

# Inkrementelle Thesauri am Beispiel von Spam- und Phishing-Mails

Magdalena Koj, Tatsiana Maleika, Sviatlana Danilava

JW Goethe-University Frankfurt am Main, Dept. of Computer Science

Robert-Mayer-Str. 11-15, 60486 Frankfurt am Main, Germany

Email: magda.koj@gmx.net, maleika@cs.uni-frankfurt.de, danilava@cs.uni-frankfurt.de

## Abstract

Diese Arbeit behandelt den Aufbau eines inkrementellen Thesaurus, der sich speziell auf das Thema Spam bezieht. Dabei werden in dieser Dokumentation die aktuellen Entwicklungen sowie die eigene Entwicklung eines inkrementellen Spam-Thesaurus dargestellt.

## 1 Einleitung

Sieht man die enorme Bandbreite von Spams, dann wird klar, dass ein Thesaurus bei der Bekämpfung hilfreich sein kann. Die Spam-Mails ändern ihre Form und Inhalt sehr schnell, um von den Spam-Filtern nicht erkannt zu werden. Wenn ein Thesaurus bei der Entscheidung helfen soll, eine E-Mail als Spam oder Nicht-Spam zu behandeln, muss er auch stets auf dem neusten Stand sein, und dies ist bei dem inkrementellen Ansatz möglich. Der Thesaurus baut sich selbst auf, findet Zusammenhänge und hat somit keinen festen Endzustand. Weiterhin ermöglicht er eine schnelle Abarbeitung von großen Datenmengen und benötigt kein Expertenwissen oder menschliches Input. Doch birgt ein inkrementeller Thesaurus auch Probleme. Um sich selbst zu generieren, muss die Maschine fähig sein Texte zu verstehen und zu unterscheiden (in Spam und Nicht-Spam). Weiterhin kann auf Expertenwissen nicht vollständig verzichtet werden, da Maschinen auf schon gewonnenes Wissen und Erfahrungswerte nicht zurückgreifen können. Es wird zunächst untersucht, was bisher zu diesem Thema entwickelt wurde. Im Weiteren wird der von uns implementierte inkrementelle Spam-Thesaurus beschrieben. Der Einstieg in das Thema führt zuerst über eine grundlegende Begriffsklärung. Daher folgen nun einige Definitionen, auf die zukünftig Bezug genommen werden soll.

Der Begriff *Thesaurus* ([9]) stammt aus dem griechischen (*thesauros*: *Schatz*, *Schatzhaus*) und kann mit *Wortnetz* übersetzt werden. Linguistisch gesehen ist ein Thesaurus eine Sammlung von Wörtern, die in irgendeiner Art und Weise miteinander Verbunden sind, z.B. ein Schlagwort und alle zugehörigen Synonyme und Antonyme. Der erste dieser Art entstand 1852, *Roget's Thesaurus of English Words and Phrases* von Peter Mark Roget. Ein weiteres Einsatzgebiet der Thesauri ist

das Information Retrieval. Dabei werden alle zu einem Fachgebiet gehörenden Dokumente mit Schlag- und Stichwörtern versehen, sie werden Indexiert. Im Thesaurus werden diese Indizes miteinander vernetzt. Dabei entsteht ein kontrolliertes Vokabular, welches durch hierarchische, Assoziations- und Äquivalenzrelationen miteinander verbunden ist. Folgende *Relationsarten* sind in der DIN 1463-1 bzw. in der ISO 2788 Norm festgelegt:

DIN Norm	ISO Norm
BF - Benutzt für	UF - Used for
BS - Benutze Synonym	USE/SYN - Use synonym
OB - Oberbegriff	BT - Broader term
UB - Unterbegriff	NT - Narrower term
VB - Verwandter Begriff	RT - Related term
SB - Spitzenbegriff	TT - Top term

Ein inkrementeller Thesaurus ist ein Thesaurus der sich selbst aufbaut und aktualisiert. Dabei hat er keinen Endzustand, wie eine bestimmte Wortmenge. Jedes Wort, welches als wichtig erkannt wird, wird in den Thesaurus übernommen, die Relationen zu anderen Wörtern werden festgestellt und immer wieder auf den neusten Stand gebracht.

*Spam* ([9]) ist die massenhafte Verbreitung von unerwünschten Nachrichten (meist Werbung) via E-Mail, News- und Mailinglisten. Mittlerweile haben sich Unterarten wie **SPIN** (*Spam over Instant Messaging*, wie z.B. ISQ) und **SPIT** (*Spam over Internet Telephony*) entwickelt, dies sei aber nur am Rande erwähnt. Der Begriff Spam kommt aus dem englischen und bedeutet 'Spiced beef and ham', dies ist Dosenfleisch, das meist kalt als Brotbelag verzehrt wird. Die Bezeichnung der E-Mail-Überflutung als Spam geht auf einen Sketch der Serie *Monty Python's Flying Circus* aus dem Jahr 1970 zurück. Darin wird in einem Cafe jedes Gericht mit Spam (also Dosenfleisch) serviert; das Wort Spam wird dabei rund 132-mal genannt. Eine Spam-Flut sozusagen. Da wir uns bei diesem Praktikum ausschließlich mit E-Mail-Spam auseinandersetzen, soll hier diese Art des Spammens vorgezogen werden. E-Mail-Spam wird auch als **UBE** (*unsolicited bulk e-mail*: unerwünschte massenhafte E-Mail) bezeichnet. Dabei unterscheidet man folgende Unterarten:

- **UCE** (*unsolicited commercial e-mail*) ist Werbung, deren Inhalt oft Pharmazeutika (Viagra), Software oder dubiose medizinische Eingriffe (Penisverlängerungen) anpreist.
- **SCAM** (*engl. Betrug*) ist E-Mail-Betrug. Es handelt sich hier um ein Schneeball-System in dem Leute um ihr Geld gebracht werden; vor allem fällt hier die Nigeria-Connection auf.
- **Phishing** (*(Zusammengesetzt aus dem englischen Wort fishing für abfischen und dem ph aus Phreaking, um die Hinterhältigkeit der Tat anzuzeigen)*) bezeichnet eine Form des Trickbetrugs per E-Mail, in der es um das Erlangen von vertraulichen Daten geht.
- **Würmer und Vieren** werden in großen Mengen verschickt um möglichst viele Systeme zu beeinträchtigen oder zu zerstören.
- **Joe Jobs** sind Spam-Mails die von der Adresse einer dritten Person verschickt werden, um dieser Person zu schaden.
- **HOAXes** (*engl. Scherz, Streich*) sind Falschmeldungen, man könnte sie auch digitale Zeitungsenten nennen. Ihr Inhalt hat oftmals eine solche Brisanz, dass der Empfänger diese E-Mails an viele seiner Bekannten weiterleitet.

## 2 State of the Art

Bei der Suche nach Unterlagen zum Thema Spamthesaurus haben wir festgestellt, dass die Anwendung von Thesaurusen als Spamfilter nicht gängig ist. Vielmehr wird oft darauf hingewiesen, dass ein Thesaurus für die Breite der Spams nicht praktikabel ist, da er nicht so effizient und treffsicher arbeiten könne, wie die zur Zeit angewendeten Spamfilter. Trotzdem haben wir ein paar interessante Arbeiten zu Thema gefunden, die kurz vorgestellt werden. Einige dieser Ideen wurden in unserem eigenen Spamthesaurus aufgegriffen.

### 2.1 Domänenspezifische Thesauri

Im Internet gibt es immer mehr Projekte, die sich mit der Erstellung von Thesauri beschäftigen. Eines der größten ist *WordNet* ([10]). Es ist ein lexikalisches Referenzsystem, also ein Wörterbuch. Es enthält Nomen, Verben, Adjektive und Adverbien der englischen Sprache. Diese sind in Klassen, so genannte Sets, geteilt und werden durch verschiedene Relationen, wie Synonymie und Homonymie, miteinander verbunden. Die Benutzung von *WordNet* ist kostenlos. Das Deutsche Pendant dazu ist *Germanet* ([2]). Allerdings ist die Benutzung kostenpflichtig. Der Inhalt und Aufbau sind ähnlich. *OpenThesaurus* ([5])

ist ein Open Source Projekt in diesem Gebiet. Die Anmeldung und Benutzung ist, wie bei *WordNet*, kostenlos. *OpenThesaurus* steht in den Sprachen deutsch, polnisch, spanisch, slowakisch und norwegisch zur Verfügung.

### 2.2 Generating a Domain-specific Thesaurus Automatically: An Experiment on FlyBase

Dies ist eine Arbeit von *Hsinchun Chen, Bruce Schatz, Joanne Martinez* und *Tobun Dorbin* ([3]). Sie haben auf der Basis der Datenbank *FlyBase* einen automatischen, domänenspezifischen Thesaurus zum Thema Fruchtfliegen (*Drosophila*) erstellt, um das Suchen in wissenschaftlichen Ausarbeitungen zu erleichtern. Diese Arbeit stellt eine der Möglichkeiten dar, wie man domänenspezifische Thesauri automatisch erstellen kann. Der Aufbau erfolgt in drei Schritten:

#### 2.2.1 Object Filtering

Die Texte werden nach Schlüsselwörtern durchsucht, die in domänenspezifischen kontrollierten Listen festgehalten sind. Um die Artikel zu filtern werden vier Listen mit den Domänen Gennamen, Funktionsnamen, Author und Themengebiet benutzt. Die gefundenen Dokumente gelten als Kandidaten für die Indexierung.

#### 2.2.2 Automatic Indexing und Term Weighting

Bei der *automatischen Indexierung* sollen aus den Abstracts der Dokumente Wörter oder Wortpaare extrahiert werden, die nähere Informationen über das Dokument liefern. Es wird wie folgt vorgegangen:

- **Wörter identifizieren:** Dabei werden mit Hilfe von Wörterbüchern domänenspezifische Eigenheiten festgehalten.
- **Stop-wording:** Es werden alle Wörter entfernt, die nicht als Index verwendet werden, da sie thematisch nicht wichtig sind, wie z.B. *und, auf, von, mit, gehen, lesen* usw. Dazu werden Stopwörterlisten genutzt.
- **Term-phrase formation:** Die restlichen Wörter setzt man miteinander in Zusammenhang, indem aus benachbarten Wörtern, die in Titeln und Abstracts zu finden sind, Phrasen von bis zu 3 Wörtern bildet.
- **Berechnung der Term- und Dokument-Frequenz:** Dabei sagt die Termfrequenz  $tf_{ij}$  aus, wie oft ein Term  $j$  in Dokument  $i$  vorkommt und die Dokumentfrequenz  $df_j$  zeigt, in wie vielen von  $n$  Dokumenten Term  $j$  vorkommt. Phrasen, die im Dokumententitel stehen, haben ein höheres Gewicht als solche im Abstract oder im Text, da sie eine größere Aussagekraft besitzen.

- **Gewichtsberechnung:** Aus den oben genannten Frequenzen wird das kombinierte Gewicht  $d_{ij}$  von Term  $j$  in Dokument  $i$  berechnet. Dies geschieht anhand folgender Formel:

$$d_{ij} = tf_{ij} \times \log \left( \frac{N}{df_i} \times w_j \right) \quad (1)$$

### 2.2.3 Co-Occurrence Analysis

Nun wird berechnet, mit welcher Wahrscheinlichkeit Wörter gemeinsam auftreten. Dazu wird für jedes Wortpaar ein *Clustergewicht*  $CW(ClusterWeight)$  berechnet:

$$CW(T_j, T_k) = \frac{\sum_{i=1}^n d_{ijk}}{\sum_{i=1}^n d_{ij}} \times WeightingFactor(T_k) \quad (2)$$

$$CW(T_k, T_j) = \frac{\sum_{i=1}^n d_{ikj}}{\sum_{i=1}^n d_{ik}} \times WeightingFactor(T_j) \quad (3)$$

Dabei werden die Kombinationen beider Richtungen berücksichtigt. Die Werte von  $d_{ij}$  und  $d_{ik}$  werden mit (1) berechnet.  $d_{ijk}$  steht für das kombinierte Gewicht von Term  $T_j$  mit Term  $T_k$  bzw.  $d_{ikj}$  für das kombinierte Gewicht von Term  $T_k$  mit Term  $T_j$ .

$$d_{ijk} = tf_{ijk} \times \log \left( \frac{N}{df_{jk}} \times w_j \right) \quad (4)$$

$tf_{ijk}$  zeigt dabei, wie häufig die Terme  $T_j$  und  $T_k$  gemeinsam in Dokument  $i$  vorkommen.  $df_{jk}$  steht für die Anzahl von Dokumenten in denen die Terme  $T_j$  und  $T_k$  gemeinsam vorkommen.

Zuletzt werden allgemeine Wörter, die in Texten sehr oft vorkommen, durch eine kleinere Gewichtung nach hinten gesetzt; diese Terme haben einen hohen  $df_k$  – Wert. Dadurch rücken die beschreibenden Terme an eine vordere Stelle.

$$WeightingFactor(T_k) = \frac{\log \frac{N}{df_k}}{\log N} \quad (5)$$

$$WeightingFactor(T_j) = \frac{\log \frac{N}{df_j}}{\log N} \quad (6)$$

## 3 Entwicklung eines inkrementellen Spamthesaurus

Ausgehend von der State of die Art stellt sich die Frage nach der eigentlichen Vorgehensweise. Fakt ist, dass

schon ein Dutzend 'gute' Wörter in einer Spam-Mail die Wahrscheinlichkeit bis auf 50% erhöhen, dass sie vom Filter nicht aufgehalten wird. Das erschwert die Suche nach 'typischen' Wörtern und Phrasen, die in Spam-Mails vorkommen. Für die Entwicklung wurden uns 13.298 anonymisierte Spam-Mails zur Verfügung gestellt, welche wir als Testmenge nutzen konnten. Als Grundlage haben wir das Verfahren aus Kapitel 3.2 genommen. Durch die zeitliche Begrenzung des Praktikums war uns die Erstellung eines allgemeinen Spamthesaurus nicht möglich. Daher die Entscheidung den Thesaurus nur aus den Subjectlines der Spam-Mails zu erstellen; außerdem ist er nur auf E-Mails der Spamdomeäne *Pharmazie* und, da es eng daran geknüpft ist, *Sex* beschränkt.

### 3.1 Datenvorbereitung

Wie bereits erwähnt bestand die Testmenge aus 13.298 E-Mails bzw. Subjectlines. Im ersten Schritt wurden alle Subjectlines von dem Rest der E-Mails (Adresslines, Headers, Text-Bodies etc.) getrennt. Weiterhin wurden alle Duplikate, leeren Subjectlines und Zeilen mit dem Inhalt *Delivery Failure* oder nur Zahlen entfernt. Dadurch schrumpfte die Testmenge auf 'nur noch' 8.701 Subjectlines. Im nächsten Schritt wurden alle nicht-pharmazeutischen (beispielsweise Finanz- und Computerbereich) Subjectlines entfernt. Dafür wurde eine Liste aller Wörter erstellt, so dass die Subjectlines, die diese Wörter enthalten, entfernt werden konnten. Dabei waren die vielen Spelling-Variationen eines der Hauptprobleme. Um diese zu erkennen mussten für manche Wörter reguläre Ausdrücke geschrieben werden. Die somit entstandene Testmenge von *Pharmazie*- und *Sex*-Spam betrug 6.000 Subjectlines. Die Vielfalt an Spelling-Varianten musste auch in der 'puren' Testmenge beseitigt werden. Dies geschah wiederum mit Hilfe von regulären Ausdrücken. Als Folge der Angleichung mussten erneut alle Duplikate entfernt werden, danach enthielt die Testmenge 5.786 Subjectlines. Nun wurden alle Subjectlines in ihre einzelnen Wörter zerlegt. Da nach der Anwendung der regulären Ausdrücke viel Zeichenabfall entstanden ist, wurden alle Zeichenfolgen der Länge  $\leq 2$  entfernt, da diese keine relevante Bedeutung und teilweise auch keinen Sinn hatten. Außerdem wurde eine Blacklist manuell erstellt. Die wird später benutzt, um festzustellen, ob alles, was einem Menschen auffällt, auch von der Maschine statistisch berechnet werden kann.

### 3.2 Stop-Wording

Wie schon in Abschnitt 3.2.2 erwähnt, war das Entfernen von Stopwörtern nötig, da diese zwar sehr häufig in Texten vorkommen, aber nichts zum Inhalt beitragen. Die dazu benutzte Stop-Wort-Liste haben wir manuell erstellt und dann implementiert.

### 3.3 Porter Stemmer

Anschließend wurden alle Wörter dem Stemming unterzogen. Dabei wird jedes Wort durch das Programm auf seinen Wortstamm zurückgeführt und in der Stammform gespeichert. Für dieses Programm haben wir den Stemming-Algorithmus von Martin Porter (1979, Cambridge University) genutzt ([6]). Es soll angemerkt werden, dass dieser Stemmer keineswegs optimal ist. Er hat einige Fehler und Unvollständigkeiten; manche von ihnen wurden für unsere Zwecke durch Anpassungen behoben. Allerdings blieb für eine umfangreiche Optimierung keine Zeit.

### 3.4 Apriori

Der Apriori-Algorithmus (Rakesh Agrawal et al., IBM Research Lab, Almaden) ([1]) dient der Erstellung von Worttupeln und den zugehörigen Regeln. Zur Funktionsweise: Jede gestemnte Subjectline wird von links nach rechts eingelesen. Dabei müssen alle Wörter bzw. Worttupel  $\geq 5$  Mal auftreten, damit sie vom Programm zur weiteren Verarbeitung genutzt werden. 5 ist eine intuitiv gewählte Grenze für unsere Testmenge; ein Wort bzw. Worttupel, welches in 5 Spam-Mails vorkommt, wird wahrscheinlich auch in Zukunft verwendet werden. Aus den gefundenen einzelnen Wörtern wurden Tupel erstellt, die in alphabetischer Sortierung vorliegen. Dabei wird in jedem Schritt die Länge der Tupel um 1 erhöht. Dies geschieht in einer Schleife des Prüfens und Tupelgenerierens so lange, bis keine weiteren Tupel mehr gefunden werden können. Die Bildung von Tupeln erfolgt nur, wenn die ersten  $n - 1$  Glieder beider Tupel gleich sind. Das soll die Hypothese überprüfen, dass wenn das Wort  $a$  mit dem Wort  $b$  zusammen vorkommt ( $Tupel(a, b)$ ) und das Wort  $a$  mit dem Wort  $c$  zusammen vorkommt ( $Tupel(a, c)$ ), dann auch die drei Wörter zusammen vorkommen können.

$$\begin{aligned} a_1, \dots, a_{n-1} &= b_1, \dots, b_{n-1} \\ a_n &\neq b_n \end{aligned}$$

Ein neues Tupel sieht wie folgt aus:

$$a_1, \dots, a_n, b_n$$

Die Tabelle zeigt die Ergebnisse unseres Apriori-Algorithmus:

Tupellänge	Gesamt Tupel	Vorkommen $\geq 5$
1	5.198	654
2	9.255	454
3	587	355
4	501	494
5	560	560
6	448	448
7	248	248

Das Wort, welches am häufigsten gezählt wurde ist *Via-gra* mit 243 Einträgen in Subjectlines.

### 3.5 Clustering

Nach der Generierung der verschiedenen Tupel folgte die Gewichtung, wobei nur Gewichte von Termen der Länge 1 und 2 berechnet wurden. Es werden nur Beziehungen von Wort zu Wort und Wortpaar zu Wort betrachtet. Dies geschah auf Grund der Auswertung der Apriori-Ergebnisse, bei der wir gesehen haben, dass ab einer Worttupellänge von  $\geq 4$  nur noch Medikamentennamen genannt werden. Es gelten folgende Definitionen:

$tf_{ij}$ : Häufigkeit von Term  $j$  in Dokument  $i$

$df_j$ : Häufigkeit von Term  $j$  in Dokumentenmenge  $n$

$d_{ij}$ : kombiniertes Gewicht von  $j$  und  $i$

$d_{ijk}$ : kombiniertes Gewicht der Terme  $j$  und  $k$  in  $i$

$tf_{ijk}$ : Häufigkeit des gemeinsamen Vorkommens von  $j$  und  $k$  in  $i$

$df_{jk}$ : Anzahl der Dokumente in denen  $j$  und  $k$  gemeinsam auftreten

In unserem Fall ist  $tf_{ij}$  immer = 1, da jedes Wort in einer Subjectline nur ein Mal vorkommt. Es mag Ausnahmen geben, diese haben wir aber nicht beachtet.

Unsere Formel für das kombinierte Gewicht lautet:

$$d_{ij} = 1 \times \left( \frac{df_i}{N} \right) \quad (7)$$

Wir haben auf den Logarithmus verzichtet, da die Gewichte sonst zu klein würden. Die WeightingFactors aus (2) und (3) sind weggefallen, da für uns häufig vorkommende Wörter, wie z.B. Viagra, wichtig sind. Außerdem wurde der Kehrwert des Bruchs verwendet, da wir das prozentuale Vorkommen von Wörtern in der Dokumentenmenge erfahren wollten.

$$CW(T_j, T_k) = \frac{\sum_{i=1}^n d_{ijk}}{\sum_{i=1}^n d_{ij}} \quad (8)$$

$$CW(T_k, T_j) = \frac{\sum_{i=1}^n d_{ikj}}{\sum_{i=1}^n d_{ik}} \quad (9)$$

$$d_{ijk} = 1 \times \left( \frac{df_{jk}}{N} \right) \quad (10)$$

Um Einträge in den Thesaurus vorzunehmen, muss eine Mindestschranke überschritten werden:

$$CW \geq 0,01$$

### 3.6 Thesaurus

Insgesamt wurden 654 Wörter in den Thesaurus eingetragen; 428 von ihnen hatten keine Beziehungen. Werden in den Thesaurus auch Wortpaare aufgenommen, so erhöht sich die Anzahl auf 861 Einträge. Bei den Einträgen, zu denen Beziehungen errechnet wurden, handelte es sich meist um Medikamente. Ein Thesauruseintrag sieht wie folgt aus:

```
'shop'[['shop','shopping'], 10,
0.0049140049140049139,['onlin',
0.64000000000000001, 0.0039312039312039311]]
```

Zuerst wird der Schlüssel genannt; dazu benutzen wir das gestemte Wort. Danach folgen die nicht-gestemten Wortformen, die im Text gefunden wurden. Die Zahl 10 ( $df_i$ ) sagt aus, wie häufig ein Wort in der Dokumentenmenge vorgekommen ist. Es folgt das Gewicht des Schlüsselwortes  $d_{ij}$  und die Relationsliste mit dem Clustergewicht von  $CW(T_j, T_k)$  und dem kombinierten Gewicht  $d_{ijk}$ .

### 3.7 Auswertungen

Wir haben eine zeitliche Auswertung der Daten unternommen. Dabei blieben alle Programme und Formeln gleich, nur die Testmenge wurde in 3 Teile gespalten, die alle unabhängig voneinander untersucht wurden. Teil 1 deckte den Zeitraum vom 05. Januar bis Ende März ab. Dabei wurden 4.286 Spam-Mails gezählt von denen, nach Anwendung der Datenvorbereitungsprogramme, noch 2.035 übrig blieben. Teil 2 enthielt E-Mails aus dem Zeitraum von April bis Ende Mai. Hier wurden 4.778 Spam-Mails gezählt, nach Anwendung der Datenvorbereitungsprogramme waren noch 2.081 übrig. Abschließend zeigt Teil 3 den Zeitraum von Juni bis zum 23. September. Es wurden 4.234 Spam-Mails gezählt, nach Anwendung der Datenvorbereitungsprogramme betrug der Rest 2.075.

Der Apriori-Algorithmus lieferte folgende Ausgaben:

Tupellänge	Teil 1	Teil 2	Teil 3
1	251	289	253
2	121	128	105
3	144	72	126
4	163	72	210
5	127	56	252
6	62	28	210
7	17	8	120

Das Wort, welches am häufigsten gezählt wurde ist im ersten Teil der Daten *online* mit 90 Einträgen in Subjectlines. In Teilen 2 und 3 kommt das Wort *Viagra* am häufigsten vor, mit 83 respektive 117 Einträgen in Subjectlines vor. Es ist auffällig, dass im dritten Teil die meisten Spam-Mails verschickt wurden, die Medikamente be-

worben haben. Die folgende Tabelle fasst die Ergebnisse der drei separat erstellten Thesauri zusammen:

Einträge	Teil 1	Teil 2	Teil 3
Insgesamt	336	350	313
Davon Wortpaare	85	61	60
Ohne Relationen	188	198	180

Die Wörtern, die die längsten Relationslisten haben, sind Medikamentennamen.

Teil 1 der Daten wurde als Trainingsmenge für den Aufbau des inkrementellen Thesaurus benutzt, nach der oben beschriebenen Methode wurde ein Thesaurus erstellt, der im Weiteren mit den Daten aus Teil 2 und 3 aktualisiert wurde. Die Aktualisierung geschieht wie folgt:

- Bereite die Daten aus der Updatemenge vor (siehe Kapitel 4.1 bis 4.3).
- Wähle 'Kandidaten' für Schlüsselwörter für die Aktualisierung (auch Wortpaare und Tripel, das macht in unserer Implementierung der Apriori-Algorithmus).
- Die Menge der Schlüssel vom aktuellen Thesaurus wird mit der Menge der Schlüssel aus der Updatemenge vereinigt.
- Die Clustergewichte werden neu berechnet. Mit diesen Ergebnissen wird der neue Thesaurus erstellt.

In der folgenden Tabelle sind die Ergebnisse der Vereinigung der ersten zwei Datenmengen enthalten:

Tupellänge	Teil 1	Teil 2	Tupel Gesamt
1	251	289	377
2	121	128	211
3	144	72	185

In dem aktualisierten Thesaurus finden sich 491 Einträge, davon 114 Paare als Schlüssel, 259 Einträge haben keine Beziehungen. Als letztes wird der dritte Teil hinzugefügt. Die Vorgehensweise ist die Gleiche.

Tupellänge	Teil 1 & 2	Teil 3	Tupel Gesamt
1	377	251	377
2	211	121	211
3	185	144	185

Die Eintragszahl im aktualisierten Thesaurus beträgt 486, davon 109 Paare, 258 Schlüssel stehen ohne Beziehungen. Sie ändert sich, weil alle Gewichte für die neue Anzahl der Dokumente, also für das neue  $N$ , umgerechnet werden.

## 4 Ergebnisauswertung

In diesem Kapitel wollen wir die Güte unseres inkrementellen Thesaurus bewerten. Zunächst werden die Ergebnisse der separaten Thesauruserstellung verglichen:

- Gemeinsame Elemente in allen 3 Teilen: **124**
- Elemente die nur in Teil 1 vorkommen: **134** (von 336)
- Elemente die nur in Teil 2 vorkommen: **92** (von 350)
- Elemente die nur in Teil 3 vorkommen: **107** (von 313)

Aus der oberen Tabelle sieht man, dass sich die Anzahl der Schlüssel im inkrementellen Thesaurus nach dem Hinzufügen der Daten aus Teil 3 nicht geändert hat, das heißt nicht unbedingt, dass keine neuen Schlüsselwörter in den Thesaurus aufgenommen wurden, manche Wörter können nach der Berechnung der neuen relativen Gewichte an Bedeutung verloren haben. Vergleicht man den inkrementellen Thesaurus mit dem Anfangsthesaurus (Abschnitt 4.6) so ist der inkrementelle um fast 50% kleiner (iT: 486; T:861). Es kommen 484 gemeinsame Elemente vor. Zwei Wortpaare kommen nicht im Thesaurus vor: 'chemist quality' und 'drug online'. Das bedeutet, dass die Wörter, die in der ganzen Datenmenge versteckt waren und in jedem Teil weniger als 5 Mal vorgekommen sind, sind in den inkrementellen Thesaurus nicht aufgenommen worden, in dem ersten großen Thesaurus doch enthalten waren. Andererseits sind von den inkrementellen Thesaurus 2 Schlüssel erkannt worden, die nicht in dem Anfangsthesaurus vorkommen, das kommt davon, dass die Gewichte partiell berechnet werden. Ein großer Vorteil des inkrementellen Thesaurus ist, dass die Datenmengen, die bei der Aktualisierung durchsucht werden, nicht größer werden, es werden nur die Gewichte neu berechnet, und die arithmetischen Operationen sind viel 'billiger' als Suchen, Sortieren und Mustererkennung (im Sinne von regulären Ausdrücken). Es sind zu viele relationslose Schlüssel im Thesaurus enthalten. Solche Einträge haben keine Information über Beziehungen zwischen den Wörtern und sind nichts anderes, als eine Liste, die aber mit einem großen Aufwand erstellt wurde. Diese Liste kann man aber nicht als Blacklist bezeichnen, weil sie solche Wörter enthält wie 'go', 'man', 'sleep' und Ähnliches. Ein Vergleich mit der Spam-Blacklist, die wir bei der Sichtung der gesamten Testmenge erstellt haben, zeigt, dass 58 Wörter nicht im Thesaurus und 69 nicht im inkrementellen Thesaurus vorkommen. Die Blackliste hat 103 Einträge. Das bedeutet, dass einem Menschen schon beim kurzen Anschauen Dinge auffallen, die man mit einem Algorithmus nicht entdecken kann (schon ein kurzer Blick auf eine E-Mail reicht, um zu wissen, ob es Spam ist, ohne den Text zu lesen).

## 5 Fazit

Im Verlauf des Praktikums wurde immer klarer, wieso bei unserer vorabrecherche Thesauri als Spam-Filter nicht

gewünscht waren. Das Problem liegt nicht allein in den vielen Domänen des Spams. Es ist der Mangel an Flexibilität, der Thesauri als ungeeignet ausweist. Die Spamflut wächst mit jedem Tag und genauso schnell verändert sie sich. Da ein Thesaurus die Wörter einer Spam-Mail untersucht und anhand dieser Regeln für die Aussortierung von E-Mails generiert, werden für alle Spam-Typen Wörter in den Thesaurus gespeichert. Durch die Tatsache, dass immer mehr Spams keine verdächtigen Inhalte haben, werden irgendwann auch 'üblich benutzte' Wörter Teil des Spam-Thesaurus. Weiterhin erkennt der Thesaurus nicht alle Wörter, die ein Mensch sofort als Spamverdächtig ausweisen würde. Das Problem hier ist die Frequenz der jeweiligen Wörter bzw. Wortpaare. Lässt man zu, dass auch einmalig genannte Begriffe in den Thesaurus aufgenommen werden, läuft dieser schnell mit 'alltäglichen' Wörtern bzw. Phrasen voll. Was ist aber mit Begriffen wie 'child porn'. Schon beim einmaligen Auftreten ist für den Mensch erkennbar, dass diese E-Mail nicht nur unerwünscht, sondern auch illegal ist. In dieser Situation ist ein Blacklist-Filter effektiver als jeder Thesaurus. Abschließend kann man sagen, dass ein Thesaurus als Spamfilter zwar möglich, in der Praxis aber nicht anwendbar ist.

## A Acknowledgement

Die vorliegende Arbeit fand im Rahmen des Praktikums *Text Mining and Retrieval* im Sommersemester 2006 an der JW Goethe-Universität Frankfurt am Main unter der Leitung von Prof. Dr. Christoph Schommer<sup>1</sup> statt.

## B References

- [1] Apriori Algorithmus. <http://www-agrw.informatik.uni-kl.de/damit/a/apriori.html> (Stand 2006).
- [2] GermaNet. <http://www.sfs.uni-tuebingen.de/lzd/> (Stand 2006).
- [3] Hsinchun Chen, Bruce Schatz, Joanne Martinez, Tobun Dorbin Ng.: Generating a Domain-specific Thesaurus Automatically: An Experiment on FlyBase. <http://ai.bpa.arizona.edu/papers/ijmms94/ijmms94.html>.
- [4] Thomas Karow und Ruth Lang-Roth. Allgemeine und spezielle Pharmakologie und Toxikologie 2005.
- [5] OpenThesaurus - Deutscher Thesaurus. <http://www.openthesaurus.de/> (Stand 2006).

<sup>1</sup>University of Luxembourg, Campus Kirchberg, Dept. of Computer Science and Communication, 6 Rue Coudenhove-Kalergi; 1359 Luxembourg, Luxembourg, Email: christoph.schommer@uni.lu

- [6] Porter's Stemmer.  
<http://www.tartarus.org/martin/PorterStemmer>  
(Stand 2006).
- [7] Pschyrembel - Klinisches Wörterbuch - Juni 2004.  
Walter de Gruyter Verlag, Berlin.
- [8] Andreas Ruß und Stefan Enders: Arzneimittel  
pocket plus 2005, Auflage 1. Bröm Bruckmeier Ver-  
lag (Stand 2006).
- [9] Wikipedia - Die freie Enzyklopedie (deutsch).  
<http://de.wikipedia.org/wiki/Hauptseite> (Stand  
2006).
- [10] WordNet - a lexical database for the English language.  
<http://wordnet.princeton.edu/> (Stand 2006).