Winnow: Software to replace journal table of contents alerts

George Chambers
and David Sterratt

Institute for
Adaptive & Neural Computation

School of informatics

The University of Edinburgh

The problems with email tables of contents alerts

- Small fraction of articles relevant
- Hence alerts go unread
- To read abstracts and articles, researcher has to deal with a variety of journal-specific interfaces
- Setting up tables of contents alerts requires dealing with multiple websites too

Winnow: a solution

- Collects the latest journal tables of contents from a database (Pubmed)
- Classifies the articles as interesting or boring using Naïve Bayes
- Learns from the user which articles are interesting or boring
- Displays abstracts; easy access to PDFs
- Can save references in BibTeX

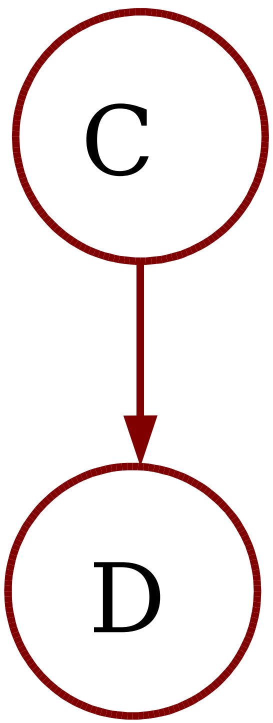
Implementation

- Winnow is a Java application
- It interacts with PubMed, hosted by the National Center for Biotechnology Information (NCBI)
- Search and retrieval of data is via the E-Utilities, a set of HTTP tools
- Results are returned in XML format

Data Sources

- Considerations:
  - Protocol (e.g. Z39.50)
  - Authentication
  - Availability of full text and abstracts
- Z39.50 gives access to:
  - ZETOC (comprehensive coverage; titles only; no authentication)
  - BIOSIS
  - Perhaps Web of Knowledge in the future

Naïve Bayes Algorithm



$$\text{Class C can be } c_1 \text{ (good) } c_2 \text{ (bad)}$$

$$\text{Document } d \text{ comprises } N_t \text{ instances of word } t\,(w_t) \text{ out of a vocabulary of } V$$

$$\text{Likelihood of generating document } d \text{ given } C=c_j\text{:}$$

$$P(D=d|C=c_j) \propto \prod_{t=1}^{V} \frac{P(w_t|c_j)^{N_t}}{N_t!}$$

$$(\text{Multinomial formula; independence; bag of words})$$

$$\text{Bayes: } P(C=c_j|D=d) = \frac{P(D=d|C=c_j)\,P(C=c_j)}{\sum_{k=1}^{2} P(D=d|C=c_k)\,P(C=c_k)}$$

$$\text{Posterior ratio } L = \frac{P(C=c_1|D=d)}{P(C=c_2|D=d)} = \frac{P(C=c_1)}{P(C=c_2)} \prod_{t=1}^{V} \left(\frac{P(w_t|c_1)}{P(w_t|c_2)}\right)^{N_t}$$

$$P(C=c_1|D=d) = \frac{1}{1+1/L}$$

Training Naïve Bayes

$$P(C=c_j)=\frac{1+m_j}{2+m_1+m_2}\propto 1+m_j$$

where m_j is no. articles seen of each class

$$P(w_t|C=c_j)=\frac{1+n_{tj}}{\sum_{t=1}^{V}(1+n_{tj})}$$

where n_{tj} is no. occurences of word t in class j counted over all documents, d_i i.e.

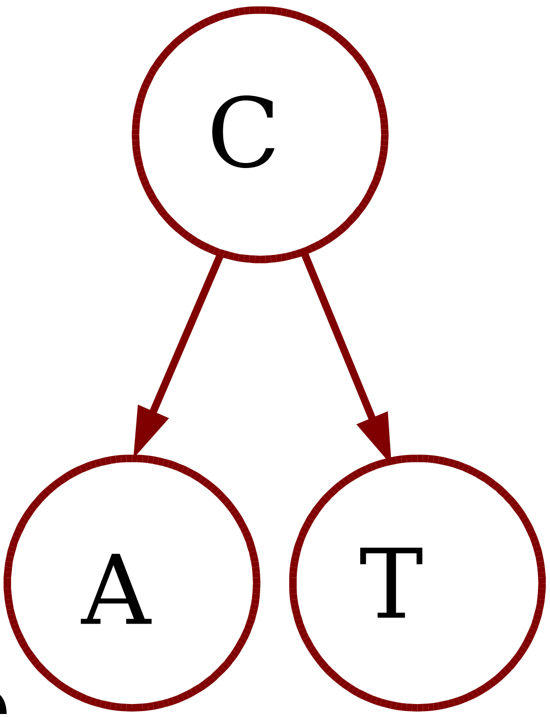
$$n_{tj}=\sum_i N_{it}P(c_j|D=d_i)$$

where $P(C=c_j|D=d_i)$ is the user's rating (0 or 1)

Stored in “good” and bad hashtables of words and counts.
Increment count for word in appropriate hash when training

Different Fields

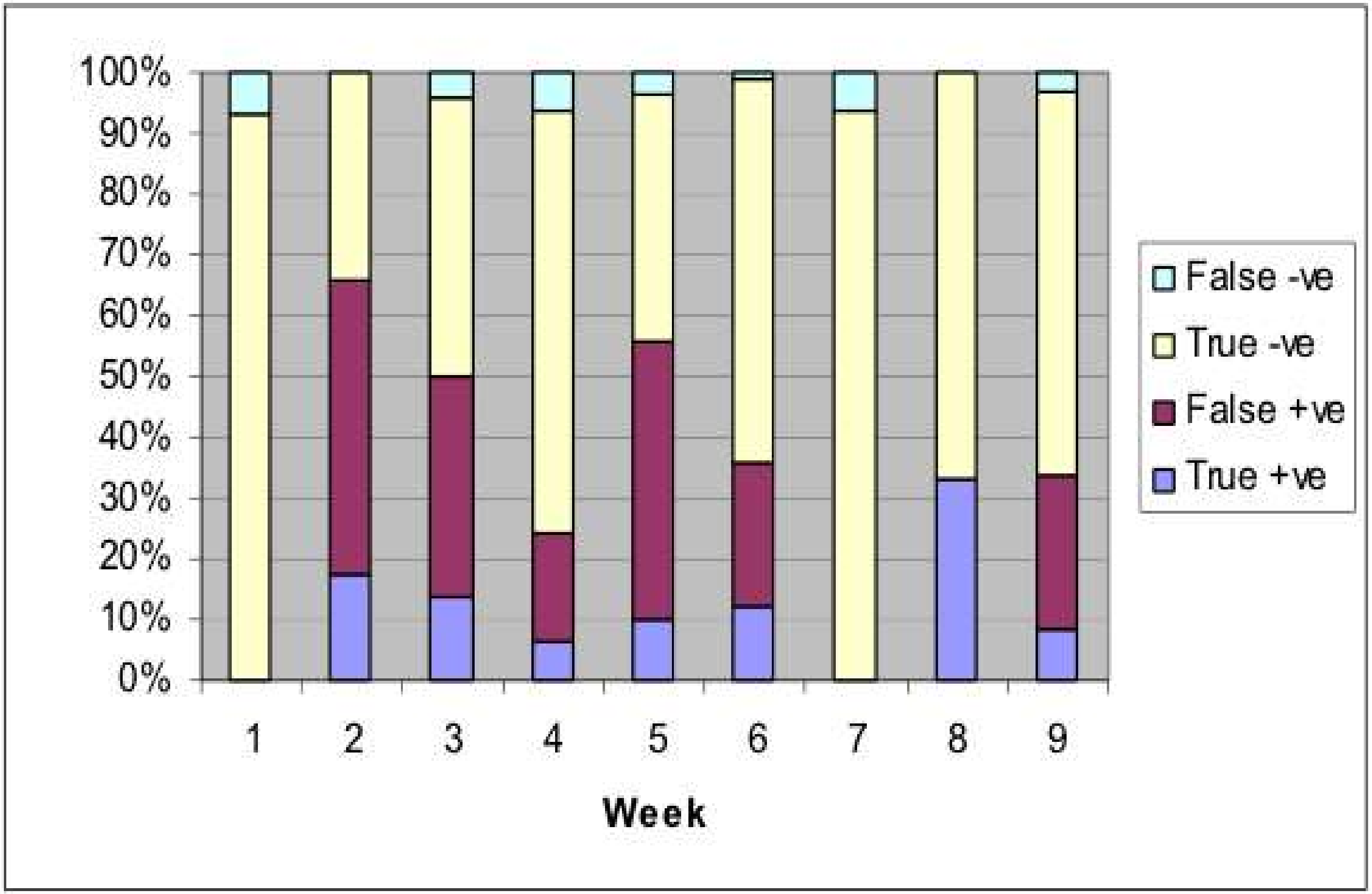
- Titles, abstract and authors contain different types of information
- Some articles contain only title & author information
  - e.g. Nature N&V, ZETOC alerts
- Hence have combine separate conditional probability tables
  - Bayesian chain rule



Performance

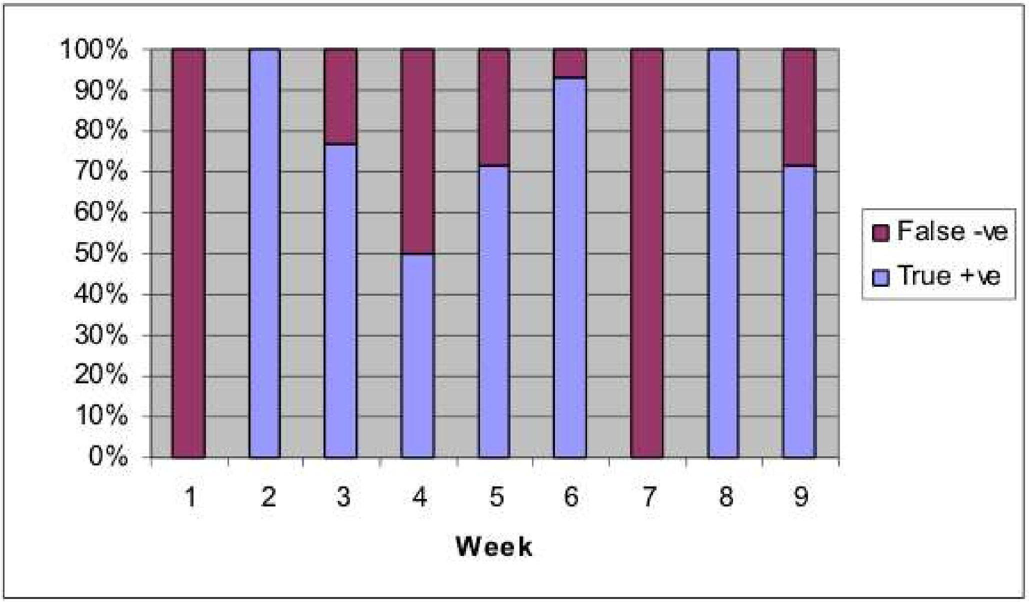
- Software tested by user over 10 weeks
- Classification based on title and abstract lumped together
- Training on every example – even if classified correctly
- 906 articles

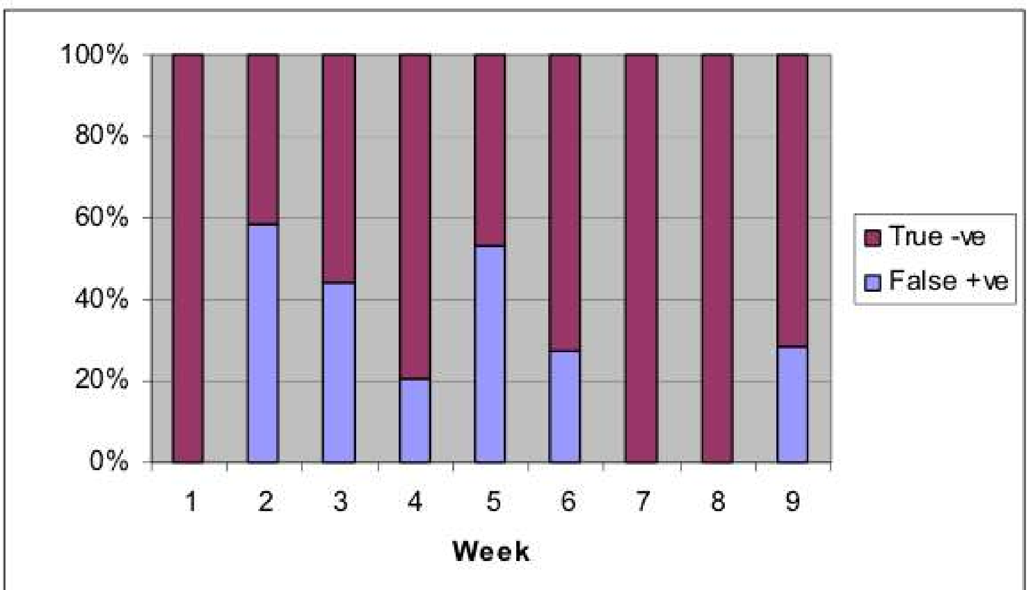
Classification



About 10%positives

Interesting and uninteresting articles



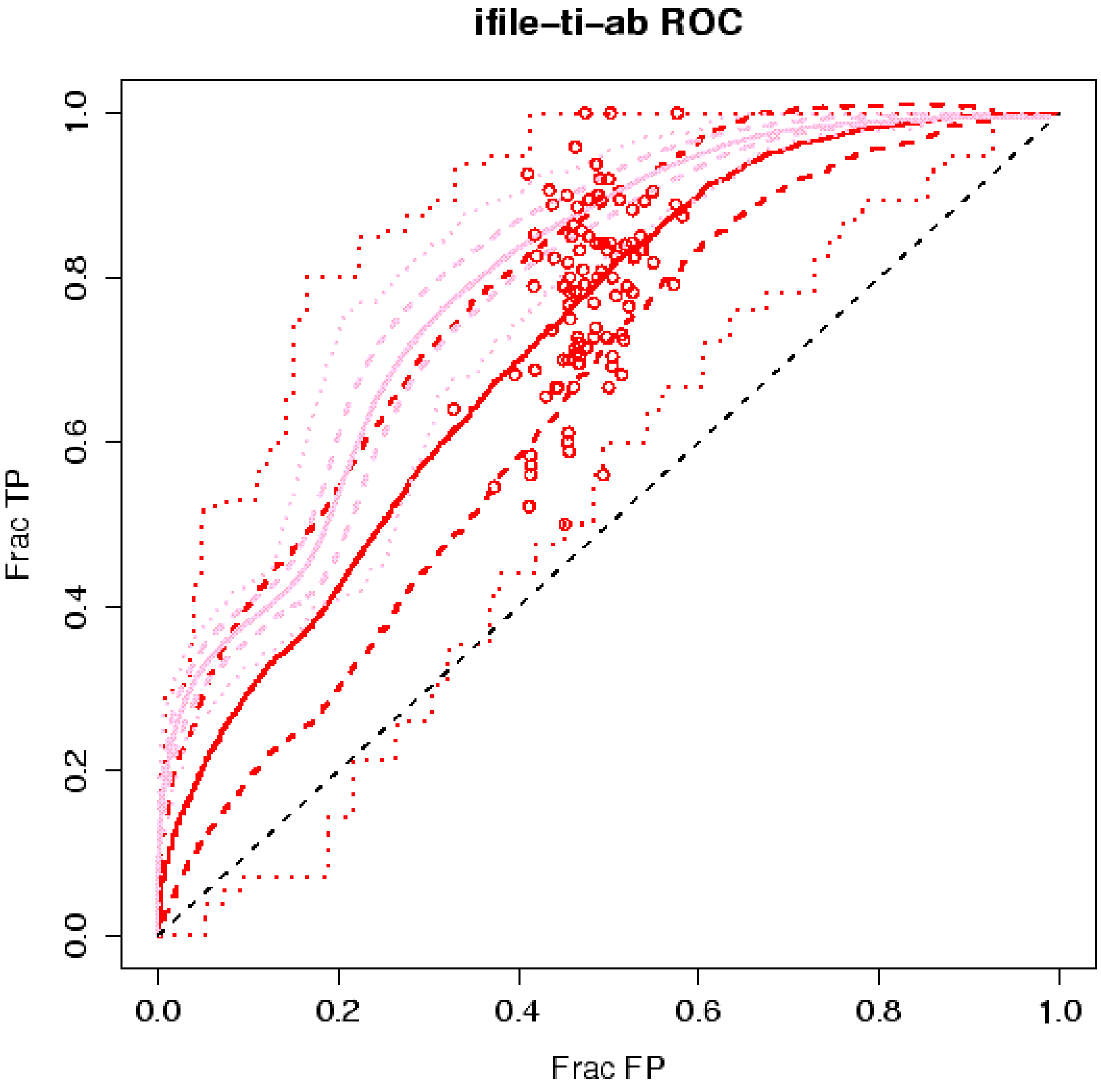


- Overall 72% of interesting articles classified correctly
- 68% of uninteresting articles classified correctly
  - i.e would have seen 32% of possible false positives

Crossvalidation study

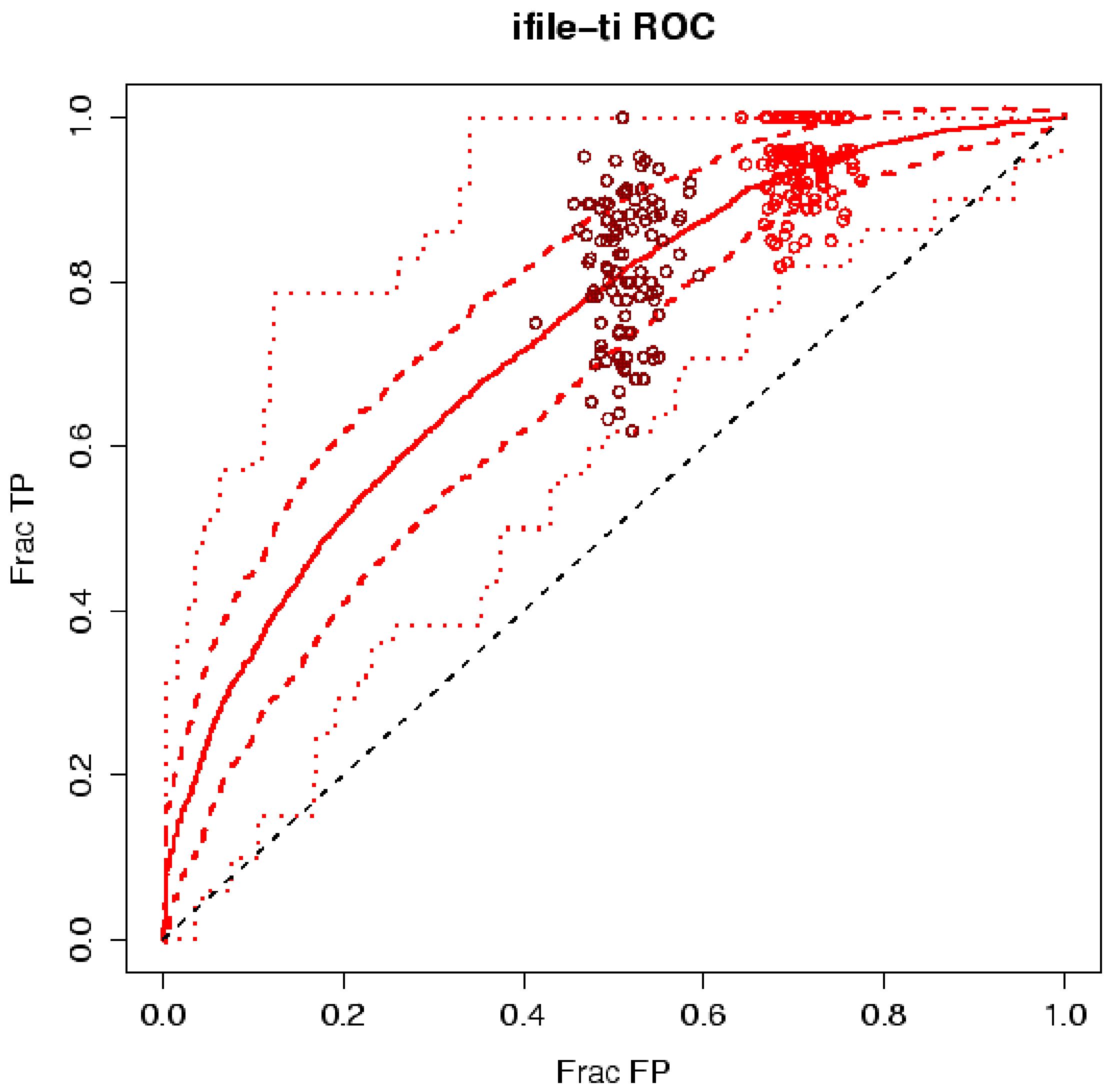
- Corpus of 2662 articles, 1047 with empty abstracts
- 218 interesting articles, 2444 boring
- Ten by tenfold crossvalidation procedure
- Naïve Bayes (ifile implementation) and CRM114 (another mail filter; more complex algorithm)

Naïve Bayes: title and abstract (lumped together)



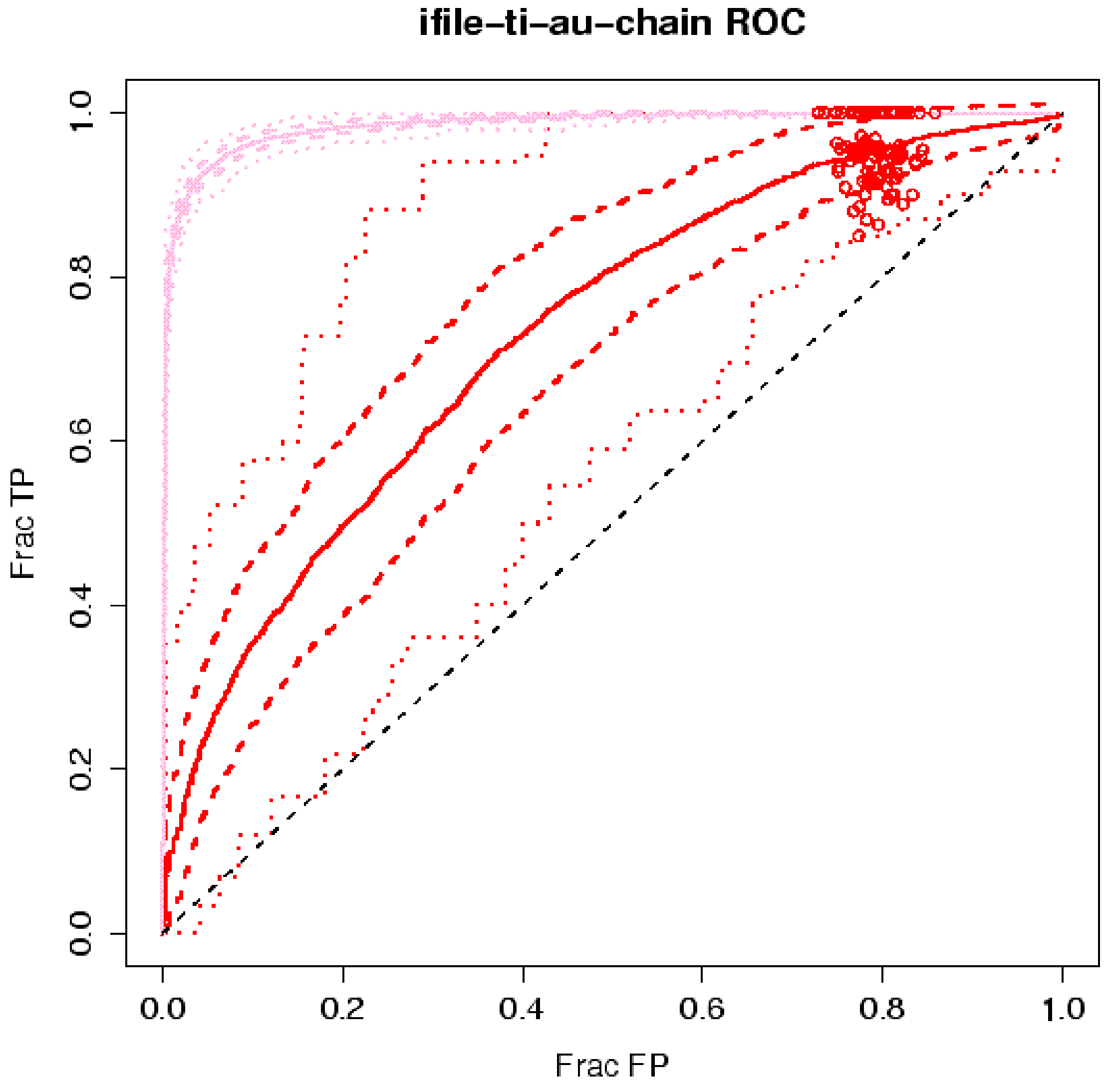
48±4% FP
78±11% TP

Naïve Bayes – titles only



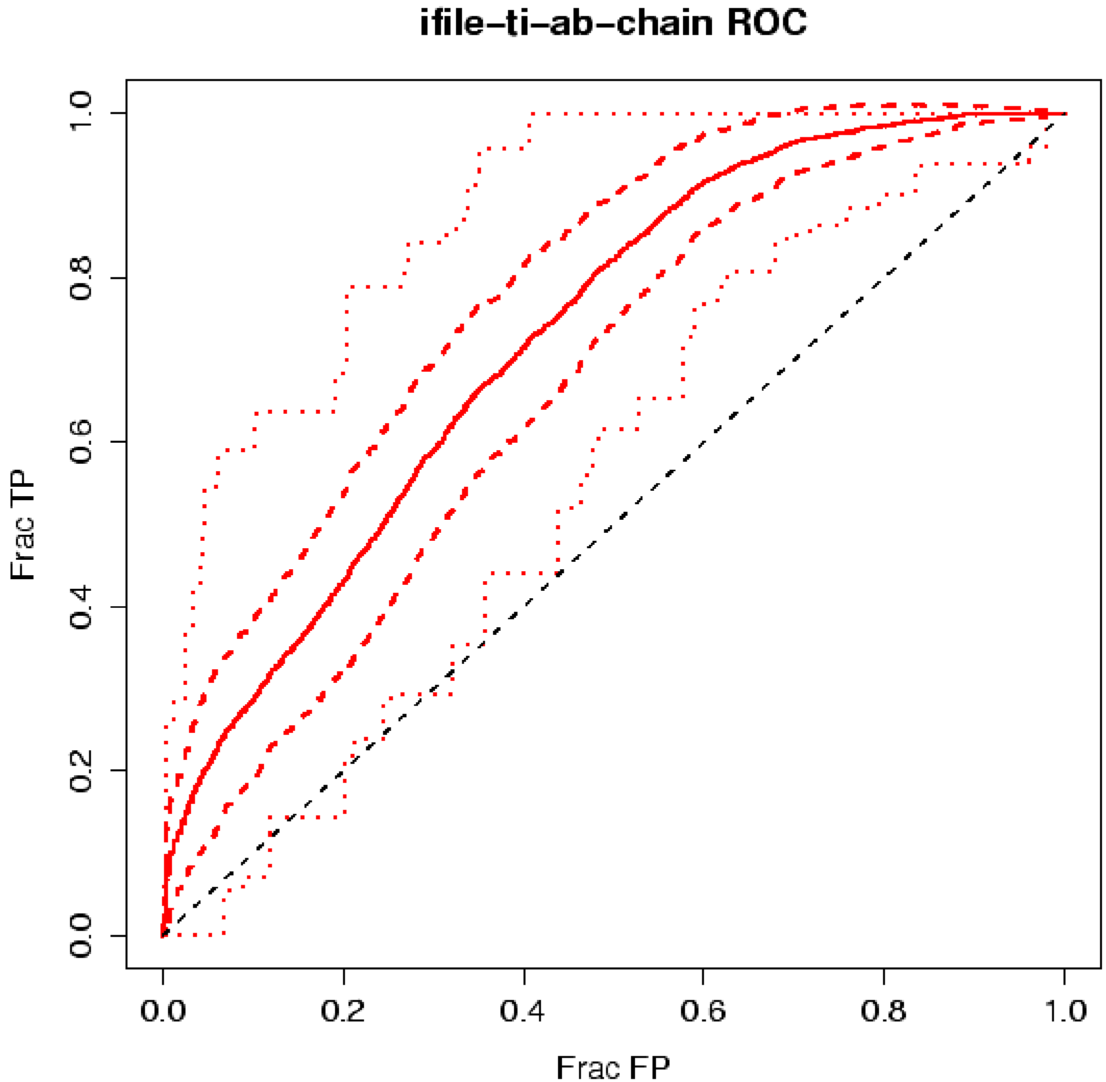
71±3% FP
94±5% TP

Naïve Bayes – titles and authors



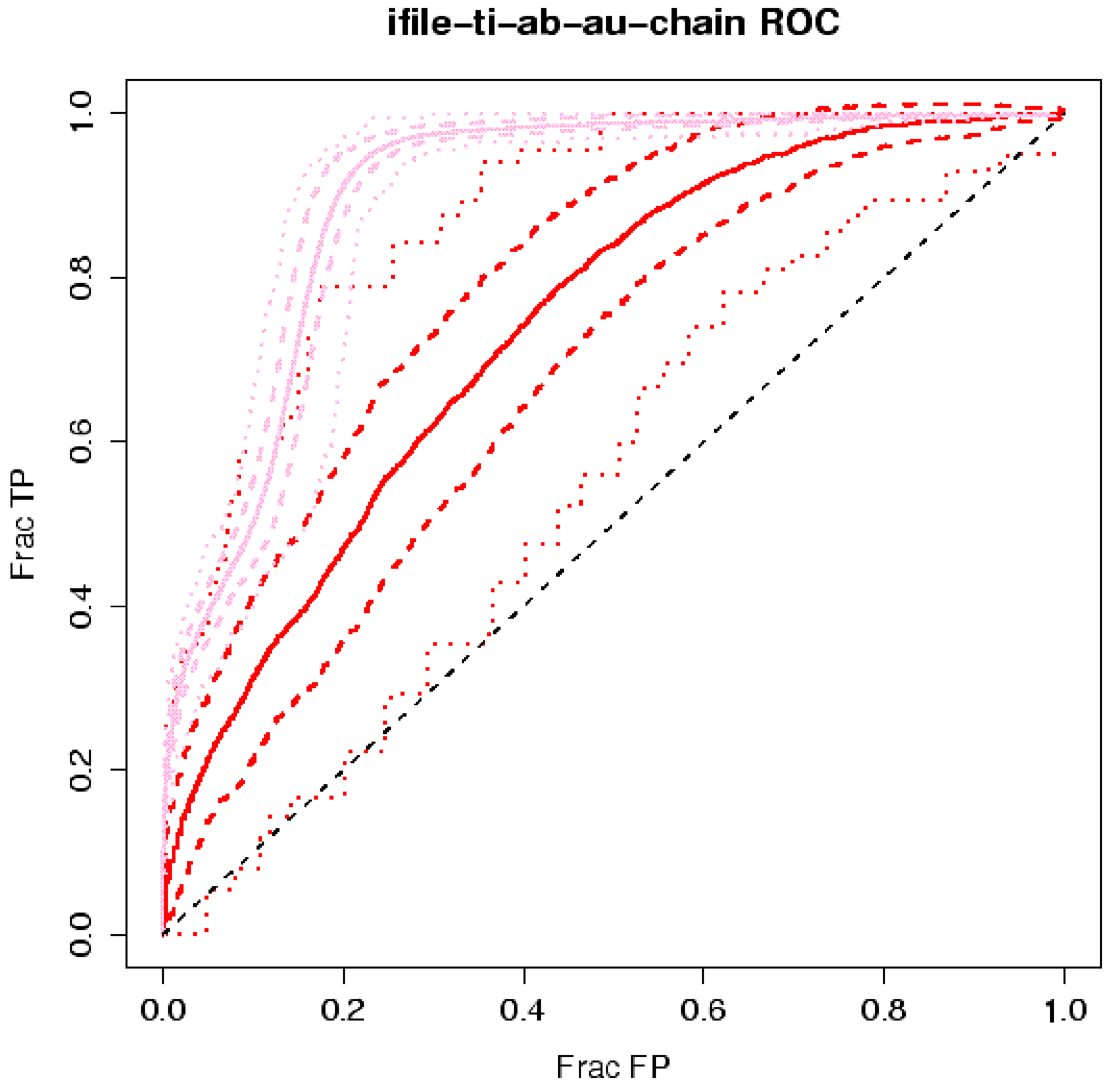
79±2% FP
96±4% TP

Naïve Bayes – Abstracts and Titles



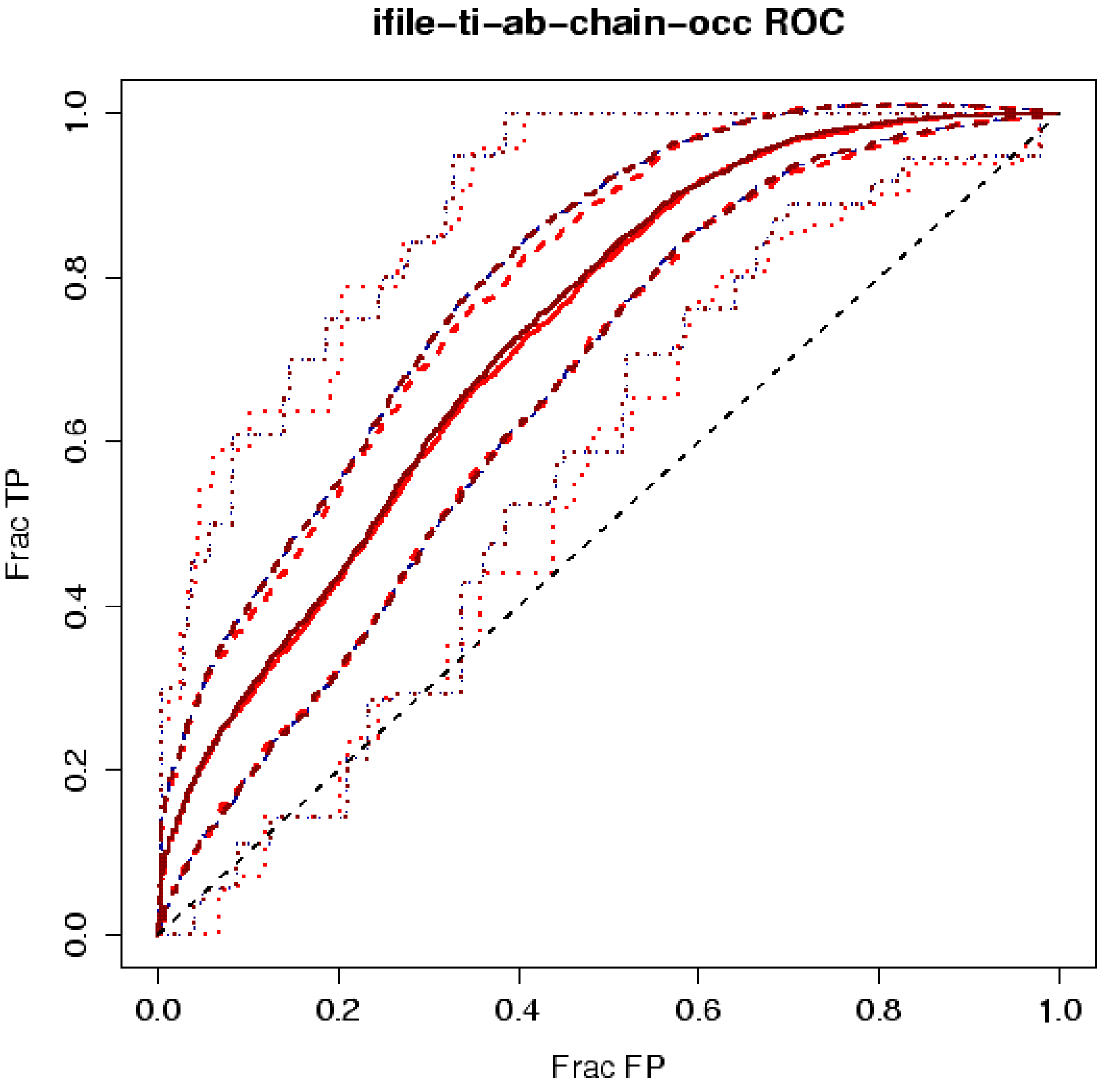
59±4% FP
90±7% TP

NB Titles + Abstracts + Authors

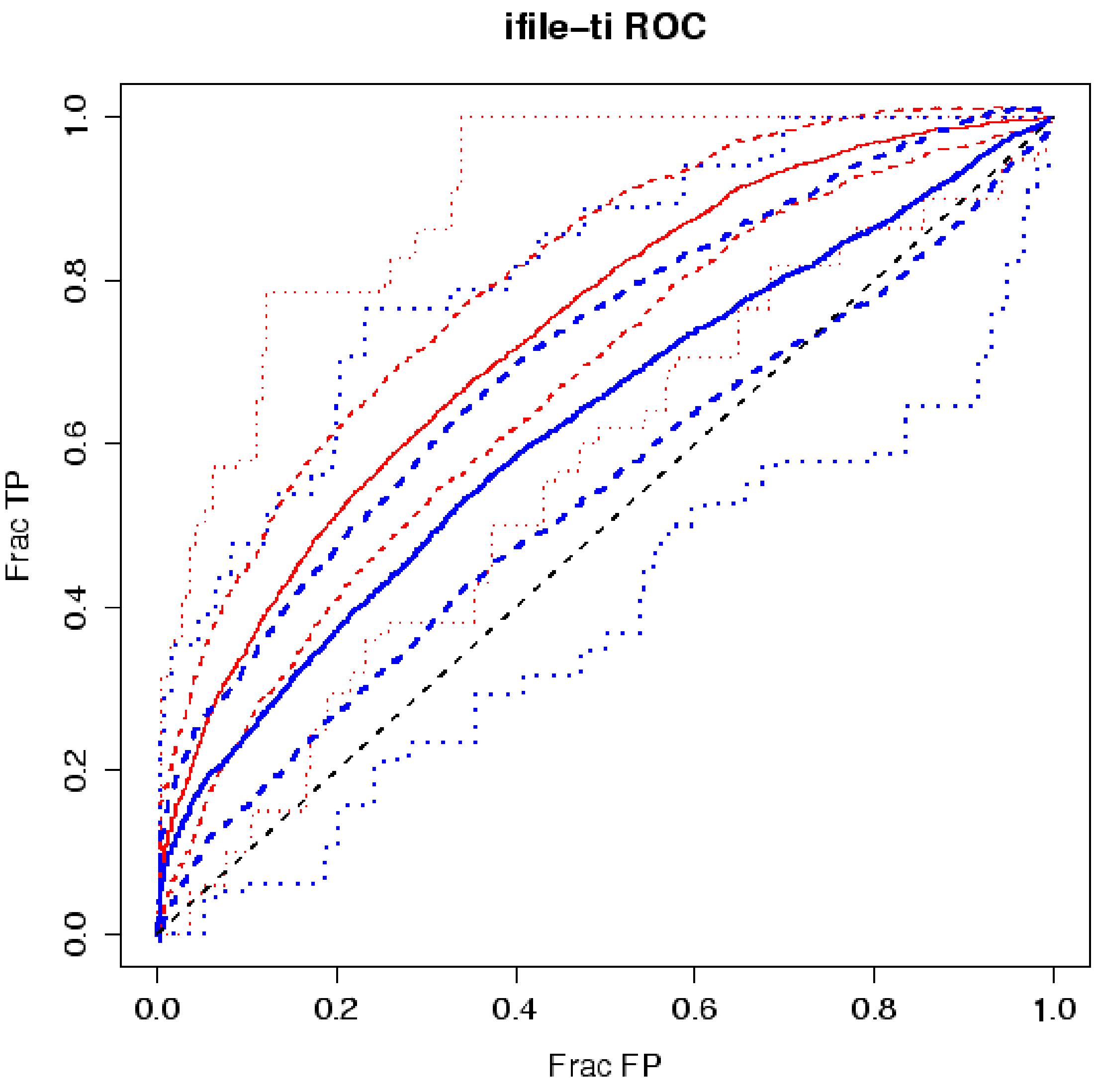


64±4% FP
93±6% TP

Naïve Bayes – occurrence vs counts



CRM titles vs NB titles



Conclusions

- The algorithm does cut down on the number of uninteresting articles to be skimmed for finding a given fraction of interesting articles
  - But performance is not great
- Performance on titles is comparable with title + authors/abstracts
- A more complex algorithm doesn't do as well
  - Overfitting?

The future

- Improve algorithm
  - star rating system?
- Performance improvements
- More data sources
- Corpus collection tool?
- Open source project