# *Lingo* – ein open source System für die Automatische Indexierung des Deutschen

*Klaus Lepsky / John Vorhauer*

Stand: 20. Januar 2006

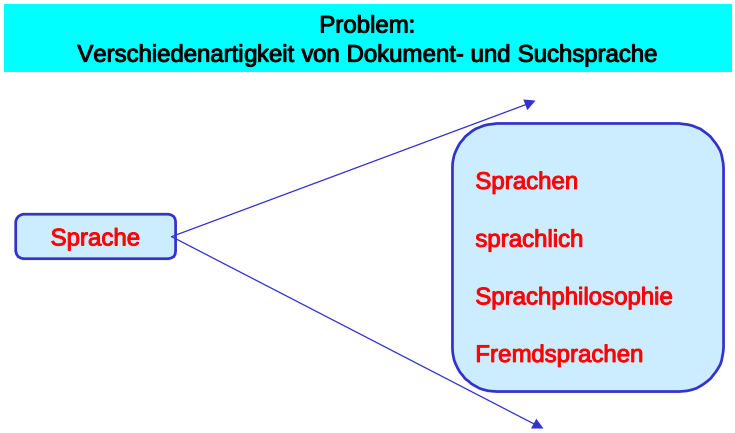
## Inhaltsverzeichnis



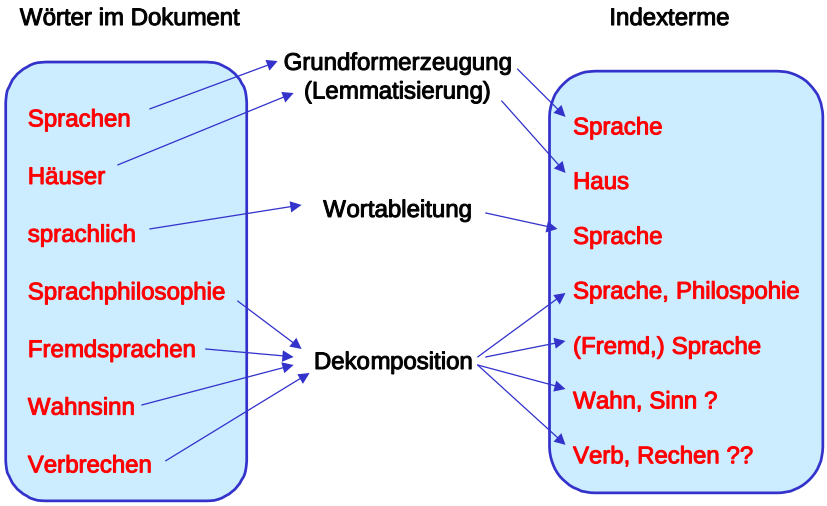
## 1. Automatische Indexierung und Information Retrieval

*Lingo* ist ein linguistisch basiertes System zur automatischen Indexierung des Deutschen. Unter automatischer Indexierung wird hier die Fähigkeit verstanden, aus elektronisch vorliegenden textbasierten Dokumenten geeignete Indexterme für ein Information Retrieval zu extrahieren. Im Gegensatz zur reinen Volltextindexierung, bei der jede Zeichenkette als Indexterm verwendet wird, erfolgt während einer automatischen Indexierung eine linguistische und in Teilen auch semantische Überarbeitung der Terme, bevor diese in den Index geschrieben werden. Der Effekt einer automatischen Indexierung hinsichtlich des Retrievals liegt vor allem in einer signifikanten Erhöhung des Recalls, d.h. der Zahl der gefundenen relevanten Dokumente in der

Suche.(1) Bewirkt wird dies durch die sprachliche Zusammenführung grammatikalisch unterschiedlicher Wortformen im Dokumenttext. So verhindert etwa die Verschiedenheit von Singular- und Pluralformen von Substantiven eine einfache Suche nach allen Dokumenten mit entweder nur der Singular- oder nur der Pluralform eines bestimmten Substantivs: die Suche nach „Sprache“ findet keine Dokumente mit dem Term „Sprachen“ (und umgekehrt), darüber hinaus ebenfalls nicht die Dokumente mit „sprachlich“, „Sprachphilosophie“ oder „Fremdsprachen“:




Im Deutschen ist die Zahl der möglichen Varianten für einen eingegebenen Suchbegriff durch die Kompositumbildung deutlich höher als z.B. im Englischen. Vereinfacht ausgedrückt bedeutet dies, dass sich Hinweise auf sinnvolle Indexterme nicht nur am Anfang einer Zeichenkette befinden können sondern ebenso gut am Ende oder in der Mitte. Die für eine automatische Indexierung benötigte Funktionalität muss daher über die reine Fähigkeit zur Grundformreduktion (Lemmatisierung) deutlich hinausgehen:




Die Beispiele verdeutlichen, dass bereits die Rückführung von Plural- auf Singularformen im Deutschen nicht trivial ist, denn die Unregelmäßigkeit der Pluralbildung kann auch zu Veränderungen im Wortstamm führen, es entstehen bereits ab dem zweiten Buchstaben zwei verschiedene Zeichenketten. Anders als im Englischen, für das es stabil arbeitende Regelwerke für das sog. Stemming gibt, ist es daher für das Deutsche üblich, für eine automatische Indexierung umfangreiche Lexika einzusetzen.

Die Wortableitung, d.h. die Überführung eines Wortes in eine andere Wortklasse – hier vom Adjektiv auf das Substantiv – erlaubt die Bereitstellung von Indextermen in der bevorzugten grammatikalischen Form, für das Retrieval in der Regel die substantivische Form. Dies ist insbesondere deshalb von Bedeutung, weil es im Deutschen grundsätzlich möglich ist, eine Aussage in drei inhaltsgleichen aber grammatikalisch stark unterschiedlichen Varianten zu treffen:

„Aufschwung der Wirtschaft“ (Wortfolge von Substantiven),

„Wirtschaftsaufschwung“ (Kompositum) und

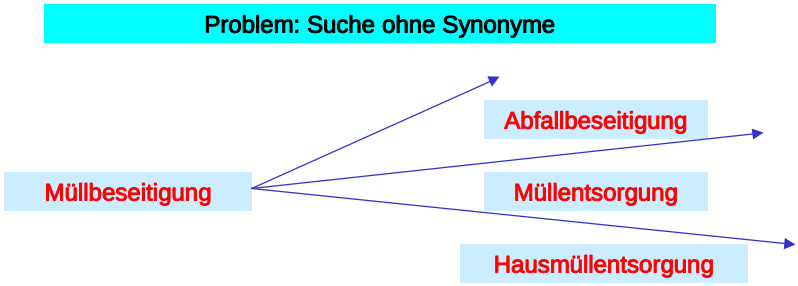
„wirtschaftlicher Aufschwung“ (Adjektiv-Substantiv-Verbindung)

sind als inhaltsgleiche Aussagen in Texten austauschbar, führen allerdings zu deutlich verschiedenen Indextermen. Durch die Wortableitung wird die Adjektiv-Substantiv-Verbindung inhaltlich identisch mit der Substantiv-Wortfolge („wirtschaftlicher Aufschwung“ -> „wirtschaft“, „aufschwung“). Durch die Kompositumzerlegung wird das Kompositum auf der Ebene der Indexterme identisch mit der Substantiv-Wortfolge („Wirtschaftsaufschwung -> „wirtschaft“, „aufschwung“), sprachlich wird also zusammengeführt.

Die Identifizierung von Komposita ist für die Zwecke der Indextermgenerierung unabdingbar, technisch allerdings nicht einfach zu realisieren, weil mit der Länge der zu identifizierenden Zeichenketten auch die Chancen für Fehlidentifikationen steigen. Mehrdeutigkeiten wie z.B. eine Zerlegung von „Verbrechen“ oder „Wirtschaft“ lassen

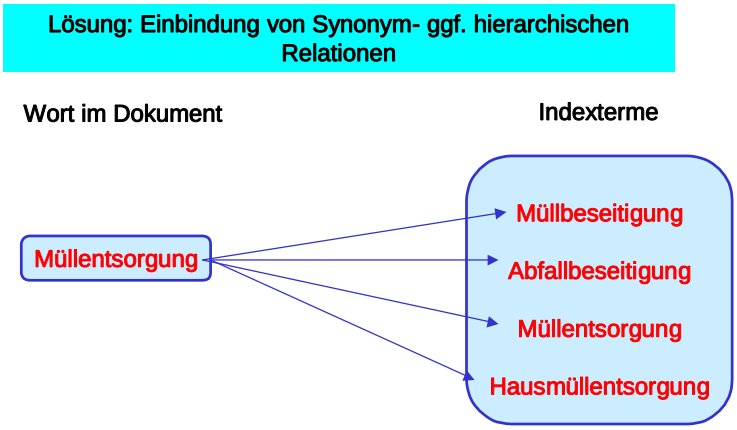
sich durch Lexikalisierung abfangen, generell fehlerfrei wird eine Zerlegung jedoch nicht arbeiten können, weil es nicht möglich (und auch nicht sinnvoll) ist, die Gesamtheit der Komposita im Deutschen zu lexikalisieren. Insgesamt überwiegt jedoch eindeutig der Nutzen der Zerlegung für die Indextermgenerierung.

Neben den grammatikalisch bedingten sprachlichen Problemen beim Information Retrieval sind die Probleme auf der Bedeutungsebene mindestens in gleichem Maße verantwortlich für zu geringen Recall. Ursache ist in erster Linie die Synonymie, d.h. die Existenz verschiedener gleichwertig zu benutzender Benennungen für einen Sachverhalt: eine Suche mit „Müllbeseitigung“ findet keine Dokumente mit „Abfallbeseitigung“, „Müllentsorgung“ oder „Hausmüllentsorgung“.:



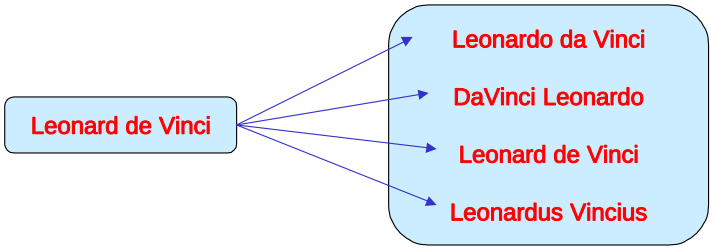

Das bewährte Instrument zur Vermeidung dieser Problematik ist natürlich eine intellektuelle Erschließung, die über die Zuteilung einer Vorzugsbenennung zum Dokument und die damit verbundenen Verweisungen garantiert, dass unabhängig vom gewählten Suchbegriff immer alle Literatur zum Thema gefunden wird. Für das Information Retrieval in nicht erschlossenen Dokumentkollektionen besteht alternativ nur die Möglichkeit, eine Suche mit allen synonymen Varianten als boolesches ODER abzuschicken – praktiziert höchstens von Profis.

Falls ein terminologisch kontrolliertes Vokabular existiert und genutzt werden kann, ist es möglich, Synonymbeziehungen (theoretisch auch andere Relationen) in eine automatische Indexierung einzubinden. Dies kann in der einfachsten Variante so erfolgen, dass die Indexierung zu einem potenziellen Indexterm alle bekannten Synonyme als zusätzliche Indexterme generiert. Es ist dann unerheblich, mit welchem Term gesucht wird, alle Dokumente verfügen über alle Indexterme einer Äquivalenzklasse.




Da synonyme Wortformen nicht nur als Einzelbegriffe sondern auch als Mehrwortbegriffe vorkommen („Bayerische Motorenwerke“, „BMW“), ist es für eine umfassende Relationierung erforderlich, dass die automatische Indexierung über Mechanismen zur Erkennung von Mehrwortgruppen verfügt. Dies kann durch Lexikalisierung gelöst werden, wobei dann Mechanismen greifen müssen, die die Unregelmäßigkeit innerhalb der Mehrwortgruppe abfangen („den Bayerische**n** Motorenwerke**n**“). Dies kann aber auch durch regelbasierte Ansätze geschehen, wobei nach erfolgter Grundformreduktion Muster von Wortfolgen analysiert und extrahiert werden, z.B. also alle identifizierten Folgen von Adjektiv und Substantiv.(2) Der Vorteil dieses Verfahrens ist natürlich, dass auch nicht lexikalisierte Mehrwortgruppen gefunden werden. Nachteilig kann sein, dass auch nichtssagende Gruppen extrahiert werden („blauer Himmel“).

Der Nutzen einer Erkennung von Mehrwortgruppen lässt sich beim Vorhandensein entsprechender Vokabularien auf die Identifizierung von z.B. Eigennamen ausweiten:




Im Überblick stellt sich die sinnvolle Funktionalität einer linguistisch basierten automatischen Indexierung des Deutschen für eine deutliche Verbesserung des Retrievalerfolgs damit wie folgt dar:

Erzeugung von grammatikalischen Grundformen:

```
Informationen                 ,     Information
```

Zerlegung von Komposita:

```
Informationserschließung      ,     Information, Erschließung
```

Bildung von Wortableitungen (bevorzugt adjektivische auf substantivische Form):

```
technisch                     ,     Technik
```

Erkennen von Mehrwortgruppen, festen Wendungen:

```
„objektorientierte Datenbank“ ,     Datenbank, objektorientiert
„Joseph Weizenbaum“           ,     Weizenbaum, Joseph
```

Relationierung von Synonymen (bzw. hierarchischen Beziehungen)

```
elektronische Datenverarbeitung     ,     EDV
```

## 2. Die automatische Indexierung *lingo*(3)

*Lingo* bezieht sich auf eine Indexierungsidee des open source-Systems FREELING(4), das eine linguistische Indexierung des Spanischen, Katalanischen und Englischen leistet. Der in FREELING verfolgte Ansatz, die Grundformidentifizierung bzw. -reduktion mit einem Grundformenwörterbuch und einer zugehörigen einfachen Suffixliste zu realisieren, wurde für *lingo* und damit für das Deutsche übernommen. Jedoch

wurden für *lingo* zusätzlich eine algorithmische Kompositumzerlegung, eine lexikalische und algorithmische Mehrworterkennung und eine allgemeine lexikalische Relationierung realisiert.

*Lingo* ist vollständig in der Programmiersprache ruby(5) programmiert ist. Da *lingo* v.a. für Forschung und Lehre eingesetzt wird, wurde bei der Entwicklung großer Wert auf die Transparenz des Programms und seiner Bestandteile sowie auf die Transparenz der Programmabläufe bei einer Indexierung gelegt. Alle Funktionen des Systems sind nahezu vollständig konfigurierbar, damit für eigene Einsatzzwecke leicht anzupassen.

Als wörterbuchbasiertes linguistisches System ist *lingo* von den verfügbaren und eingebundenen elektronischen Wörterbüchern direkt abhängig. *Lingo*-Wörterbücher sind als Quelltext reine Textfiles, die mit jedem Editor bearbeitet werden können. Für die Verarbeitung durch *lingo* werden aus den Wörterbuchquellen Datenbanken (sdbm(6)) generiert. Das für *lingo* aufgebaute Systemwörterbuch basiert auf Terminologie, die aus Einträgen aus dem Grundwörterbuch von MORPHY(7) und dem deutschsprachigen MySpell-Lexikon(8) stammt. Das Synonym-Wörterbuch entspricht dem deutschen OpenThesaurus(9).

## 2.1 Grundformerkennung

Das Wörterbuchkonzept von *lingo* ist bewusst einfach gehalten, um eine leicht zu realisierende Möglichkeit für eigene Wörterbucherweiterungen zu bieten. Das Systemwörterbuch (system_dic) enthält ca. 150.000 Grundformeinträge in folgender Form:

```
abbitte,abbitte #s

häuten,häuten #v haut #s

schreibung,schreibung #s
```

oder allgemein:

```
wortform,grundform1 #wortklasse1 (grundform2 #wortklasse2 etc.)
```

Die Identifizierung einer Zeichenkette erfolgt über den Eintrag „wortform“ in Verbindung mit einer Suffixliste, die mögliche Endungen einer Grundform enthält und damit eine Erkennung von „wortform+endung“ erreicht. Dabei ist die Zuordnung der Suffixe zum Wörterbucheintrag wortklassenspezifisch, d.h. für die Erkennung von Substantiveinträgen werden nur die Substantiv-Suffixe zugelassen. Die Suffixliste für z.B. Substantive sieht so aus:

```
#    Suffixliste, Stand: 30-06-2005
#    1. Spalte:  Suffixe
#    2. Spalte:  Suffixersetzung
#    3. Spalte:  Wortklasse
#    #s Substantiv
#    #a Adjektiv
#    #v Verb
#    #e Eigenname
#    [#w Wortform]
#    #f Fugung
#    Substantiv-Endungen
- [ e,     "*", "#s" ]
- [ en,    "*", "#s" ]
- [ er,    "*", "#s" ]
- [ ern,   "*", "#s" ]
```

```
- [ es,    "*", "#s" ]
- [ n,     "*", "#s" ]
- [ s,     "*", "#s" ]
- [ se,    "*", "#s" ]
- [ sen,   "*", "#s" ]
- [ ses,   "*", "#s" ]
```

In Verbindung mit dem o.g. Beispiel „Schreibung“ führt der Wörterbucheintrag „schreibung“ + Suffix „en“ zur korrekten Identifizierung der Pluralform „Schreibungen“. *Lingo* schreibt im Falle der Identifizierung die Grundform bzw. Grundformen hinter dem Komma des Eintrags zusammen mit dem Kürzel für die Wortklasse in das Indexierungsergebnis. Über die Wortklassenkennung kann später eine Selektion der Ergebnisse erfolgen, so dass z.B. bestimmte Wortklassen gar nicht erst in das Indexierungsergebnis übernommen werden, weil sie als Indexterm evtl. weniger relevant sind (Verbinfinitive sind z.B. nur in Ausnahmefällen sinnvolle Suchbegriffe).

*Lingo* verfolgt auch in der Differenzierung von Wortklassen ein möglichst einfaches Konzept und unterscheidet lediglich die folgenden Wortklassen:

```
Substantiv          = 's'
Adjektiv            = 'a'
Verb                = 'v'
Eigenname           = 'e'
Kompositum          = 'k'
Mehrwortbegriff     = 'm'
Wortform            = 'w'
Synonym             = 'y'
Takeitasis          = 'x'
Unbekannt           = '?'
```

Neben dem Systemwörterbuch für die Identifizierung gibt es ein bereits angelegtes Benutzer-Grundformenwörterbuch (user01_dic), in das neue Einträge (unbekannte Wörter) aufgenommen werden können; das Format entspricht dem des Systemwörterbuchs:

```
bibliografie,bibliografie #s
```

Das Anlegen weiterer Benutzerwörterbücher ist problemlos möglich, deren Einbindung in die Indexierung wird über eine zentrale Konfigurationsdatei gesteuert. Die einfache Form des Wörterbucheintrags und der Verzicht auf eine stark differenzierende grammatikalische Spezifikation im System erlaubt die Erweiterung bzw. Überarbeitung von Wörterbüchern auch ohne vertiefte sprachliche Kenntnisse.

## 2.2 Kompositumerkennung

Die Kompositumerkennung bzw. –zerlegung von *lingo* greift ebenfalls auf die Identifizierungswörterbücher zu und versucht unbekannte Zeichenketten dadurch zu erkennen, dass mögliche Kompositumbestandteile in den Wörterbüchern gesucht werden. Dabei wendet *lingo* eine einfache Strategie an: Falls eine Zeichenkette aus der Grundformerkennung als unbekannt hervorgeht, versucht *lingo* den von Rechts längstmöglichen Bestandteil (longest matching) in der Zeichenkette zu identifizieren:

```
Informationswirtschaft  →     „wirtschaft“
```

Die Regel des longest matching verhindert, dass nicht zunächst das Substantiv „Schaft“ erkannt wird und vermeidet damit eine mögliche Überidentifizierung. Die Identifizierung des hinteren Teils eines Kompositums (des Kompositumkopfes) berücksichtigt dabei die wortklassenspezifischen Suffixmuster wie in der Grundformerkennung. Für die Erkennung des vorderen Teils eines Kompositums ist dies nicht möglich, weil für den ersten Teil des Kompositums andere Endungen möglich sind, die sog. Fugungen. Das Substantiv „Information“ kennt alleine stehend nicht die Endung „s“, diese bekommt es als Fugung nur im Kompositum. Aus diesem Grund wird für die Erkennung von vorderen Kompositumbestandteilen eine eigene Liste mit im Deutschen erlaubten Fugungen herangezogen. Dies ermöglicht im Beispiel die Identifizierung von „Information“ + Fugung „s“.

Da ein Wörterbucheintrag eines Kompositums dessen Zerlegung verhindert – in die Kompositumzerlegung gelangen ja nur unbekannte Zeichenketten –, sind die in den ursprünglichen Wörterbuchquellen vorhandenen Komposita systematisch entfernt worden, um eine höhere Zerlegungsquote zu erreichen. Umgekehrt ist es natürlich einfach möglich, durch die Lexikalisierung von Komposita deren unerwünschte Zerlegung zu verhindern, z.B. wird die formal mögliche Zerlegung von „Wirtschaft“ in „Wirt“ und „Schaft“ durch den Grundformeintrag „Wirtschaft“ im Lexikon verhindert. Prinzipiell können erwünschte Zerlegungen jedoch auch durch Lexikalisierung erreicht werden, um z.B. eine Überzerlegung langer Komposita zu verhindern („Datenbankmanagementsystem“ statt in „Daten“, „Bank“, „Management“, „System“ nur in „Datenbank“ und „System“) oder eine vollständige, jedoch nicht gewünschte Zerlegung zu vermeiden („Wahnsinn“ statt in „Wahn“ und „Sinn“ nur in „Wahn“). Lexikalisierte Dekompositionen wären ähnlich zu behandeln wie Synonymrelationen und könnten im Wörterbuch folgende Form haben:

```
datenbankmanagementsystem*datenbank,managementsystem
wahnsinn*wahn,sinn
```

Grundsätzlich verfolgt *lingo* aber nicht den Ansatz, Komposita im Sinne einer korrekten Zerlegung mit ihren Zerlegungen systematisch zu lexikalisieren.

## 2.3 Relationierung von Synonymen

Die Zuordnung von Synonymen zu erkannten Grundformen wird über Synonymwörterbücher realisiert, die als Wörterbuchquelle zwei Typen zulassen:

- der Typus Äquivalenzklasse enthält Einträge in der Form

  ```
  Abschreibung;Steuerabschreibung;Wertverminderung
  ```

*lingo* generiert daraus eine Datenbank, in der jeder Eintrag einer Äquivalenzklasse mit allen anderen Einträgen relationiert wird.

- der Typus einfache Relation enthält nur zweiseitige Einträge in der Form

  ```
  abdominalchirurgie*bauchchirurgie
  ```

Eine Wortklassenkodierung kann für beide Typen entfallen, die Einträge bekommen bei der Generierung der Wörterbuch-Datenbank die Standard-Synonymwortklasse „#y“ vom System zugewiesen.

Das mit dem System ausgelieferte Synonymwörterbuch „openthes“ ist vom Typus „Äquivalenzklasse“ und enthält als Thesaurus im Sinne einer Schreib- und Formulier-

hilfe zahlreiche Synonymrelationen, die für die Nutzung als Indexterme kaum geeignet sind. Insgesamt sind die Einträge für diesen Zweck zu allgemeinsprachlich und die Synonymrelationen zu weit gefasst.

Das darüber hinaus vorhandene zweite Synonymwörterbuch „system_syn“ enthält beispielhaft Synonymeinträge vom Typus einfache zweiseitige Relation, allerdings in der mitgelieferten Variante nur Synonyme mit dem Anfangsbuchstaben „A“.(10)

Für ernsthafte Anwendungen der automatischen Indexierung ist es daher zweckmäßig, evtl. vorhandene eigene kontrollierte Vokabularien („echte“ Thesauri) als *lingo*-Wörterbuch aufzubereiten und einzubinden. Der einfache Listencharakter der Synonymwörterbuchtypen und der Verzicht auf eine Wortklassenkennung kommen dem entgegen.

Die Synonymrelationierung setzt auf erkannten Grundformen bzw. erkannten Komposita bzw. erkannten Mehrwortgruppen auf, d.h. es muss eine Beziehung zwischen den Einträgen im Grundformwörterbuch und im Synonymwörterbuch bestehen. Vorbereitend müssen daher unbekannte Wörter aus dem Synonymbestand lexikalisiert werden (für Komposita, deren Bestandteile bekannt sind, ist dies nicht nötig).

## 2.4 Erkennung von Mehrwortgruppen

*Lingo* kennt zwei Verfahren für die Erkennung von Mehrwortgruppen, die jeweils einen eigenen Nutzen haben. Die lexikalische Mehrworterkennung (der sog. „multiworder“) setzt auf einem Wörterbuch mit Mehrwortbegriffen auf, die algorithmische Mehrworterkennung analysiert die Abfolge von Wörtern in Sätzen hinsichtlich ihrer Wortklassen. Beide Ansätze gemeinsam ermöglichen eine weit reichende Erkennung von Mehrwortgruppen in Dokumenten für die Bereitstellung als Indexterme aber auch als Voraussetzung für z.B. die Relationierung von Synonymen (s.o.).

Die lexikon-basierte Mehrworterkennung stützt sich auf ein oder mehrere Wörterbücher, in denen Mehrwortgruppen systematisch in folgender Form erfasst werden:

```
abfallrelevante datenbank*abfallrelevante datenbank
```

Beim Aufbau der Wörterbuch-Datenbank wird der erste Wortlaut (Eintrag vor dem *) durch eine *lingo*-Indexierung zunächst auf die Grundform gebracht, um eine Erkennung auch bei flektierten Varianten im Text zu ermöglichen. Die Mehrworterkennung greift während der Indexierung also auf folgenden Wörterbucheintrag zu:

```
abfallrelevant datenbank*abfallrelevante datenbank
```

Dadurch werden die Varianten „abfallrelevante Datenbank“, „abfallrelevante Datenbanken“, „abfallrelevanten Datenbank“ und „abfallrelevanten Datenbanken“ korrekt identifiziert und der Indexterm „abfallrelevante Datenbank“ als Mehrwort-Grundform generiert.

Der Nutzen einer Mehrworterkennung für das Information Retrieval ist in den meisten Indexierungskontexten als eher gering einzuschätzen, weil die gezielte Nutzung von Phrasen als Indexterme selten möglich bzw. sinnvoll ist. Anders ist dies, wenn eine Suche die Einbindung der Indexinhalte ausdrücklich unterstützt, mehrwortige Indexterme also für die Selektion verwendet werden können; dann lassen sich identifizierte und invertierte Mehrwortbegriffe zur Bedeutungsdifferenzierung im Index heranziehen:

```
datenbank, abfallrelevant       2
```

```
datenbank, objektorientiert    35
datenbank, relational         128
datenbank, volltext            23
```

*Lingo* unterstützt die Erstellung derartiger zusätzlicher Indexeinträge durch eine entsprechende Funktion zur Invertierung erkannter Mehrwortgruppen.

Ergänzend zum wörterbuchgestützten „multiworder“ identifiziert der sog. „sequencer“ Kandidaten für Mehrwortgruppen in Texten auf der Basis der Abfolge von Wortklassen im Text. Ausgangspunkt ist die These, dass im Deutschen alle Folgen von z.B. Adjektiv-Substantiv potenzielle Mehrwortgruppen sind. Über eine Selektion aller Gruppen, die einer solchen Mustervorgabe entsprechen, erhält man Listen von Dokument- bzw. Kollektionsspezifischen Mehrwortgruppen Diese können wiederum Grundlage für die Pflege des eigenen Mehrwortlexikons sein, d.h. die Qualität der lexikonbasierten Mehrworterkennung vergrößern. Voraussetzung für den sequencer ist eine erfolgte Grundformreduktion, die erkannte Grundformen immer auch mit den zugehörigen Wortklassen versieht. Über eine Konfiguration können dem sequencer die Muster für den Wortklassenabgleich übergeben werden, der folgende Eintrag

```
sequencer:
      sequences: [ [AS, "2, 1"], [AK, "2, 1"] ]
```

extrahiert z.B. alle Wortfolgen aus „Adjektiv, Substantiv“ und „Adjektiv, Kompositum“ und invertiert die extrahierten Einträge.

Besondere Bedeutung kommt der Erkennung von Mehrwortbegriffen für die Generierung von Synonymen zu, denn Einträge in Synonymwörterbüchern können durchaus Relationen des Typus

```
zweites deutsches fernsehen*zdf
```

enthalten, die nur über eine zuvor erfolgte lexikalische Mehrworterkennung zu nutzen sind.

## 2.5 OCR-Behandlung

Zunehmend werden Daten aus Digitalisierungsprozessen zu Inhalten von Information Retrieval-Lösungen. Obwohl die Qualität der Texterkennung inzwischen beachtlich ist, kommt es insb. bei älteren Vorlagen häufig noch immer zu zahlreichen Lesefehlern. Zur weitgehend automatischen Behebung dieser Lesefehler verfügt *lingo* über eine Korrekturfunktionalität, die lexikalisch nach möglichen korrekten Wortformen sucht, wenn sie auf unbekannte Zeichenketten trifft. Dabei bedient sich der sog. „ocr-variator“ einfacher Austauschmuster für Zeichenketten, die dann in der variierten Form erneut am Wörterbuch abgeglichen werden. Solche Austauschmuster können z.B. so aussehen:

```
ocr_variator:
  variations:
    - [ fch,  sch ]
    - [ fp,   sp  ]
    - [ fl,   st  ]
```

Die hier gezeigten Ersetzungsmuster ermöglichen dann z.B. folgende Identifizierungen:

„Flafche“ - „Flasche“

„Wefpe“ - „Wespe“

„Wefle“- „Weste“

Hintergrund dieser „f“-zu-„s“-Ersetzungen ist ein OCR-Ergebnis einer Vorlage mit älterer Typographie, die noch über ein sog. „langes s“ verfügt, dass vom OCR-Prozess häufig als „f“ erkannt wird. *Lingo* unterzieht im Indexierungslauf alle unbekannten Zeichenketten den Ersetzungen gemäß Variationsmustern (schrittweise in der angegebenen Reihenfolge) und überprüft das jeweilige Ergebnis im Wörterbuch. Bei erfolgreicher Identifizierung wird die Ersetzung durchgeführt.

## 2.6 Gewichtung

*Lingo* verfügt über umfangreich konfigurierbare Ausgabemöglichkeiten für die Indexierungsergebnisse. Aus dem kontinuierlichen Datenstrom aller erzeugten Indexterme können durch ein spezielles Modul, den „vector_filter“, spezielle Teilströme ausgefiltert werden, die in einzelnen Ergebnisdateien zur weiteren Nutzung im Retrieval-System abgelegt werden können. Während der Generierung der Ergebnisse kann mit *lingo* eine einfache Termgewichtung durchgeführt werden. Zur Verfügung stehen die einfache Termhäufigkeit (TF), d.h. die Häufigkeit eines Terms je Dokument, und die relative Termhäufigkeit (WDF), d.h. die Häufigkeit eines Terms in Relation zur Gesamtzahl der Terme eines Dokuments. Die Nutzung der Gewichtungswerte kann im Retrieval-System z.B. in Form eines Relevance-Rankings der Treffermenge erfolgen. *Lingo* erzeugt lediglich die dafür erforderlichen Termgewichte, für eine Nutzung dieser Gewichtungswerte sind weitere Vorkehrungen im Retrieval-System notwendig. Ein von *lingo* erzeugtes gewichtetes Ergebnis kann z.B. so aussehen:

```
66 erschließung
50 automatisch
38 bild
34 beschreibung
34 normierung
33 dokument
33 indexierung
26 form
26 retrieval
24 datei
...
```

## 2.7 Ergebnisaufbereitung

Für die Ausgabe der Indexterme verfügt *lingo* über flexible Konfigurationsmöglichkeiten. Die für die Ergebnisselektion und Ergebnisaufbereitung zuständigen module „vector_filter“ und „textwriter“erlauben eine freie Gestaltung von Ausgabedateien. So ist es z.B. möglich, die Ausgabe von Indextermen auf bestimmte Wortklassen zu beschränken (z.B. nur Substantive und Adjektive) oder unterschiedliche Wortklassen in getrennten Dateien auszugeben. Für die Verarbeitung von Quelldaten unterschiedlicher Formate stehen darüber hinaus zwei grundlegend verschiedene Grundkonfigurationen mit jeweils zugehörigen Ausgabekonfigurationen zur Verfügung. *Lingo* unterscheidet zwei Typen von zu indexierenden Daten: Einzeldokumente in Form von einzelnen Dateien, wobei immer eine Datei ein Dokument ist und Sammeldokumente in Form von großen Dokumenten mit einer Vielzahl einzelner Datensätze. Für Einzel-

dokumente, z.B. also viele Texte in einem Verzeichnis, generiert *lingo* als Standardausgabe einen sog Vektor, der alle Indexterme je Dokument in zu bestimmender Sortierung (vgl. das Beispiel in 2.6) enthält. Für Sammeldokumente im Datenbankformat generiert *lingo* eine Standardausgabe als comma-separated-file, um eine möglichst einfache Möglichkeit zu bieten, die Indexierungsergebnisse zu den ursprünglichen Daten hinzuzuspielen.

Beispiel einer Quelldatei als Sammeldokument:

```
[00001.]

020: Die Aufgabenteilung zwischen Wortschatz und Grammatik in
einer Indexsprache.

[00002.]

020: Nicht-konventionelle Thesaurusrelationen als
Orientierungshilfen für Indexierung und Recherche: Analyse
ausgewählter Beispiele.

[00003.]

020: PRECIS: ein englisches Indexierungsverfahren für deutsche
Bibliotheken?.

...
```

Ergebnisdatei im comma-separated Format (hier mir relativer Termfrequenz)(11):

```
00001*aufgabe {0.12500} aufgabenteilung {0.12500} aufgeben
{0.12500} einer {0.12500} grammatik {0.12500} index {0.12500}
indexsprache {0.12500} sprache {0.12500} teilung {0.12500}
wortschatz {0.12500}

00002*analyse {0.05263} ausgewählt {0.05263} beispiel {0.05263}
hilfe {0.05263} indexierung {0.05263} konventionell {0.05263}
nicht-konventionell {0.05263} orientierung {0.05263}
orientierungshilfe {0.05263} recherche {0.05263} relation
{0.05263} thesaurus {0.05263} thesaurusrelation {0.05263}

00003*verfahren {0.07692} bibliothek {0.03846} deutsch {0.03846}
englisch {0.03846} indexierung {0.03846} indexierungsverfahren
{0.03846}

...
```

Beide Grundkonfigurationen können leicht erweitert und verändert werden und so speziellen Einsatzumgebungen angepasst werden.

## 3. Systemarchitektur

Die Entwicklung von *lingo* erfolgte in erster Linie im Hinblick auf einen Einsatz in Forschung und Lehre, weshalb Transparenz und größtmögliche Konfigurierbarkeit als Zielvorstellungen im Vordergrund standen. *Lingo* ist vollständig in der objektorientierten Sprache ruby geschrieben, die als interpretierte Sprache unmittelbaren Einblick in den Programmcode erlaubt. Die Systemarchitektur ist weitestgehend modular, das gedankliche Vorbild folgt dem Baukastenprinzip, mit dem durch freie Kombinierbarkeit der Bestandteile eigene Systeme gestaltet werden können.

Eine *lingo*-Sitzung ist aufgebaut wie eine Besprechung mit mehreren Teilnehmern, die über eine Konfiguration „eingeladen“ werden. Die Fähigkeiten der eingeladenen Teilnehmer (attendees) bestimmen das Besprechungsergebnis, d.h. die Ergebnisse der

Indexierung. Eine typische Konfiguration (in minimaler Form zur Veranschaulichung) einer *lingo*-Sitzung sieht dann z.B. so aus:

```
meeting:
  protocol: '$(status)'
  attendees:
    - textreader:     { out: lines,  files: '$(files)' }
    - tokenizer:      { in: lines, out: token }
    - wordsearcher:   { in: token, out: words, source: 'sys-dic' }
    - decomposer:     { in: words, out: comps, source: 'sys-dic' }
    - sequencer:      { in: comps, out: sequ, stopper: 'PUNCT,OTH_C',
                        source: 'sys-mul' }
    - multiworder:    { in: sequ, out: multi, stopper: 'PUNCT,OTH_C',
                        source: 'sys-mul' }
    - synonymer:      { in: multi, out: split, skip: '?', source: 'sys-
                        syn' }
    - vector_filter:  { in: split, out: vector, lexicals: '[ksavem]',
                        sort: term_abs }
    - textwriter:     { in: vector, ext: vec, sep: "\n" }
```

Der normale Ablauf einer *lingo*-Sitzung beginnt üblicherweise mit dem Teilnehmer „textreader“, der eine Quelldatei zeilenweise einliest und das Ergebnis den anderen Teilnehmern zur Verfügung stellt. Da nicht alle Teilnehmer mit dieser Information etwas anfangen können, wird in der Konfiguration von *lingo* bestimmt, wer zielgerichtet mit wem redet (über in- und out-Attribute).

Der „textreader“ gibt die Zeilen in unserem Beispiel weiter an den „tokenizer“, der die Zeilen in Abhängigkeit von weiteren Konfigurationseinstellungen in einzelne „token“, d.h. Zeichenketten zerlegt – „token“ sind das Ausgangsmaterial für die spätere Wortanalyse. Diese übernimmt der „wordsearcher“, der die ihm übergebenen „token“ an den zugeschalteten Wörterbüchern überprüft und die Grundform ermittelt. Faktisch macht der „wordsearcher“ aus Zeichenketten Wörter. Noch nicht erkannte Zeichenketten werden vom „decomposer“ untersucht, der zunächst unterstellt, dass unbekannte Zeichenketten in Wirklichkeit Komposita sind, die aus bekannten Zeichenketten bestehen. Die beiden für Mehrwortbegriffe zuständigen attendees „sequencer“ und „multiworder“ untersuchen das bisherige Ergebnis lexikalisch und über Musterabgleich auf Syntagmen (vgl. 2.4), der „synonymer“ fügt lexikalisch bekannte Synonyme hinzu (vgl. 2.3). Aus der so generierten Ergebnismenge werden durch den „vector_filter“ die tatsächlich benötigten Ergebnisse herausgezogen, z.B. also alle Indexterme, die Substantiv sind. Der „textwriter“ schließlich schreibt dieses Filterergebnis in eine Ausgabedatei und beendet die Sitzung.

## 4. Einsatzmöglichkeiten und Systemgrenzen

Als System für Forschung und Lehre verfolgt *lingo* primär die Ziele, eine automatische Indexierung im konkreten Einsatz unter Laborbedingungen zu ermöglichen sowie ein leicht erweiterbares System für den Einsatz in Projekten bereitzustellen. Im Laboreinsatz bewährt sich *lingo* bereits seit längerem, im Projekteinsatz wird derzeit eine sehr aufwändige automatische Indexierung für das Reallexikon zur Deutschen Kunstgeschichte realisiert.(12) *Lingo* übernimmt dabei die linguistischen Basisfunktionen, generiert also grammatikalisch normierte Indexterme, wird aber auch in größerem Umfang zur Identifizierung von Mehrwortgruppen verwendet, um anhand umfangreicher normierter Vokabulare Personennamen, Werktitel und Geografika zu identifizieren.(13)

Stabilität, Offenheit und Leistungsumfang des Systems ermöglichen auch einen Einsatz in Produktivumgebungen. Durch die derzeitige Beschränkung auf die Verarbeitung deutscher Sprache ist die Einbindung der *lingo*-Funktionalität zumindest für alle Retrieval-Umgebungen mit ausschließlich oder überwiegend deutschsprachigen Dokumentkollektionen sinnvoll. Der Nutzen einer solchen Indexierung ist als Recall-erhöhendes Instrument unbestritten, der technische wie administrative Aufwand für die Indexierung dabei vergleichsweise gering.

*Lingo* stellt als Indexierungssystem keine Datenbank- oder Retrievalfunktionalität bereit, sorgt vielmehr auf der Datenseite für eine Verbesserung bereits vorhandener Funktionalitäten durch die Bereitstellung zusätzlicher, grammatikalisch normierter Indexterme. Dabei bleibt die linguistische Verarbeitung immer auf der sprachlichen Ebene, d.h. unterhalb der Semantik: eine semantische Differenzierung der Grundformen kann z.B. nicht erfolgen. Beispielsweise wird nicht zwischen "Schloss" (Schließvorrichtung) und "Schloss" (Gebäude) unterschieden, auch wenn dies auf Grund des Titelkontexts intellektuell natürlich möglich wäre.

Die Nachteile einer linguistischen automatischen Indexierung sind gering, diese bestehen v.a. in der Möglichkeit einer sog. Überindexierung hervorgerufen durch potenzielle Mehrdeutigkeit eines Wortes im Quelltext; "Weinen" wird beispielsweise stets (wegen der Substantivierungsmöglichkeit auch im Satzinneren) auf "weinen" (Verb) und "Wein" (Substantiv) abgebildet. Ähnlich wie die nicht erwünschte Zerlegung von Eigennamen, die formal auch Kompositum sein können - „Silberstein“ in „Silber“ und „Stein“ - entstehen durch derartige „Fehler“(14) Indexterme, die im schlimmsten Fall für eine Verschlechterung der Präzision dadurch sorgen, dass eine Suche nach „Silber“ nun auch ein Dokument mit „Silberstein“ findet. Nach allen Erfahrungen und verschiedenen Retrievaltests(15) besteht allerdings kein Grund, diesen Effekt überzubewerten.

Die *lingo*-Indexierung kann eine intellektuelle Verschlagwortung, die den ganzen Dokumentinhalt für eine verdichtende Indexierung heranzieht, nicht ersetzen sondern nur ergänzen. Dass eine Kombination beider Verfahren die optimale Lösung darstellt, ist eine allgemeine Erkenntnis der Retrievalforschung. Umgekehrt haben alle Retrievaltests eindeutig belegt, dass eine linguistisch basierte Indexierung der "reinen" Stichwort-Wortformen-Indexierung (Vollinvertierung) in jeder Hinsicht überlegen ist.

## 5. Erweiterungen des Systems

Durch seinen offenen und transparenten Aufbau ist *lingo* für das Deutsche mit relativ einfachen Mitteln um weitere Funktionalitäten zu erweitern und hinsichtlich seiner Wörterbuchbasis beliebig auszubauen. Anpassungen für spezifische Einsatzumgebungen und Indexierungsziele, z.B. fachspezifische Datenkollektionen mit bereits vorhandenen Erschließungsterminologien, sind mit vertretbarem Aufwand zu realisieren.

Auch die Ausweitung des Systems auf eine Indexierung weiterer Sprachen ist vergleichsweise wenig aufwändig. Die grundlegende lexikalische Arbeitsweise mit Grundformenwörterbüchern und Suffixlisten ist grundsätzlich auf weitere Sprachen übertragbar. Voraussetzung für die Implementierung zusätzlicher Sprachen ist lediglich das Vorhandensein einer Wörterbuchbasis mit Wortklassenkennungen. Eine *lingo*-Implementierung für das Englische ist eines der nächsten Ziele. Die Entwickler erhoffen sich, dass die Verfügbarkeit von *lingo* als open source zu entsprechenden Aktivitäten der Community führen wird.

### Anmerkungen:

(1) Lepsky, K., J. Siepmann u. A. Zimmermann: Automatische Indexierung für Online-Kataloge: Ergebnisse eines Retrievaltests. In: Zeitschrift für Bibliothekswesen und Bibliographie. 43(1996) H.1, S.47-56.

(2) Voraussetzung für einen solchen Algorithmus ist natürlich, dass die lexikalische Basis der automatischen Indexierung über entsprechende Wortklasseninformationen verfügt.

(3) Die aktuelle *lingo*-Version kann unter http://www.lex-lingo.de heruntergeladen werden.

(4) Freeling 1.2: http://garraf.epsevg.upc.es/freeling/.

(5) http://ruby-lang.org/en

(6) Sdbm und ruby vgl.: http://www.ruby-doc.org/stdlib/libdoc/sdbm/rdoc/index.html.

(7) http://www.lezius.de/wolfgang/morphy/

(8) http://lingucomponent.openoffice.org/spell_dic.html

(9) http://www.openthesaurus.de

(10) Die im „system_syn“ verwendeten Synonymrelationen stammen aus der Schlagwortnormdatei (SWD) und sind nur unvollständig intellektuell überarbeitet.

(11) Im Original befindet sich die Daten zu jedem Datensatz auf einer neuen Zeile, die Umbrüche sind hier durch das Layout bedingt.

(12) Im von der Deutschen Forschungsgemeinschaft geförderten Projekt „RDK-Web: Erstellung einer WEB-Version des Reallexikons zur Deutschen Kunstgeschichte" wird in Zusammenarbeit zwischen dem Zentralinstitut für Kunstgeschichte, München, und dem Institut für Informationswissenschaft, Fachhochschule Köln, eine digitale automatisch erschlossene Version des Lexikons erstellt.

(13) Vgl. Lepsky, Klaus: Ist automatische Normierung möglich? In: Regelwerke für die Sacherschließung – sexy oder uncool? Hrsg. von Joern Sieglerschmidt. Berlin 2004, S. 40-50. (http://titan.bsz-bw.de/cms/service/museen/publ/eva71lepsky4.pdf).

(14) Eigentlich handelt es sich bei diesen Beispielen nicht um „Fehler“, weil die Manipulationen sprachlich begründet sind – dies kann also durchaus im Kontrast zu Erwünschtheit eines Indexterms stehen.

(15) Lepsky, K., H.H. Zimmermann: Katalogerweiterung durch Scanning und Automatische Dokumenterschließung: Das DFG-Projekt KASCADE. In: ABI-Technik. 18(1998) H.1, S.56-60. Gödert, W., K. Lepsky: Semantische Umfeldsuche im Information Retrieval. In: Zeitschrift für Bibliothekswesen und Bibliographie. 45(1998) H.4, S.401-423. Sachse, E., M. Liebig u. W. Gödert: Automatische Indexierung unter Einbeziehung semantischer Relationen: Ergebnisse des Retrievaltests zum MILOS II-Projekt. Köln: FH Köln, Fachbereich Bibliotheks- und Informationswesen 1998. 66 S. (Kölner Arbeitspapiere zur Bibliotheks- und Informationswissenschaft; Bd.14)

**Anschriften der Verfasser:**

Prof. Dr. Klaus Lepsky
Institut für Informationswissenschaft
Fachhochschule Köln
Claudiusstraße 1
50678 Köln

klaus.lepsky@fh-koeln.de

John Vorhauer
Gustavstraße 6
50937 Köln

john@vorhauer.de