Newsfiltering am Beispiel des IntFilter-Projekts

Nils Malzahn

nils@fsinfo.cs.uni-dortmund.de

11. März 1999

Benutzte Literatur

Literatur

[1] Intelligent Information Filtering, The IntFilter Project, Frederik Kilander, Eva Fåhræus, Jacob Palme, Dpt of Computer Sciences, Stockholm University, 17. Feb 1997 http://www.dsv.su.se/˜fk/ifDoc/IntFilter.html

[2] PEFNA The Private Filtering News Agent, The IntFilter Project, Fredrik Kilander, Eva Fåhræus, Jacob Palme, Dpt of Computer Science, Stockholm University , 17. Feb 1997

[3] A Brief Comparison of News Filtering Software, Fredrik Kilander, 19. Jun 1996, http://www.dsv.su.se/˜fk

[4] Suffering from Infoload, Les Logiciels Machina Sapiens Inc. http://www.machinasapiens.qc.ca/english/products/infoscan/infoscanang.html, 7. Jan. 1999

[5] NewsClip(TM) News Filtering Language, ClariNet, http://www.clarinet.com/newsclip.html, 7. Jan 1999

[6] An Architecture for Internet Information Filtering, Jim Binkley, Leslie Young, Compuer Science Department, Portland State University, Portland, 1. Mai 1991

Istzustand vs. Sollzustand

Ausgangspunkt

Zeige alle ungelesenen Nachrichten aus den gewählten Newsgroups nach *Ankunftszeit* geordnet.

Ziel

Zeige alle *für mich interessanten* ungelesenen Nachrichten aus den gewählten Newsgroups nach *Wichtigkeit* geordnet.

Anforderungen an ein Newsfiltering-System

Es soll:

- klein aber *leistungsfähig* sein.
- *einfach* zu konfigurieren sein.
- Nachrichten sortieren - *nicht vorenthalten!*.
- dem Benutzer *klare, wahrnehmbare* Hinweise bzgl. der Wichtigkeit einer Nachricht geben.

Probleme

Das Auffinden *eines* Artikels lösen Methoden aus dem Information Retrieval. Was ist mit der zeitlichen Abhängigkeit eines Diskussionsstrangs?

Es können nicht alle Newsartikel vorgehalten werden, da das zuviel Speicherplatz benötigt.

maschinelle Lösungsversuche

Lösungsansatz	Nachteile
Killfiles (binäre Klassifizierung)	Es können wichtige Informationen verloren gehen.
Keywords	Fehlklassifizierung wahrscheinlich Ergebnisse nur brauchbar, wenn eine kleine Menge von Newsgroups betrachtet wird
Rating (diskrete Skala)	Hinter den Punkten der Ratingskala verbergen sich keine Bedeutungen Es muss gute und schlechte Beispiele geben Eindimensionale Skalen nicht ausreichend, da uninteressante Themen mit interessanten Autoren kollidieren können
Kategorisieren (kont. Skala)	eine Verbesserung des Ratings, aber mit prinzipiell denselben Nachteilen
Ökonomisches Filtern	Da der Artikel nach dem Ressourcenverbrauch beurteilt wird, wird keine Aussage über den Inhalt getroffen.

Lösungsversuche mit starker Einbindung des Menschen

Kognitives Filtern läßt den Mensch ein paar Zeilen eines Artikels lesen. Daraufhin entscheidet die Person, ob sie weiterliest.
Nachteil: Der Mensch macht doch die gesamte Arbeit.

Soziales Filtern beruht auf der Annahme, dass Menschen, die in der Vergangenheit dieselben Interessen hatten, sich auch in Zukunft ähnlich verhalten werden.
Nachteil: Die ersten Leser eines Artikels hat keinen Nutzen vom System. Der Filter leistet nur für gemeinschaftskonforme Menschen gute Dienste.

Erfahrungen

- Es ist schwierig den Computer lesenswerte/nicht-lesenswerte Artikel unterscheiden zu lassen.
- Der normale Benutzer scheint i.d. Lage zu sein diese Entscheidung intuitiv zu treffen.

$\hookrightarrow$ Lassen wir den Computer das machen was er gut kann:
  - Informationen darstellen
  - entscheidungsrelevante Daten/Informationen hervorheben.

Probleme:

- Eine grafische Visualisierung ist nicht für jedeN BenutzerIn gleich gut.
- Quotes erschweren das Hervorheben von relevanten *neuen* Informationen.
- Ein zu guter Filter wendet sich gegen sich selbst, weil er nur noch positive Beispiele lernt und dem Leser potentiell interessante Artikel vorenthält (Interessensverschiebung)

Auswahlkriterien

Prozent	Kriterium
80%	Subject (Was ich *dachte*,dass der Artikel beinhaltet
40%	Das Genre, zu dem der Artikel gehört
30%	Position im Artikelthread
25%	die präsentierte Anordnung

positive Auswahl - Quelle: [1]

Prozent	Kriterium
80%	Subject
35%	Länge des Artikels
30%	Quotemenge
25%	Autor des Artikels
25%	Struktur des Artikels
25%	Genre des Artikels

negative Auswahl - Quelle: [1]

Benutzerführung

- Das User Interface ist aus Ergonomie- und Motivationsgründen einfach zu halten.
- Diskrete Rankingsysteme sehr beliebt. Nur 1 von 5 Knöpfen drücken. Fertig!
- Zeitbasierte Rankings zu Unscharf (Kaffeeholen zwischen zwei uninteressanten Artikeln.)

Das System GHOST

GHOST benutzt ein forward-chaining Produktionensystem mit dubiosen Eigenschaften.

- Die Konfliktlösungsstrategie zwischen zwei Regeln ist nicht-deterministisch. Es beruht auf einer Zufallsentscheidung.

- Die Regeln können sowohl kategorisierend als auch vernichtend sein.

$\hookrightarrow$ Es ist schwierig sicherzustellen, dass keine Nachricht aus Versehen gelöscht wird.

$\hookrightarrow$ Die Arbeit an GHOST wurde eingestellt.

Das System Scripts

Ansammlung von Unix-Shellskripten und standard Unix-Utilities (sed, grep, wc, comm, awk, sort) um die Nachrichten von einem Newsserver zu holen und anschließend mit einer Anzahl von Wörtern aus einer Menge von Beispieltexten zu vergleichen.

Nur Wörter mit mehr als 5 Buchstaben und die Anzahl der Originaltextzeilen werden im Ergebnis berücksichtigt.

Die Ausgabe besteht aus einer HTML-Seite, die die Artikelreferenzen nach absteigender Gewichtung auflistet.

Ergebnisse

- Das System funktioniert
- Das 5 Buchstaben-Limit war zu eng gefaßt, da Akronyme nicht erfaßt werden.
- Es ist einfach zu langsam
- **ABER:** Besser ein nicht völlig kompetenter Filter als gar keiner!

Arbeitsplan des Klassifizierungsagenten

1. Der Agent öffnet das .newsrc-file und holt alle ungelesenen Artikel vom NNTP-Server.

2. Jeder ungelesene und noch nicht bewertete Artikel wird den Kategorien zugeordnet; das Ergebnis wird in die Artikeldatenbank eingetragen.

3. Wenn eine Kategorie verändert wird (hinzufügen/löschen von Bsp.-Artikeln), werden alle ungelesenen Artikel erneut geholt und bewertet.

4. Klassifizierte und gelesene Artikel werden nach einer best. Zeit aus der Artikeldatenbank[1] gelöscht.

[1] Die Datenbank enthält nur die Message-ID der Nachricht, nicht den Text.

Arbeitsweise des Klassifizierungsagenten

1. Die guten Beipiele werden den Kategorien zugeordnet.
2. Es werden Wortlisten bestehend aus allen Wörtern aller Artikel der Kategorie erstellt.
3. Ein Artikel wird ebenfalls als Wortliste dargestellt.
4. Das Cosinus-Maß wird zur Ähnlichkeitsbestimmung zwischen Artikel und Kategorie erstellt.
5. Bei Threads wird der *Thread* einer Kategorie zugeordnet. Die Kategorie, die den meisten Artikeln zugeordnet ist, gewinnt.

Effizienz

- Aus Testgründen wurden die Wortlisten für jeden Beispielartikel behalten.
- Die Wortlisten werden als Hashtable abgelegt und das Wort selbst nur einmal gespeichert.
- Praktikables Limit: 50-90 Beispielartikel insgesamt. (9 Kategorien, 6 NG $\rightsquigarrow$ 15min Laufzeit)

Wortklassen

Wort-klasse	Inhalt	Beispiel
1	Sehr häufig vorkommende Wörter, die nur helfen die Sprache/Grammatik herauszufinden. Sie tragen nichts zur Klassifikation bei.	ist, is, est-ce, dem, den...
2	Nicht direkt zum Thema gehörende Wörter, die den Stil des Autors bestimmen.	"Mary didn't want to ride any more" "Mary was disinclined to further horseback exercise"
3	Themenspezifische Wörter	HTML, KI, WWW, DMA

Die Grenzen zwischen den einzelnen Klassen sind fließend.

Stoplisten - und alles ist wunderbar?

- Wörter der Klasse 2 sind problematisch. Die Anzahl dieser Wörter nimmt zu, je mehr Artikel und Unterthemen hinzukommen.

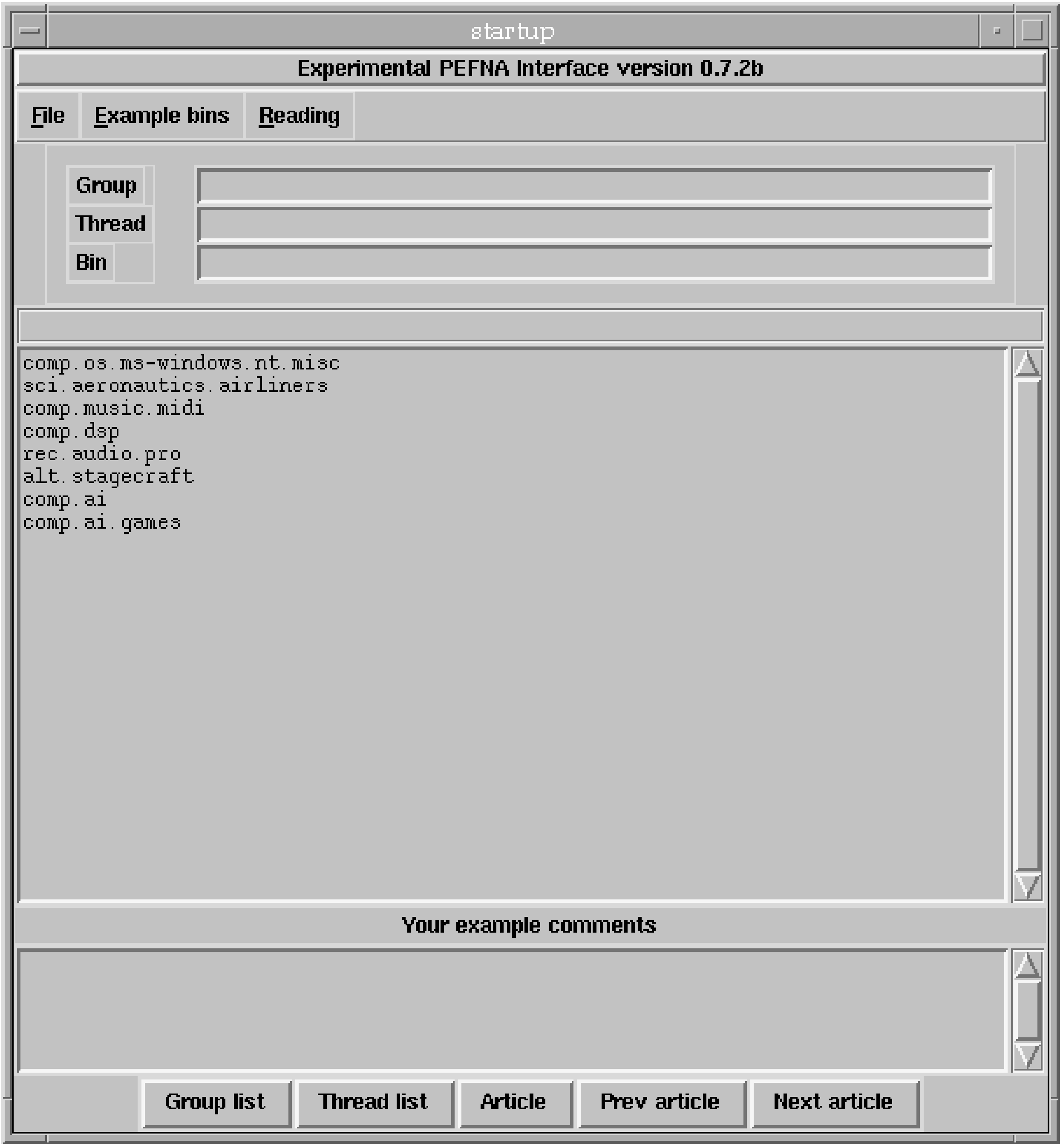
- Wörter der Klasse 3 kommen nur selten bis gar nicht in den kurzen Newsartikeln vor. Sie sind unterrepräsentiert.

↪ Wörter der Klasse 2 dominieren das Maß, wenn viele Beispielartikel für eine Kategorie vorhanden sind.
  - IDF hilft nicht bei diesem Problem, sondern verschlimmert es nur.
  - Wahrscheinlichkeitsbetrachtungen sind genauso gut wie einfaches Zählen des Wortvorkommens.
  - Informationsgehaltsmaß liefert gute Ergebnisse.

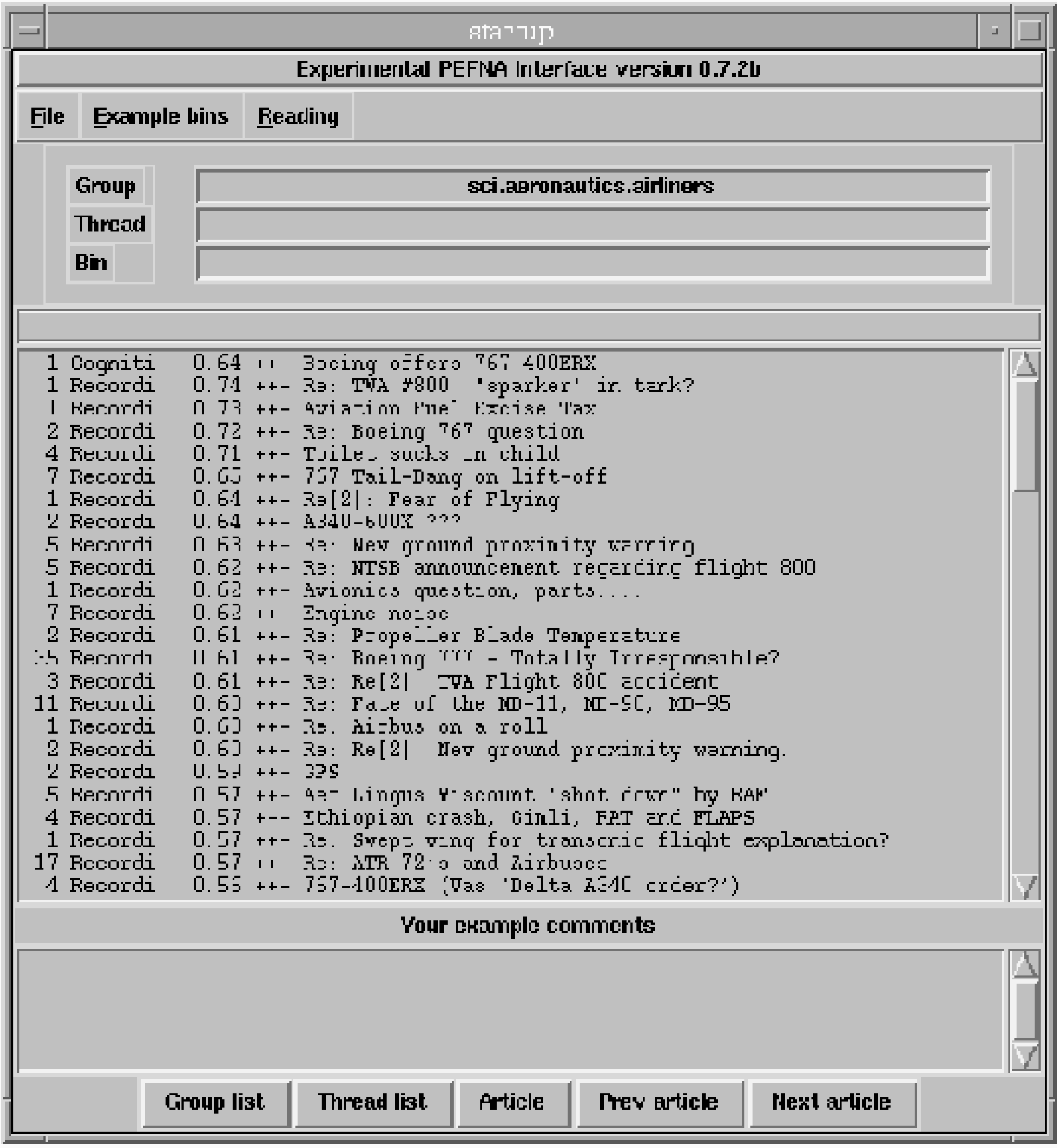
Repräsentation des Ergebnisses

- Die Artikel werden in der vom Benutzer bestimmten Reihenfolge der Kategorien angezeigt.
- In der zweiten Spalte wird die Ähnlichkeit des Artikels zur Kategorie angezeigt.
- In einer dritten Spalte wird der Abstand der Gewinnerkategorie zu allen anderen Kategorien angezeigt.
- Der Benutzer kann sich die Abstände der Artikel als Graph anzeigen lassen.

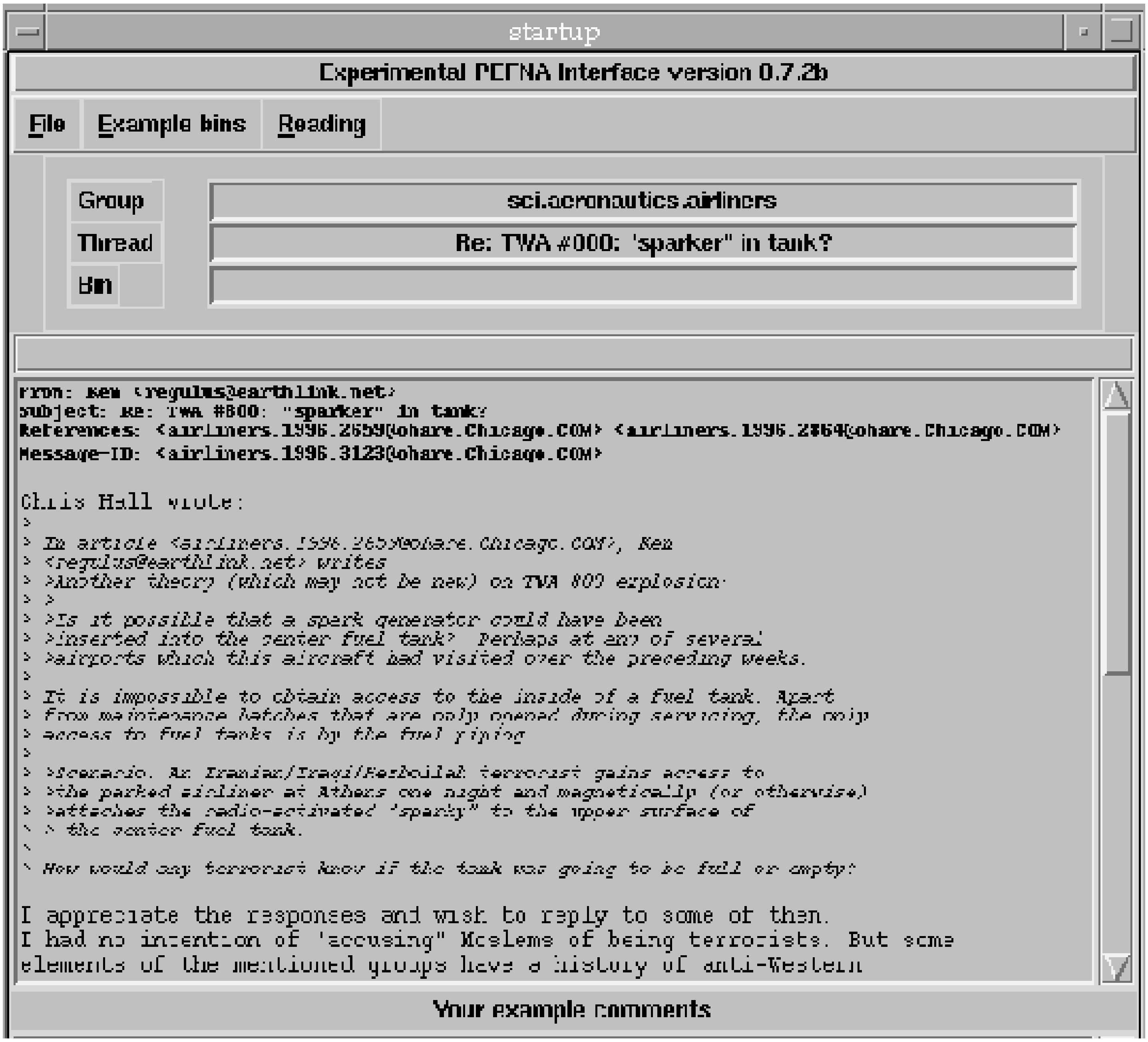
Die Newsgruppenauswahl



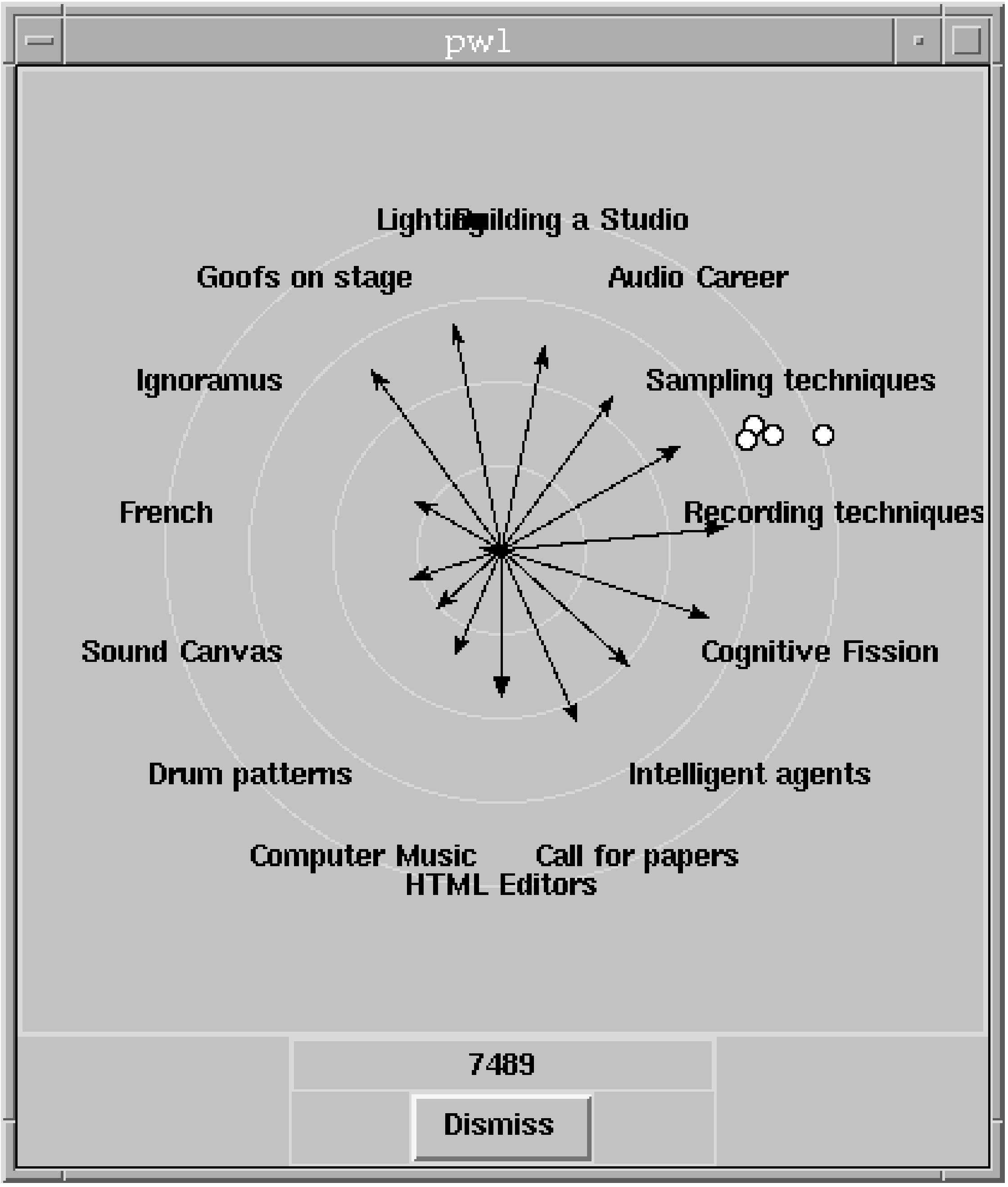
Die Thread-Auswahl



Die Artikeldarstellung

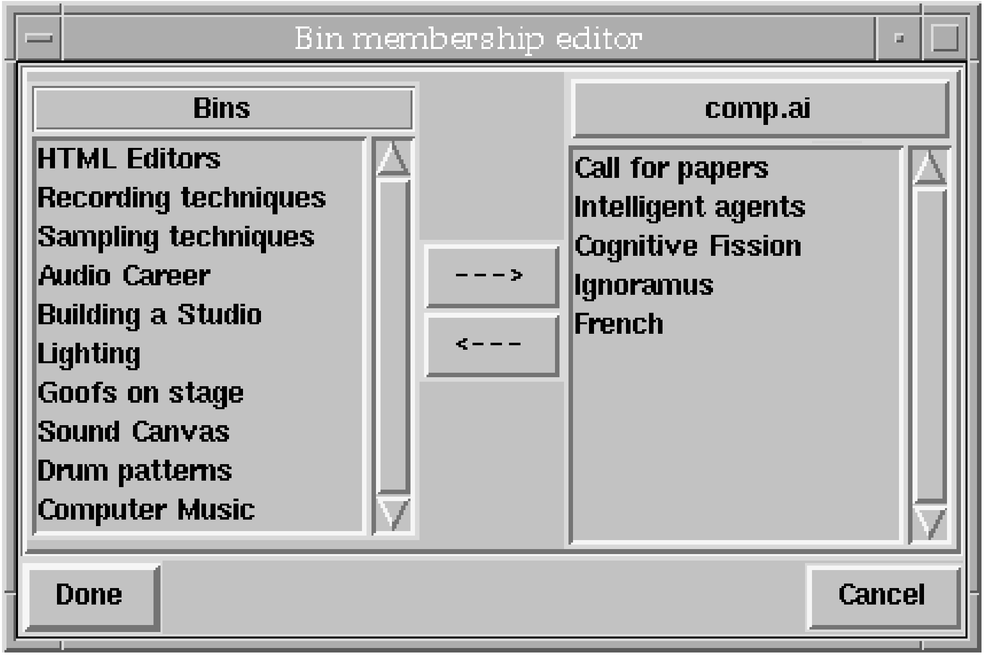


Die Kategorie-Distanz-Visualisierung



Die Folge der Kategorien

Auswahl zulässiger Kategorien



Vergleichsmerkmale

1. Modell - Werden *ausgewählte* News zugänglich gemacht (Retrieval) oder werden alle Artikel *bewertet* (Rating)?
2. Adaptivität - Wie wird der Filter an die individuellen Bedürfnisse des Nutzers angepasst?
3. Klassifikation - Wie klassifiziert der Filter?
4. Motivation - Wie begründet/verteidigt der Filter seine Klassifikation gegenüber dem Benutzer?
5. Operation - Wie und wann arbeitet der Filter?
6. Implementation - Wie ist der Filter implementiert?
7. Ergebnisse

Sift

1. Retrievalsystem
2. WWW-Benutzerprofile
3. - boolean Matching - alle Schlüsselwörter müssen vorhanden sein
   - weighed Matching - Artikel müssen eine Bewertungsschranke überschreiten
4. -
5. Remote Site, die in Intervallen Artikel per Mail verschickt
6. Die Implementation erfolgte in C
7. Es wurde nur die Geschwindigkeit betrachtet

RAMA

1. Retrievalsystem, dass auch FTP-Server in die Suche einschließt
2. Der Benutzer schickt per Mail Anfragen an den RAMA-Server.
3. Cosinus-Maß + Patternmatchig (agrep) + Boolean Matching
4. Mit der e-mail wird die passende Anfrage und das Gewicht des Artikels verschickt
5. Das Ergebnis wird per e-mail an den Benutzer verschickt.
6. Das System arbeitet nachts die Anfragen des Tages ab und verschickt Artikel per e-mail
7. Nur Filtergeschwindigkeit getestet.

Lurker

1. Retrievalsystem
2. Der Benutzer schreibt eine Menge von Regeln für jede NG. Es handelt sich um boolsche Ausdrücke
3. Anwendung der gegebenen Regeln und Abholen aller passender Artikel
4. -
5. Der Filter läuft, wann immer der Benutzer ihn ausführt.
6. Lurker ist in Perl geschrieben
7. -

NewsClip

1. Binäres Ratingsystem
2. Der Benutzer programmiert den Filter in NewsClip-Sprache
3. Wendet die geschriebenen Regeln nacheinander an
4. -
5. Der Filter arbeitet wann immer er aufgerufen wird und ändert die .newsrc
6. NewsClip übersetzt die Filtersprache in C.
7. -

http://www.clarinet.com/newsclip.html

Browse

1. Ratingsystem
2. Gelesene und ungelesene Artikel werden als vom Benutzer bewertet betrachtet
3. Verschiedene NGs werden unterschiedlich behandelt, aber das System beachtet keine Threads
4. Neurales Netz stellt Beziehungen zwischen Wörtern her
5. -
6. Implementation in C und X-Window als GUI
7. Die Resultate befassen sich mit der Präzision und Erinnerung des neuralen Netzes

Grouplens

1. Ratingsystem
2. Der Benutzer kann die Artikel bewerten, während er sie liest.
3. Kolloboratives Filtern
4. Das System zeigt dem Benutzer die vorhergesagte Wertung des Artikels
5. -
6. Es wurden verschiedene Newsreader ausprobiert und modifiziert, so dass Präsentation und Wertung der Artikel möglich wird
7. Das GUI wurde immer weiter verbessert

SMART

1. Retrieval System
2. Der Benutzer wird wiederholt befragt
3. Cosinus-Maß
4. Keine spezielle Begründung ausgegeben, da der Benutzer ständig verfeinerte Anfragen stellt
5. Es handelt sich eher um ein DBMS
6. Die Implementierung erfolgte in C/C++
7. Statt der Gelehrigkeit wurde das Retrivalsystem getestet

Borges

1. Retrievalsystem
2. Der Benutzer schickt Queries an den Borges-WWW-Server
3. ?
4. ?
5. Täglich wird eine WWW-Seite erstellt, die die Suchergebnisse enthält.
6. Der Filter läuft einem WWW-Server, so dass jeder WWW-Browser geeignet ist
7. ?

http://borges.compapp.dcu.ie

InfoScan

1. Retrievalsystem
2. Schlüsselwörter können vom Benutzer gewichtet werden und die Anfrage kann interaktiv verbessert werden..
3. Der Benutzer definiert Schlüsselwörter und Schlüsselwortsammlungen, die zu NGs gehören.
4. Ein Ähnlichkeitsradar zeigt die Ähnlichkeit eines Artikels.
5. Der Filter arbeitet in regelmässigen Intervallen und besteht aus mehreren Subfiltern.
6. ?
7. -

http://www.machineasapiens.qc.ca/english/infoscanang.html

Newsweeder

1. Kombination aus Retrieval- und Rating-System
2. Der Benutzer ordnet gelesene Artikel 1 von 5 Kategorien zu
3.
   - Cosinus-Maß
   - MDL
4. Wenn kolloborativ gefiltert wird, dann wird die durchschn. Wertung anderer angezeigt.
5. Training des Benutzerprofils erfolgt nachts.
6. (gehackter?) WWW-Browser
7. 44% bis 59% Vorhersagegenauigkeit


Fichier Édition Lecture Outils Profil Aide 11:56:09
InfoScan
REQUÊTES
Alliances strat.
COLLECTIONS
Entreprises
FILTRES
Technologie
IBM achéte Lotus !
produits
finances
ventes
bourse
stratégie
35/237

Filtermodelle, Filteradaptierung

Retrieval	Rating
SIFT	Newsweeder
Browse	
SMART	Infoscan
Lurker	GroupLens
NewsClip	PEFNA
RAMA	
BORGES	

Tabelle 1: [3] Table 1

Programmieren	Anfrage	Training	Kolloborativ
NewsClip	SIFT	NewsWeeder	
Lurker	SMART	PEFNA	GroupLens
	Browse		
	RAMA		
	Borges		
	InfoScan		

Tabelle 2: [3] Table 2

Klassifikation der Newsfiltering-Programme

boolesche Ausdrücke	Cosinus Maß	MDL	neurale Netze	Benutzerprofil Korrelation
NewsClip Sift Lurker	Newsweeder Sift SMART RAMA InfoScan Borges PEFNA	Newsweeder	Browse	GroupLens

Tabelle 3: [3] Table 3

Filterarbeitsweisen

<table>
<tr><th>Interaktiv</th><th>Batch</th></tr>
<tr><td>Smart</td><td>Lurker</td></tr>
<tr><td>InfoScan</td><td>SIFT</td></tr>
<tr><td colspan="2" align="center">NewsWeeder</td></tr>
<tr><td colspan="2" align="center">Browse</td></tr>
<tr><td colspan="2" align="center">BORGES</td></tr>
<tr><td colspan="2" align="center">GroupLens</td></tr>
<tr><td colspan="2" align="center">PEFNA</td></tr>
<tr><td></td><td>NewsClip</td></tr>
<tr><td></td><td>RAMA</td></tr>
</table>

Tabelle 4: [3] Table 4

Fazit

Die Begründung für die Ergebnisse, die die verschiedenen Newsfiltersysteme liefern, lassen zu wünschen übrig. Es gibt drei Hauptunterschiede:

1. News als
   - Nachrichtenpool, aus dem man interessante Nachrichten *extrahieren* kann
   - Strom neuer Nachrichten, von denen einige interessanter sind als andere.
2. Der Filter ist
   - ein komplexes Produkt aus viel Erfahrung aus maschinellem Lernen oder Information Retrieval
   - ein simples Patternmatching System
3. Der Filter arbeitet
   - mit einem speziell angepassten Userinterface
   - still und leise im Hintergrund

Warum überhaupt filtern

- Die Datenmenge ist zu groß.
- Viele Leute verzichten daher sogar gänzlich auf News.
- Für den Benutzer muß die Datenflut vorverarbeitet (gefiltert) werden.

Was ein Visualisierungstool können sollte

- Die Benutzerin sollte Schlüsselwörter angeben können
  - bevorzugte Wörter werden dann z.B. grün gefärbt
  - abgelehnte Wörter z.B. rot
- Die Benutzerin sollte Minimal- und Maximalgrenzen für die Länge eines Textes angeben können.
- Die Benutzerin sollte eine Maximalgrenze Quote-Anteil angeben können.
- Die Texte sollten grafisch repräsentiert werden, so dass der Grafik entnommen werden kann, ob sie bevorzugte/abgelehnte Wörter beinhalten.
- Wenn ein Text angewählt wird, dann sollte die Nähe zur gewählten Kategorie erscheinen.
- Die üblichen Eigenschaften eines guten Newsclients sollten verwirklicht werden.

Analyse des Cosinus-Ähnlichkeitsmaß

Zur Erinnerung:

$$\text{Ähnlichkeit}_{\text{Artikel,Kategorie}} = \frac{\overrightarrow{\text{Gewicht}_{\text{Artikel,Wort}}} \cdot \overrightarrow{\text{Gewicht}_{\text{Kategorie,Wort}}}}{|\overrightarrow{\text{Gewicht}_{\text{Artikel,Wort}}}| \cdot |\overrightarrow{\text{Gewicht}_{\text{Kategorie,Wort}}}|}$$

Wörter mit gleicher Verteilung beeinflussen die Ähnlichkeit am meisten (Wörter der Klasse 1)

$\hookrightarrow$ Einführung von Stoplisten.

Um wirklich gute Ergebnisse zu erzielen sind längere Texte erforderlich.

† Newsartikel sind i.d.R zu kurz.

Term Gewichtung

Drei Arten der Kategorisierung getestet:

1. **Term Frequency**. Je öfter es in der Kategorie vorkommt, desto wichtiger ist es.
2. **Inverse Document Frequency**. Die Termhäufigkeit wird durch die Anzahl der Kategorien, in denen es vorkommt, geteilt.
3. **Discriminative informativity**. Bewertet Wörter, die nur selten in anderen Kategorien vorkommen hoch u.u.

Das Verfahren 3 brachte die besten Ergebnisse. (Immerhin 7,6mal so gut wie eine zufällige Verteilung der Artikel auf die Kategorien.)

Das Informationsgehaltsmaß

Das Informationgehaltssmaß [discriminative power] ist besser als das einfache Zählen.

$$I(t) = -\sum_{i=1}^{N} P(t|C_i) \log_2 P(t|C_i)$$

Gewichtung durch:

$$w_t = \max(I(\cdot)) - I(t) \quad \text{oder } w_t = \frac{\max(I(\cdot)) - I(t)}{\max(I(\cdot))}$$

$\max I(\cdot)$ ist das informationsloseste Wort.

$\hookrightarrow$ Informationsgewichtung ist besser für kleinere Mengen von Beispielen.
Eine kleine Menge von präzisen Beispielen ist besser als eine große Menge von diffusen Beispielen.

Geschichte des Intfilter Projekts

1. Am Anfang stand das System GHOST
2. Im August '94 wurde GHOST nochmal beurteilt
3. Es entstand Scripts
4. Es wurden Benutzerverhaltensstudien durchgeführt
5. Es wurden Algorithmenstudien durgeführt:
   - Substring-Indexing[1]
   - MDL (als zu schwierig verworfen)
   - Cosinus-Maß(für gut befunden)

$$\frac{2|V_q \cap V_d|}{|V_q| + |V_d|}$$

$$\cos(q,d) = \frac{\vec{w}_d \cdot \vec{w}_q}{|\vec{w}_d| \cdot |\vec{w}_q|}$$

[1] Ermittlung des Verhältnisses zwischen gemeinsamen Wörtern zweier Dokumente und der Gesamtlänge der Dokumente

6. Es wurde überlegt wo der Filter arbeitet:
   - ProxyNewsserver (wie GHOST)
   - modifizierter Newsreader
   - Newsserver
7. Entschluß zwei Komponenten zu bauen
   - Tcl/Tk Newsreader mit fünf Knöpfen zur Bewertung der Artikel
   - stand-alone Filterprogramm in C, dass das Cosinus-Maß verwendet.

   Die Kommunikation erfolgt über eine Datenbank, in der der Filter die Informationen über die bewerteten Artikel ablegt.

   Nachteil der Trennung: der Grund für eine Bewertung durch den Filter kann der Newsclient nicht wiedergeben.

   Vorteil der Trennung: Es kann einfach ein neuer Filter eingesetzt werden ohne den Client zu beeinflussen. Es ist sogar möglich mehrere Filterprogramme auf der gleichen Datenbasis laufen zu lassen.

8. Im September '95 stößt Eva Fåhræus zur IntFilter-Gruppe und veranlaßt einen Umschwung auf Kategorien statt diskrete Ratings.
   Der Filter unterscheidet die verschiedenen Kategorien nicht mehr nach Gewichtung, sondern schiebt die Artikel einfach in die passendste. Die 5 Knöpfe wurden daher durch ein Pulldown-Menü ersetzt. Der Benutzende ordnet nun einen besonders beispielhaften Artikel einer Kategorie zu.
9. Die Hinzunahme des Informations-Entropie-Maß hat die Kategorisierung verbessert.
10. Eine weitere Verbesserung brachte das einführen von Kategorien für jede einzelne Newsgroup.

Fazit

Das PEFNA-System tut, was es soll, aber im Laufe der Arbeit hat sich herausgestellt, dass das Problem schwieriger ist, als von einer Schlüsselwortsuche erwartet wurde, da die Struktur nat. Sprache schwierig zu erfassen ist..

Die geforderte Visualisierung der Nachrichten und der Begründung für die Kategorisierung sollte noch fortgesetzt werden.