} + (1-\alpha) \frac{1}{recall}} \quad (3.4)$$

wobei  $\alpha$  ein Faktor zum Gewichten von *precision* und *recall* ist. Gewöhnlich wird für  $\alpha$  ein Wert von 0,5 gewählt, der die beiden Maße gleich gewichtet. Die resultierende Formel für *F* ist:

$$F = \frac{2 * precision * recall}{precision + recall} \quad (3.5)$$

### 3.1 Evaluationsverfahren

Nicht nur die Angabe der Qualität eines WSD-Algorithmus ist unterschiedlich, sondern auch die zugrunde liegenden Evaluationsverfahren. Die drei meistgenutzten Verfahren sind *line-hard-serve*, *Senseval* und das pseudowortbasierende; sie gehören alle zu den *in vitro*<sup>23</sup> Verfahren.

#### 3.1.1 Evaluation anhand des line-hard-serve Korpus

Die Evaluation mit den Wörtern *line*, *hard*, *serve* (und zum Teil auch *interest*) werden gewählt, da sie sehr ambig sind und es relativ große, annotierte Textmengen gibt (je etwa 4000 Beispiele im *line-hard-serve* Korpus). Weiterhin decken die drei Wörter die Wortklassen Nomen (*line*), Verb (*serve*) und Adjektiv (*hard*) ab. Da das Korpus bereits mit Bedeutungen aus WordNet 1.5 annotiert ist, kann es sowohl zum Trainieren, Testen und als Kontrollmenge benutzt werden. Der Vorteil ist, dass, im Gegensatz zu dem pseudowortbasierenden Verfahren, reale Mehrdeutigkeiten disambiguiert werden, jedoch ist die Methode durch die Verwendung von lediglich drei Wörtern weniger breit gefächert. (Agirre & Edmonds, 2006; Manning & Schütze, 1999)

#### 3.1.2 Senseval-Evaluation

*Senseval* (inzwischen *Semeval*) ist ein Wettbewerb, um WSD-Systeme zu vergleichen und aneinander zu messen. Der erste Wettbewerb fand 1998 mit *Senseval-1* statt; es folgten 2001

---

<sup>23</sup> WSD-Systeme werden unabhängig von einer Applikation getestet, indem die Ausgaben für gegebene Eingaben verglichen werden. (Ide & Véronis, Introduction to the special issue on Word Sense Disambiguation: The state of the art, 1998)

*Senseval-2*, 2004 *Senseval-3* und kürzlich in diesem Jahr *Senseval-4* beziehungsweise *Semeval-1*. Es werden für die Teilnehmer per Hand annotierte Trainingsmengen bereitgestellt, mit der sie ihr Programm trainieren und Bedeutungen unterscheiden. Das Disambiguieren findet auf einer Testmenge statt, und die Ergebnisse werden von den Organisatoren gegen den jeweiligen Gold Standard bewertet. Der Gold Standard ist allgemein eine bekannte Quelle, gegen die getestet wird. Bei *Senseval* gibt dieser die Bedeutungen als Maßstab vor. So wurden für die englische Sprache beispielsweise das Hector-Lexikon, WordNet und FrameNet benutzt. (Agirre & Edmonds, 2006)

*Senseval* deckt zwar ein breiteres Spektrum von mehrdeutigen Wörtern ab und ist ein anerkanntes Verfahren für WSD-Algorithmen (Agirre & Edmonds, 2006), was einen Vergleich erleichtert, ist aber auf dieses WSD-Verfahren nicht ohne Weiteres anwendbar. Der Grund dafür ist, dass bei *Senseval* Lemmata disambiguiert werden, also beispielsweise nicht zwischen den Wortformen *activated* und *activating* unterschieden wird, sondern dem Lemma *activate* zugeordnet sind. (Senseval web page) Das ist jedoch Wissen, das im vorliegenden WSD-Algorithmus nicht beachtet wird<sup>24</sup>.

### 3.1.3 Evaluation mittels Pseudowörtern

Bei einer Evaluation mittels Pseudowörtern wird ein künstliches mehrdeutiges Wort (Pseudowort) aus zwei oder mehreren beliebigen Wörtern gebildet. So können etwa die Wörter *banana* und *door* zu dem zweideutigen Wort *banana-door* zusammengefasst werden, welches dann zwei Bedeutungen hat: die von *banana* und die von *door*. In den Evaluationssätzen wird jedes Auftreten von *banana* und *door* durch *banana-door* ersetzt; findet der WSD-Algorithmus in der Analyse die ursprüngliche Bedeutung heraus, ist die Entscheidung richtig, andernfalls ist sie falsch. (Bordag, 2006; Agirre & Edmonds, 2006)<sup>25</sup> Dieses Verfahren simuliert aufgrund der meist bedeutungsfremden Quellwörter in etwa die Granularität von Homografien, auf der nach Ide und Wilks der Fokus der Disambiguierung liegen sollte (Ide & Wilks, 2006).

Dieses Evaluationsverfahren wurde für diesen WSD-Algorithmus gewählt, da sich die Ergebnisse mit denen des WSI-Algorithmus vergleichen lassen.

---

<sup>24</sup> Siehe dazu auch Abschnitt 1.2.

<sup>25</sup> Ursprünglich beschrieben in: Schütze, H. (1992). Context space. (R. Goldman, P. Norvig, E. Charniak, & B. Gale, Hrsg.) *Working Notes of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, 113–120.

### 3.1.4 Durchführung der Evaluation

Als Evaluationswörter wurden je fünf Wörter aus neun Gruppen gewählt, die eine möglichst breite Abdeckung bieten<sup>26</sup>:

- Für die englische Sprache:
  - Hochfrequente Nomen ( $N_h$ ): picture, average, blood, committee, economy
  - Mittelfrequente Nomen ( $N_m$ ): disintegration, substrate, emigration, thirst, saucepan
  - Niedrigfrequente Nomen ( $N_l$ ): paratuberculosis, gravitation, pharmacology, papillomavirus, sceptre
  - Hochfrequente Verben ( $V_h$ ): avoid, accept, walk, agree, write
  - Mittelfrequente Verben ( $V_m$ ): rend, confine, uphold, evoke, varnish
  - Niedrigfrequente Verben ( $V_l$ ): immerse, disengage, memorize, typify, depute
  - Hochfrequente Adjektive/Adverbien ( $A_h$ ): useful, deep, effective, considerable, traditional
  - Mittelfrequente Adjektive/Adverbien ( $A_m$ ): ferocious, normative, phenomenal, vibrant, inactive
  - Niedrigfrequente Adjektive/Adverbien ( $A_l$ ): astrological, crispy, unrepresented, homoclinic, bitchy
- Für die deutsche Sprache:
  - $N_h$ : Rahmen, Alter, Sohn, Preise, Schulen
  - $N_m$ : Wodka, Urkunde, Extremismus, Gangart, Kunststoff
  - $N_l$ : Computernutzung, Kuhherde, Transportfirma, Milliardenrisiken, Nusschale
  - $V_h$ : zahlen, verhindern, treffen, brauchen, helfen
  - $V_m$ : parken, gleiten, surfen, optimieren, mieten
  - $V_l$ : aufgriffen, eifern, bekochen, pulsieren, exhumieren
  - $A_h$ : zufrieden, direkt, frei, wichtig, neu
  - $A_m$ : endlos, lustige, ratsam, heilig, komplex
  - $A_l$ : ausfahrbare, versicherbar, pommerschen, ungeplanten, geduldiges

Jedes Evaluationswort wird bei einem Auftreten mit jedem der 44 anderen zu einem Pseudowort zusammengeführt und der Satz analysiert. Für jedes Evaluationswort wurden maximal zehn Sätze zufällig ausgewählt, in denen es vorkommt. Für die englische Sprache wurden 415 und für

---

<sup>26</sup> In Anlehnung an (Bordag, 2006). Für die englische Sprache wurden dieselben Evaluationswörter wie für den WSI-Algorithmus gewählt.

die deutsche 445 Sätze gefunden. Das entspricht bei beiden Sprachen je fast 20 Tsd. Einzel-evaluationen.

Beschreibung	<i>tp</i>	<i>fp</i>	<i>precision</i>	<i>recall</i>	<i>F-measure</i>
Englisch	13549	5151	72,45%	100,00%	84,02%
Deutsch	16308	3844	80,92%	100,00%	89,45%

**Tabelle 3.1** Evaluationsergebnisse für die deutsche und englische Sprache.

Die in Tabelle 3.1 dargestellten Ergebnisse entstanden aus Evaluationen mit der in Abschnitt 2.2 beschriebenen Faktorenbelegung und gehören in beiden Sprachen zu den besten der erzielten Resultate. Im Vergleich zu den Evaluationsergebnissen des WSI-Algorithmus für die englische Sprache (Bordag, 2006) ist die *precision* um 13% geringer (WSI: *precision* = 85,42%, *recall* = 72,90%), das *F-measure* jedoch 5% höher als beim WSI-Algorithmus (78,66%). Maßgeblich für die Qualität des WSD-Algorithmus sind jedoch andere Ergebnisse der WSI-Evaluation: *retrieval precision* und *retrieval recall*. Diese geben die Genauigkeit und Vollständigkeit der Kookkurrenzlisten der gefundenen Bedeutungen bezüglich der ursprünglichen Kookkurrenzliste des jeweiligen Wortes an. Je höher beide Werte sind, desto eindeutiger beschreiben die Kookkurrenzen die jeweilige Bedeutung. Falsche Kookkurrenzen zu einer Bedeutung führen bei der Disambiguierung zu inkorrekten Treffern in der Analyse. Der WSI-Algorithmus erreichte bei der Evaluation für den BNC eine *retrieval precision* von 86,83% und einen *retrieval recall* von 62,30%.

### 3.2 Obere und untere Grenze

Jeder WSD-Algorithmus hat eine obere und untere Grenze bezüglich seiner qualitativen Performanz. Die Grenzen dienen dazu, den Schwierigkeitsgrad des Problems in Zahlen zu fassen und die Evaluationsergebnisse damit in Relation zu setzen. Als untere Grenze (*lower bound*) wird einheitlich die sogenannte *baseline* verwendet: die Performanz des einfachsten Algorithmus. Bei WSD-Algorithmen ergibt sich die *baseline* meist durch die Wahl der häufigeren Bedeutung (*majority baseline*) oder auch durch eine zufällige Auswahl (*random baseline*). (Agirre & Edmonds, 2006; Manning & Schütze, 1999) Die häufigere Bedeutung wird hierbei anhand der größeren Kookkurrenzliste bestimmt, was laut dem Zusammenhang zwischen Rang und Größe der Kookkurrenz annähernd gleichbedeutend ist<sup>27</sup>.

<sup>27</sup> Siehe dazu Abschnitt 1.3.



Beschreibung	Evaluation			
	<i>tp</i>	<i>fp</i>	<i>precision</i>	<i>F-measure</i>
<i>majority baseline</i> (Englisch)	9544	9156	51,04%	67,58%
<i>random baseline</i> (Englisch)	9158	9542	48,98%	65,75%
<i>majority baseline</i> (Deutsch)	10011	10141	49,68%	66,38%
<i>random baseline</i> (Deutsch)	10003	10149	49,64%	66,34%

**Tabelle 3.2** *Baselines* für die Evaluation des WSD-Algorithmus. *Recall* beträgt in allen Fällen 100%.

Die *baselines* in Tabelle 3.2 liegen für die *random* und *majority baselines* nahe 50%: bei den *random baselines*, da eine Bedeutung von zwei möglichen zufällig ausgewählt wird. Die theoretische Wahrscheinlichkeit, dass diese Auswahl richtig ist, beträgt 50%. Da für jedes Evaluationswort etwa gleich viele Sätze verwendet werden und sie daher gleichmäßig verteilt sind, liegen auch die *majority baselines* bei etwa 50%.

Als die obere Grenze (*upper bound*) wird die menschliche Performanz angenommen (Agirre & Edmonds, 2006; Manning & Schütze, 1999): Meist mehrere Personen disambiguieren intuitiv die gleiche oder eine ähnliche Testmenge wie die WSD-Algorithmen. Die gemittelten Ergebnisse stellen die obere Grenze dar, denn es wird angenommen, dass es einem automatischen Algorithmus nicht möglich ist, besser zu disambiguieren als ein Mensch (Manning & Schütze, 1999).

Gale et al. (Gale, Church, & Yarowsky) führten solche Tests durch und ermittelten obere Grenzen zwischen 97% und 99%. Allerdings hatten viele der verwendeten Wörter wenige und klar unterscheidbare Bedeutungen, was die Aufgabe, auch für Menschen, erheblich erleichterte. In der Realität überlappen sich Bedeutungen jedoch häufig. Es wird angenommen, dass die obere Grenze bei Wörtern mit klar unterscheidbaren Bedeutungen bei 95% und höher liegt und bei polysemen Wörtern mit oft verwandten Bedeutungen bei 65% bis 70% liegt. (Manning & Schütze, 1999)

Die obere Grenze bei *Senseval-2* lag für Englisch (*all-words task*) bei 75%, die *baseline* bei 57%; der beste Algorithmus erreichte 69% (*supervised*) beziehungsweise 55% (*unsupervised*). Bei *Senseval-3* konnte der beste überwachte Algorithmus mit 65% die obere Grenze von 62% überbieten (ebenfalls Englisch, *all-words task*), da für menschlich-intuitive Disambiguierung keine Fachleute verwendet wurden. (Agirre & Edmonds, 2006)

Für diesen WSD-Algorithmus kann eine obere Grenze von etwa 97% bis 99% angenommen werden, da die Granularität etwa der von den Tests von Gale et al. (Gale, Church, & Yarowsky) entspricht.

### 3.3 Einflussnahmen und Schwellwerte

In diesem Abschnitt werden die Einflussnahmen und Schwellwerte bei der Disambiguierung analysiert und dargestellt, um Schwächen oder auch Verbesserungsmöglichkeiten des Algorithmus aufzuzeigen.

Als Grundlage dient dabei die Faktorenbelegung wie in Abschnitt 2.2 beschrieben:

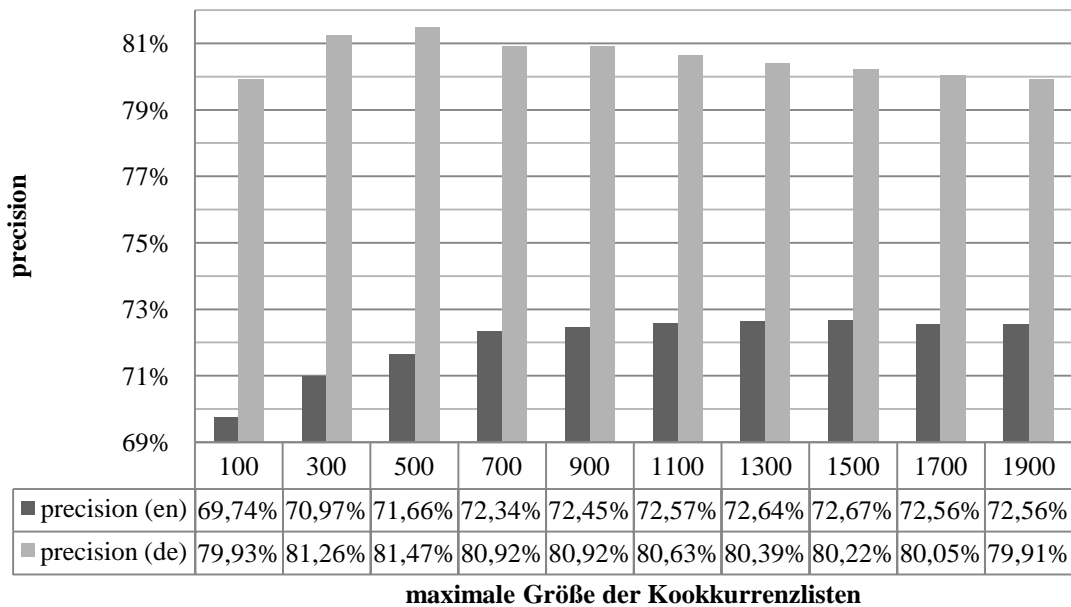
- Es werden die 900 signifikantesten Kookkurrenzen verwendet. Die Kookkurrenzen werden nicht auf einheitliche Länge gekürzt und nicht von Stoppwörtern befreit.
- Kommt ein Satzwort in den Kookkurrenzen einer Bedeutung vor, wird dies mit dem Faktor zehn multipliziert; der inverse Rang, auf dem das Satzwort steht, mit 50.
- Die gemeinsamen Kookkurrenzen von einem Satzwort und einer Bedeutung werden einfach gewertet.
- Die inverse Distanz zwischen der aktuellen Bedeutung und dem betrachteten Satzwort wird mit fünf multipliziert und mit vier potenziert.

Die automatische Evaluation hat außerdem den Vorteil, sehr viele verschiedene Faktorkombinationen automatisch testen zu können. Die gewählten Parameter gehören für beide Sprachen zu den besten der ermittelten Ergebnissen.

#### 3.3.1 Beeinflussung durch Kookkurrenzen

Die Kookkurrenzen können durch Größe, Rang, Normalisierung und Stoppwörter Einfluss auf die Disambiguierung nehmen.

Abbildung 3.1 zeigt die Evaluationsergebnisse in Abhängigkeit von der maximalen Kookkurrenzlistengröße. Zu erkennen ist, dass bei beiden Sprachen die *precision* lediglich um maximal drei Prozentpunkte variiert, alle Ergebnisse jedoch etwa 20% (Englisch) beziehungsweise 30% (Deutsch) höher als die *random baseline* liegen.



**Abbildung 3.1** Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit von der maximalen Größe der Kookkurrenzliste.

Getrennt nach der Frequenzklasse des Quellworts<sup>28</sup> betrachtet, unterscheiden sich die Ergebnisse stärker: Ist das Quellwort hochfrequent, ist eine große Kookkurrenzliste vorteilhaft, denn die beste *precision* für englische Sprache liegt bei 1900 Kookkurrenzen mit 80,48% etwa drei Prozentpunkte höher als bei 900 Kookkurrenzen. Ist das Quellwort niederfrequent, dann werden die höchsten Ergebnisse mit einer kleinen Kookkurrenzliste erzielt (beste *precision* für englische Sprache bei 100 Kookkurrenzen: mit 76,11% etwa sechs Prozentpunkte höher als mit 900 Kookkurrenzen).

Wird der Rang des Satzwortes in den Kookkurrenzen nicht beachtet, liegt die *precision* bei 57,86% (Englisch) beziehungsweise genau 60% (Deutsch). Mit Beachtung des Ranges ist es allerdings vergleichsweise unerheblich, mit welchem Faktor dieser gewichtet wird: Getestet wurden die Faktoren 1, 2, 4, 8, 16, 32, 64, 128, 256 und 512; die Differenz zwischen der höchsten und niedrigsten *precision* in Abhängigkeit von dem Faktor für den Rang beträgt bei beiden Sprachen lediglich 0,14 Prozentpunkte.

Die Motivation für das Normieren der Kookkurrenzlisten, also das Kürzen auf eine einheitliche Länge im Satz, war es zu testen, ob damit Übervorteilungen durch hochfrequente Wörter beziehungsweise Bedeutungen ausgeglichen werden können. Das erzielt jedoch insgesamt keinen Anstieg; die *precision* mit Normierung liegt sechs (Englisch) beziehungsweise fünf Prozent-

<sup>28</sup> Das Quellwort ist das ursprüngliche Wort im Satz, welches vor der Evaluation mit dem Pseudowort ersetzt wird. Wird dieses Wort als Bedeutung ersetzt, ist also die Disambiguierung richtig.

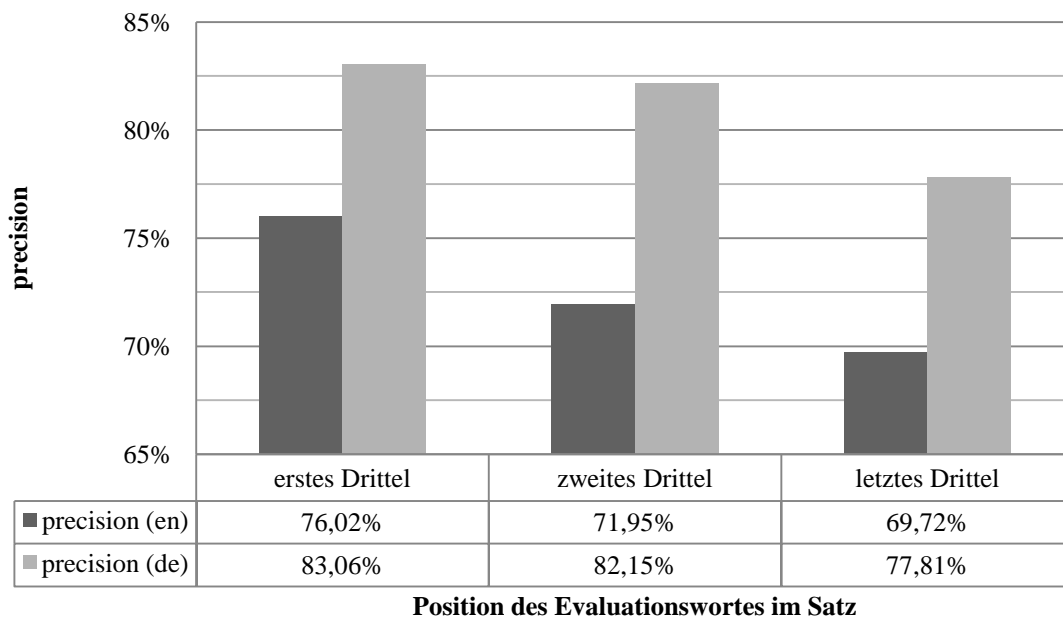
punkte (Deutsch) unter der ohne Normierung. Auch auf einzelne Frequenzgruppen bezogen, gibt es bei den Gruppen nur geringe oder keine Steigerungen. Eine Ausnahme sind die niederfrequenten Quellwörter in der deutschen Sprache: In diesem Bereich bewirkt die Normierung eine Steigerung von elf Prozentpunkten, wenn die Kookkurrenzen normiert werden.

Analog ist es bei der Stoppwortentfernung aus den Kookkurrenzen: Es sollte untersucht werden, ob sich Stoppwörter, also häufige, bedeutungsschwache Wörter, negativ auf die Qualität auswirken. Insgesamt vermindert sich jedoch die *precision* um bis zu zwei Prozentpunkte, wenn die Stoppwörter aus den Kookkurrenzen entfernt werden. Der einzige positive Effekt ist hier bei hochfrequenten Quellwörtern in der englischen Sprache mit einer Steigerung von etwa fünf Prozentpunkten beobachtbar.

Als effektivstes Verhältnis zwischen dem Faktor, mit dem Treffer aus dem Satz in den Kookkurrenzen der Bedeutung gewichtet werden, und dem Faktor, mit dem die Schnittmenge der Kookkurrenzen (Größe) von Satzwörtern und Bedeutungen multipliziert werden, hat sich zehn zu eins herausgestellt. Getestet wurden Verhältnisse von 128 zu eins bis zehn zu 256; der Unterschied zwischen dem geringsten Ergebnis und dem höchsten lag bei unter einem Prozentpunkt.

### **3.3.2 Beeinflussung durch den Satz**

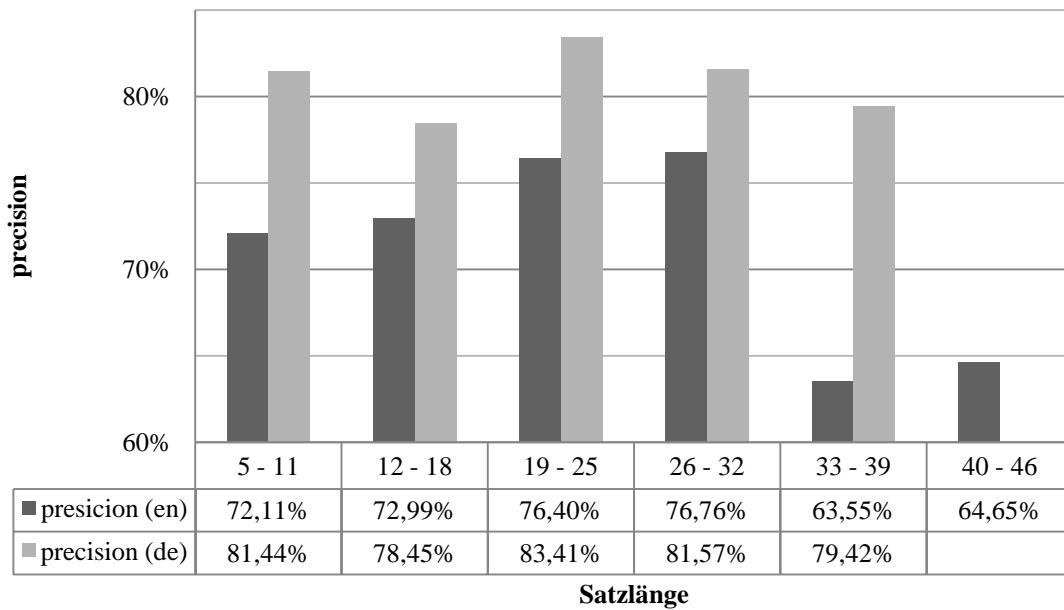
Auch die Struktur des zu disambiguierenden Satzes hat Einfluss auf die Qualität. Ein verhältnismäßig starkes Gewicht hat die Position des mehrdeutigen Wortes im Satz: Steht es im letzten Drittel des Satzes, wird es in beiden Sprachen häufiger falsch disambiguiert als im mittleren oder ersten Drittel (siehe Abbildung 3.2). Eine Ursache dafür kann sein, dass eventuell in den beiden Sprachen rechts stehende Wörter nützlicher für die Disambiguierung sind als links stehende, denn steht das ambige Wort im ersten Drittel des Satzes, hat es den Großteil des Satzes zur rechten Seite.



**Abbildung 3.2** Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit von der Position des Pseudoworts im Satz.

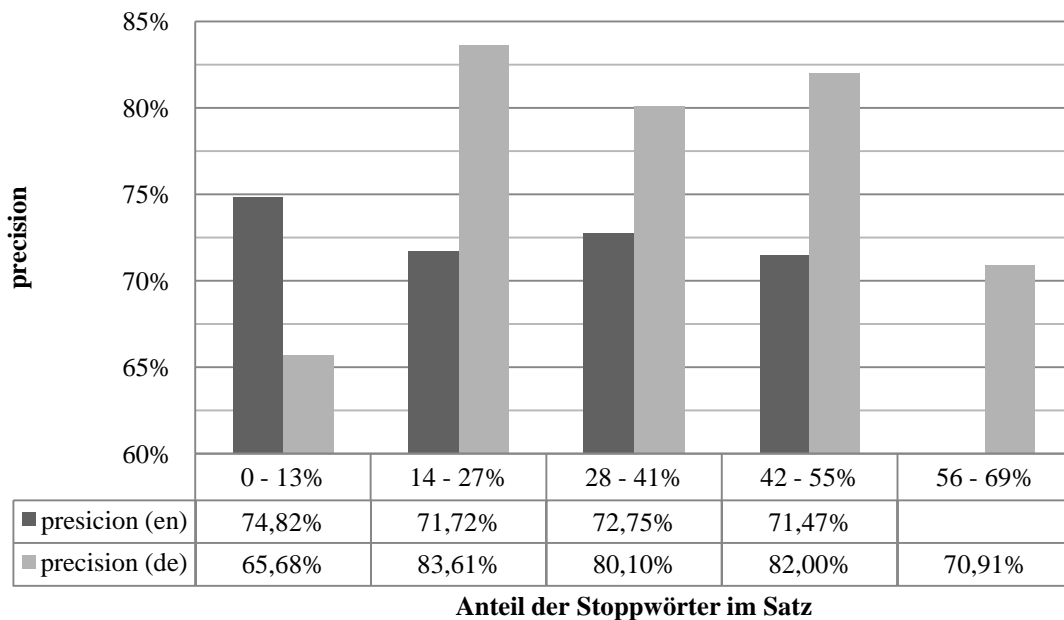
Etwas weniger markant zeigt sich der Einfluss der Distanz zwischen dem betrachteten Pseudowort und den Satzworthern, denn die Verwendung dieses Faktors bringt lediglich eine Steigerung von drei (Deutsch) beziehungsweise vier Prozentpunkten (Englisch). Welcher Wert für den Faktor gewählt wird, ist dabei jedoch ohne messbare Auswirkung; es wurden Werte von eins bis neun getestet. Als Exponent hat sich vier als bester Wert herausgestellt; der quartische Verlauf ist um 3% (Englisch) beziehungsweise 1,5% (Deutsch) wirksamer als der lineare.

Bezüglich der Satzlänge (siehe Abbildung 3.3) ergeben sich bei den beiden Sprachen unterschiedliche Ergebnisse: Während in der englischen Sprache das Mittelfeld, insbesondere im Vergleich zu langen Sätzen mit mindestens 33 Wörtern, stärker ist, sind die Ergebnisse für die deutsche Sprache dagegen eher ausgewogen. Bei gezielter Analyse des Bereiches 33 bis 46 Wörter für die englische Sprache zeigt sich, dass vor allem die Sätze häufiger falsch disambiguiert wurden, in denen das Pseudowort im mittleren Drittel des Satzes stand. Dadurch hat das Pseudowort zu beiden Seiten im Schnitt bis zu 23 Satzworthern, wobei das entfernteste Wort (erstes beziehungsweise letztes Wort im Satz), durch die Beachtung der Distanz, einen höheren Einfluss hat als das entfernteste Wort in einem Satz, in dem das Pseudowort im Randbereich des Satzes steht.



**Abbildung 3.3** Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit von der Satzlänge.

Der Anteil an Stoppwörtern (siehe Abbildung 3.4) im Satz hat in der englischen Sprache einen vergleichsweise unerheblichen Einfluss. In der deutschen Sprache ist die Evaluation bei Sätzen mit einem Stoppwortanteil von bis zu 13% überdies um 15% niedriger als das Gesamtergebnis.



**Abbildung 3.4** Evaluationsergebnisse in deutscher (de) und englischer (en) Sprache in Abhängigkeit vom Stoppwortanteil im Satz.

### 3.3.3 Beeinflussung durch Wort- und Frequenzklasse

Verglichen zu den Ergebnissen aus der WSI-Evaluation<sup>29</sup> liegt die *precision*, aufgeschlüsselt nach Wort- und Frequenzklasse, für den WSD-Algorithmus in jedem Bereich unter dem Wert aus der WSI-Evaluation; teilweise beträgt die Differenz bis zu 16%. Der Vergleich anhand des *F-measure* zeigt jedoch, dass die Ergebnisse etwa gleichauf sind, meist sogar besser (bis zu 12%) als beim WSI-Algorithmus.

Speziell ergeben sich beim Disambiguieren Probleme, wenn ein Quellwort mit einem höherfrequenten Wort (Mergewort) zu einem Pseudowort gemischt wird; dies ist deutlich bei den Ergebnissen für die deutsche Sprache zu erkennen (Tabelle 3.3).

QW \ MW	hoch-frequent	mittel-frequent	nieder-frequent
hoch-frequent	76,65%	66,98%	68,77%
mittel-frequent	76,50%	68,77%	70,88%
nieder-frequent	78,17%	71,13%	71,92%

QW \ MW	hoch-frequent	mittel-frequent	nieder-frequent
hoch-frequent	82,76%	78,10%	46,39%
mittel-frequent	92,76%	88,79%	55,54%
nieder-frequent	98,27%	98,85%	83,10%

**Tabelle 3.3** Die *precision* in Abhängigkeit von der Frequenzklasse des Quellworts (QW) und Mergeworts (MW) in englischer (links) und deutscher Sprache (rechts).

Im Gegensatz zur englischen Sprache, in der Adjektive und Adverbien als Quellwörter am defektivsten disambiguiert werden, sind in der deutschen Sprache die Nomen am problematischsten (siehe Tabelle 3.4). Die Gründe dafür können auf grammatikalische Charakteristiken der Sprachen zurückgeführt werden.

QW \ MW	Nomen	Verb	Adj./Adv.
Nomen	77,63%	73,48%	63,09%
Verb	82,00%	68,24%	64,97%
Adj./Adv.	82,00%	76,02%	66,30%

QW \ MW	Nomen	Verb	Adj./Adv.
Nomen	75,81%	86,49%	83,29%
Verb	74,71%	82,17%	82,54%
Adj./Adv.	78,32%	84,86%	80,17%

**Tabelle 3.4** Die *precision* in Abhängigkeit von der Wortklasse des Quellworts (QW) und Mergeworts (MW) in englischer (links) und deutscher Sprache (rechts).

<sup>29</sup> Dort wurden die Ergebnisse für die englische Sprache (BNC) ebenfalls nach Frequenz- und Wortklasse aufgeschlüsselt.

### 3.3.4 Beeinflussung durch den Schwellwert

Für einen praktischen Einsatz eines Disambiguierungsalgorithmus kann es sinnvoll sein, Entscheidungen nur dann zu treffen, wenn sie als relativ sicher angesehen werden. Wie in Abbildung 3.5 zu sehen, steigt die *precision*, wenn der prozentuale Unterschied zunimmt. Der Wert des prozentualen Unterschieds ist der relative Anteil der zweithöchsten Bewertung an der höchsten Bewertung; in der Evaluation ist die zweithöchste Bewertung gleichzeitig auch die niedrigste, da die Pseudowörter nur zwei Bedeutungen haben.

Beide Sprachen haben einen etwa linearen Anstieg der *precision* bis hin zu 90%, wo die *precision* wieder sinkt; sehr knappe Entscheidungen sind häufiger falsch.

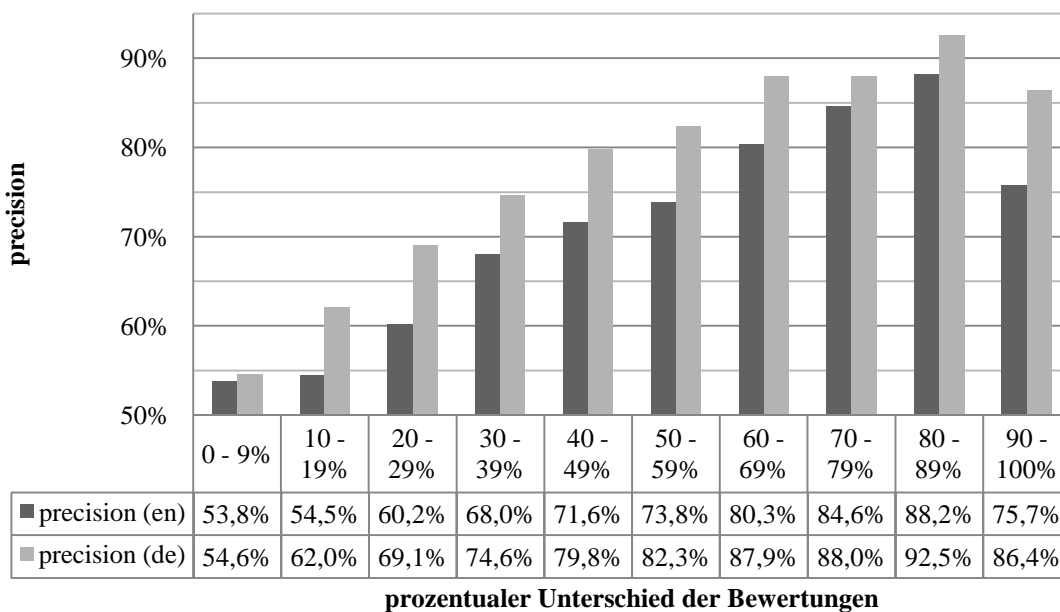


Abbildung 3.5 Die *precision* in Abhängigkeit vom prozentualen Unterschied der Bewertungen.

Wird für die deutsche Sprache ein Schwellwert 65% bis 90% festgesetzt, das heißt, nur Entscheidungen mit einem prozentualen Unterschied der Bewertungen zwischen den beiden Werten werden getroffen, ergibt sich eine *precision* von 90% und ein *recall* von 26% (*F-measure*: 40%).

## 3.4 Vergleich zu bestehenden Verfahren

Wie unter Abschnitt 3.1 beschrieben, ist es sehr schwierig, Disambiguierungsverfahren aufgrund ihrer Vielfältigkeit (automatisch - überwacht, Ressourcen, Einsatzzweck) und der verschiedenen Evaluationsverfahren zu vergleichen. Dennoch erleichtert die Angabe der unteren und oberen Grenze ein Gegenüberstellen der Ergebnisse.

Wie aus Tabelle 3.5 ersichtlich, hat der Algorithmus für die englische Sprache zwar eine geringere relative *precision* als der beste *Senseval-1*-Algorithmus (41 Wörter mit 8448 Instanzen)



und als der beste Algorithmus bei *Senseval-2* im *all-words task* (1.082 Wörter mit 2.437 Instanzen), aber eine höhere als der beste aus 26 Algorithmen im *Senseval-2 lexical sample test* (73 Wörter mit 12.939 Instanzen).

	<i>Senseval-1</i> (Englisch)	<i>Senseval-2</i> (Englisch)			dieser Algorithmus	
		<i>all-words</i> , überwacht	<i>lexical sample</i> , überwacht	<i>lexical sample</i> , automatisch	Englisch	Deutsch
untere Grenze	57%	57%	48%	16%	51%	50%
obere Grenze	95%	75%	86%	86%	98%	98%
beste <i>precision</i>	78%	69%	64%	40%	72%	81%
relative <i>precision</i> <sup>30</sup>	55%	67%	42%	34%	45%	65%

Tabelle 3.5 Gegenüberstellung von *Senseval*-Ergebnissen<sup>31</sup> und Evaluationsergebnissen dieses Algorithmus.

### 3.5 Fazit

Die beiden Sprachen unterscheiden sich im Gesamtergebnis relativ stark. Die Differenz kann zustande kommen, da der Sprachbau bei Deutsch (flektierender Sprachbau, Großschreibung von Nomen) und Englisch (isolierender Sprachbau) unterschiedlich ist, und da für die deutsche Sprache eine größere Datenbasis (Korpus) verwendet wurde. Bei *Senseval-2* wurde mit Tschechisch ebenfalls eine Sprache mit flektierendem Sprachbau evaluiert. Das System erreichte mit einer *precision* von 94% den höchsten Wert von allen Systemen aus allen Sprachen; allerdings gibt es keine Angaben zu der unteren und oberen Grenze der Evaluation. (Edmonds & Kilgarriff, Introduction to the Special Issue on Evaluating Word Sense Disambiguation Systems, 2002) Dennoch kann aufgrund der *Senseval-2*-Ergebnisse und der Ergebnisse für Deutsch im Vergleich zu Englisch angenommen werden, dass flektierende Sprachen leichter zu disambiguieren sind als isolierende. Die unterschiedlichen Reaktionen auf Einflüsse (etwa Satzlänge,

<sup>30</sup> Die relative *precision* stellt eine relative Position der eigentlichen *precision* bezüglich der unteren und oberen Grenze dar. Die obere Grenze wird dabei als 100%, die untere als 0% angesehen.

<sup>31</sup> Quellen: (Agirre & Edmonds, 2006) und (Edmonds & Kilgarriff, Introduction to the Special Issue on Evaluating Word Sense Disambiguation Systems, 2002). *Senseval-3* wurde wegen der Anomalie der oberen Grenze nicht beachtet (siehe Abschnitt 3.2).

Stoppwörter, Wortklasse) und Parameter können ebenfalls auf grammatikalische Eigenschaften der Sprachen zurückgeführt werden.

Für beide Sprachen wirken sich generell große Kookkurrenzlisten positiv aus; ist allerdings eine niederfrequente Bedeutung beteiligt (mit nur wenig vorhandenen Kookkurrenzen) wird die höherfrequente Bedeutung häufiger übervorteilt. Das Problem der Datenknappheit („Data Sparseness“) im niederfrequenten Bereich ist eine Erschwernis, die viele korpusbasierende WSD-Algorithmen gemeinsam haben. (Ide & Véronis, Introduction to the special issue on Word Sense Disambiguation: The state of the art, 1998) Die für die Evaluation gewählte Größe der Kookkurrenzliste (900) bietet jedoch ein verhältnismäßig gutes Gesamtergebnis. Es wurde darüber hinaus gezeigt, dass der Rang in den Kookkurrenzlisten (die Ordnung nach Signifikanz in Kookkurrenzlisten) wichtig für die Ermittlung der richtigen Bedeutung ist, aber ein kleiner Wert als Faktor ausreichend ist.

Die Berücksichtigung der Distanz von der betrachteten Bedeutung zu den einzelnen Satz-  
wörtern erwirkt eine weitere Steigerung der *precision* von insgesamt sechs Prozent (Faktor und Exponent), was zeigt, dass näher stehende Wörter relevanter für die Bedeutungsfindung sind als entferntere. Aus den Ergebnissen aus Abschnitt 3.3.2 bezüglich der Satzlänge lässt sich ableiten, dass lange Sätze (ab etwa 30 Wörtern) in der englischen Sprache Rauschen verursachen und zu vermehrten Fehlentscheidungen führen.

## 4. Ansätze zur Optimierung

Es ist möglich, die *precision* des Algorithmus durch Erweiterungen zusätzlich zu steigern. Dies kann beispielsweise erreicht werden, indem ein Schwellwert festgelegt wird, der gewährleistet, dass Entscheidungen über die richtige Bedeutung sicher genug sind. Eine praktische Anwendung, bei der dies sinnvoll sein kann, ist das Finden von Beispielsätzen zu Bedeutungen eines Wortes.

Im Folgenden werden einige weitere allgemeine Ansätze zur Optimierung kurz erläutert.

### 4.1 Automatische Optimierung und Kombination von Konfigurationen

Eine Methode, um den Ablauf der Disambiguierung weiter zu automatisieren, ist die automatische Optimierung der Konfiguration (Faktorenbelegung). Dazu können aus vorhandenen Daten selbstständig Evaluationswörter verschiedener Frequenz- und Wortklassen und entsprechende Evaluationssätze entnommen werden, auf dessen Basis die Konfiguration optimiert wird. Durch Filter und Vorgaben, wie etwa ein Höchst- oder Mindestmaß an Ähnlichkeit der Evaluationswörter, kann die Evaluation die realen Bedingungen präziser simulieren.

Aufgrund dieser Evaluationsergebnisse lassen sich allgemeine und sprachspezifische Schwächen in der Disambiguierung durch alternative Konfigurationen ausgleichen. So kann die verwendete Größe der Kookkurrenzlisten abhängig von der Frequenz der Bedeutungen gewählt werden, um häufige Fehlentscheidungen, beispielsweise in Deutsch bei seltenen Bedeutungen, zu mindern (siehe Abschnitt 3.3.3). Ebenso können verschiedene Konfigurationen automatisch in Abhängigkeit von Satzeigenschaften (Position des ambigen Wortes, Satzlänge, Stoppwortanteil) und Wortklasse angewandt werden, um die Performanz zu verbessern, da die Evaluationen gezeigt haben, dass in bestimmten Bereichen eine abweichende Konfiguration eine bessere Leistung erzielen kann.

Es hat sich herausgestellt, dass es nützlich sein kann, die Fenstergröße nicht auf den kompletten Satz zu belassen, sondern einen kleineren Kontext zu betrachten, um somit Rauschen durch lange Sätze zu vermindern (Abschnitt 3.3.2). Die Fenstergröße sollte für optimale Ergebnisse für jede Sprache gesondert ermittelt werden.

### 4.2 Linke und rechte Nachbarn

Auch die direkten Nachbarn eines ambigen Wortes können qualitätssteigernd bei der Disambiguierung sein. Fundiert wird diese Annahme durch die Evaluationsergebnisse bezüglich der Distanz vom ambigen Wort zu dem aktuell betrachteten Satzwort (siehe Abschnitt 3.3.2) und der Einflussnahme des Ranges in den Kookkurrenzen (siehe 3.3.1), da sich sowohl die Distanz als auch die Berücksichtigung des Ranges fördernd auswirken. Insbesondere sind linke und rechte

Nachbarn hilfreich bei Wörtern mit Bedeutungen verschiedener Wortklassen (Beispiel: *to record / the record*) oder bei Homografien (Beispiel: *der/die Kiefer*).

### 4.3 Sprach- und syntaxabhängige Optimierungen

Es sind weitere Optimierungen möglich, welche allerdings zulasten der Sprach- und Syntax-unabhängigkeit des Algorithmus gehen. So ist es möglich, einen Part-of-Speech-Tagger einzusetzen, um die Wörter und Bedeutungen Wortklassen zuzuordnen, was sich ebenfalls positiv auf Wörter mit Bedeutungen verschiedener Wortklassen auswirkt. Ist die Syntax eines Satzes für die verwendete Sprache bekannt oder kann zumindest teilweise in Regeln abgebildet werden, kann das Problem der Disambiguierung reduziert werden, da erkannt wird, welche Wortklasse (und somit Bedeutung oder Bedeutungsmenge) in einem Kontext passend ist.

In dem Zusammenhang des PoS-Taggings kann auch die Beachtung der Valenz vorteilhaft sein. Valenz in der Sprachwissenschaft bezieht sich nicht nur auf Verben, sondern kann auch auf andere Wortklassen angewandt werden und bezeichnet die Wertigkeit von Lexemen. (Hellwig, 1978) Es können dafür Valenzwörterbücher verwendet werden, in denen mögliche syntaktische Umgebungen zum Valenzeintrag vermerkt sind.

Lemmatisierung (Zuordnung der Wortformen zum entsprechenden Lemma) eignet sich dazu, die Frequenz von einzelnen Lexemen zu erhöhen und somit auch die Kookkurrenzlisten von seltenen Wortformen beziehungsweise Bedeutungen dementsprechend zu vergrößern. Weiterhin können durch Lemmatisierung Homografien aufgelöst beziehungsweise identifiziert werden (Beispiel: mehrdeutiges Wort *die Mutter* mit den Pluralen *die Muttern/Mütter*).

## 5. Fazit

Es wurde gezeigt, dass der in dieser Arbeit vorgestellte Algorithmus automatisches Disambiguieren mit guten Gesamtergebnissen ermöglicht. Die Qualität hängt dabei stark von der des WSI-Algorithmus ab. Im Speziellen sind die Ergebnisse für hochfrequente Wörter beziehungsweise Bedeutungen vergleichsweise hoch; für diesen Bereich sind große Kookkurrenzlisten opportun. Mit niedrigerer *precision* werden seltene Wörter beziehungsweise Bedeutungen disambiguiert, da sie nur über wenige Kookkurrenzen verfügen.

Herausgestellt hat sich eine signifikant höhere Qualität für die deutsche gegenüber der englischen Sprache, jedoch auch teilweise unterschiedliche Reaktionen auf Parameter und Einflussnahmen. Daraus ergibt sich, dass für optimale Ergebnisse für jede Sprache (oder gegebenenfalls Domäne) die Faktorenbelegung separat ermittelt werden sollte. Als wichtige Einflussnahmen auf die Qualität haben sich, neben der Frequenzklasse und der Größe der Kookkurrenzliste, die Distanz im Satz und der Rang in den Kookkurrenzen erwiesen.

Offen bleibt die Frage, warum die Disambiguierung häufiger fehlschlägt, wenn das Pseudowort im letzten Drittel des Satzes steht, und die *precision* am höchsten ist, wenn es im ersten Drittel steht. Angenommen wurde, dass die Ergebnisse bei einer Position im mittleren Bereich am besten sind, da die Distanz hoch bewertet wird und dadurch gerade diese Gruppe bevorteilt sein sollte, da sie im Schnitt die doppelte Menge an nahstehenden und dadurch hochbewerteten Satzworthern zur Verfügung hat.

**Literaturverzeichnis**

- Agirre, E., & Edmonds, P. (Hrsg.). (2006). *Word sense disambiguation : algorithms and applications* (Bd. 33; Text, speech and language technology). Dordrecht [u.a.]: Springer.
- Bordag, S. (2006). Word Sense Induction: Triplet-Based Clustering and Automatic Evaluation. In *Proceedings of EACL 06*. Trento.
- Brody, S., Navigli, R., & Lapata, M. (2006). Ensemble Methods for Unsupervised WSD. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics* (S. 97-104). Sydney.
- Bußmann, H. (Hrsg.). (2002). *Lexikon der Sprachwissenschaft* (3. Ausg.). Stuttgart: Kröner.
- Carstensen, K.-U., Ebert, C., Endriss, C., Jekat, S., Klabunde, R., & Langer, H. (Hrsg.). (2004). *Computerlinguistik und Sprachtechnologie : eine Einführung* (2. Ausg.). München: Elsevier.
- Dunning, T. E. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* , 19 (1), 61-74.
- Edmonds, P. (2006). Lexical disambiguation. In K. Brown (Hrsg.), *Elsevier Encyclopedia of Language and Linguistics* (2. Ausg., S. 607-630). Oxford: Elsevier.
- Edmonds, P., & Kilgarriff, A. (2002). Introduction to the Special Issue on Evaluating Word Sense Disambiguation Systems. *Natural Language Engineering* , 8 (4), 279-291.
- Gale, W., Church, K. W., & Yarowsky, D. Estimating Upper and Lower Bounds on the Performance of Word-Sense Disambiguation Programs. In *ACL 30* (S. 249-256).
- Glück, H. (Hrsg.). (2005). *Metzler Lexikon Sprache* (3. Ausg.). Stuttgart; Weimar: Metzler.
- Hellwig, P. (1978). *Formal-desambiguierte Repräsentation : Vorüberlegungen zur maschinellen Bedeutungsanalyse auf der Grundlage der Valenzidee*. Stuttgart: Hochschulverlag.
- Ide, N., & Véronis, J. (1998). Introduction to the special issue on Word Sense Disambiguation: The state of the art. *Computational Linguistics* , 24 (1), S. 1-40.
- Ide, N., & Wilks, Y. (2006). Making sense about sense. In E. Agirre, & P. Edmonds (Hrsg.), *Word sense disambiguation : algorithms and applications* (Bd. 33; Text, speech and language technology). Dordrecht [u.a.]: Springer.
- Kilgarriff, A. (15. März 1996). *Read-me for Kilgarriff's BNC word frequency lists (all.num.gz)*. Abgerufen am 05. Juli 2007 von Adam Kilgarriff: Home Page: <http://www.kilgarriff.co.uk>

- Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, Mass. [u.a.]: MIT Press.
- Patwardhan, S., Banerjee, S., & Pedersen, T. (23-24. Juni 2007). UMND1: Unsupervised Word Sense Disambiguation Using Contextual Semantic Relatedness. *Appears in the Proceedings of SemEval-2007: 4th International Workshop on Semantic Evaluations* , 390-393.
- Senseval web page*. (kein Datum). Abgerufen am 4. Oktober 2007 von Senseval web page: <http://www.senseval.org/>
- University of Oxford (Hrsg.). (kein Datum). *[bnc] About the British National Corpus: The BNC in numbers*. Abgerufen am 26. August 2007 von [bnc] British National Corpus: <http://www.natcorp.ox.ac.uk>
- University of Oxford (Hrsg.). (2007). *[bnc] About the British National Corpus: What is the BNC?* Abgerufen am 26. August 2007 von [bnc] British National Corpus: <http://www.natcorp.ox.ac.uk>
- Zipf, G. K. (1965). *Human behavior and the principle of least effort : an introduction to human ecology* (Facs. of 1949 ed.). New York [u.a.]: Hafner.

## **Erklärung**

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Leipzig

14.11.2007

Ort

Datum

Unterschrift