On-line Hierarchical Multi-label Text Classification

Jesse Read

September 7, 2007

The Problem

Learning to automatically classify text documents. Eg:

- Emails
- News Articles, Current Events (websites, RSS feeds)
- "Folksonomies" (Wikipedia, CiteULike)
- Bookmarks (Web browser, del.ic.ous, Google Bookmarks)
- Other (e.g. File System, Medical Text Classification)

Each of these examples is (or could be):

- Text
- Multi-label
- Organised in a Hierarchy
- On-line / Streamed (not Batch Learning)
- Affected by Human Interaction

Multi-label Classification

Given a label set $L = \{Sports, Environment, Science, Politics\}$;

"Single-label" (Multi-class) Classification

For a text document d, the task is to select a label $l \in L$

Multi-label Classification

For a text document d select a *label subset* $S \subseteq L$

E.g.:

Example	Labels ($S \subseteq L$)
Document 1	{Sports,Politics}
Document 2	{Science,Politics}
Document 3	{Sports}
Document 4	{Environment,Science}

Multi-label Classification

Done by transforming a multi-label problem into a single-label problem, i.e. with a **Problem Transformation method**:

1. (LC) Label Combination Method
2. (BC) Binary Classifiers Method
3. (RT) Ranking Threshold Method

Then employ a standard single-label algorithm on the resulting data.
E.g. : *Naive Bayes, C4.5, Bagging with C4.5, Support Vector Machines, k Nearest Neighbour, Neural Networks, AdaBoostM1.*
Then transform result back to multi-label representation.

1. Label Combination Method (LC)

Each combination of labels becomes a single label. A single-label classifier C learns to classify from the resulting combinations. One decision per label.

E.g.: (C) Document X either belongs to
Sports+Politics
or *Science+Politics*
or *Sports*
or *Science+Environment*

- May generate many unique combinations for few documents
- What if a document about Sports and Science turns up?
- Can run very slow if no. of unique combinations grows large

2. Binary Classifiers Method (BC)

Single-label [binary] classifiers are created *for each* possible label.
Multiple decisions per document.

E.g. Four classifiers $C_1 \cdots C_4$, one for each label. Document X
(C_1) belongs to *Sports*? YES/NO...
(C_2) belongs to *Environment*? YES/NO...
(C_3) belongs to *Science*? YES/NO...
(C_4) belongs to *Politics*? YES/NO...

- Slow, need as many classifiers as labels.
- Assumes that all labels are independent
- Often way too many labels are selected

3. Ranking Threshold Method (RT)

A single-label classifier C outputs a *ranking* of its confidence for each label.

E.g.: Document X

(C) is 95.5% likely to belong to *Science*

(C) is 81.2% likely to belong to *Environment*

(C) is 60.9% likely to belong to *Sports*

(C) is 21.3% likely to belong to *Politics*

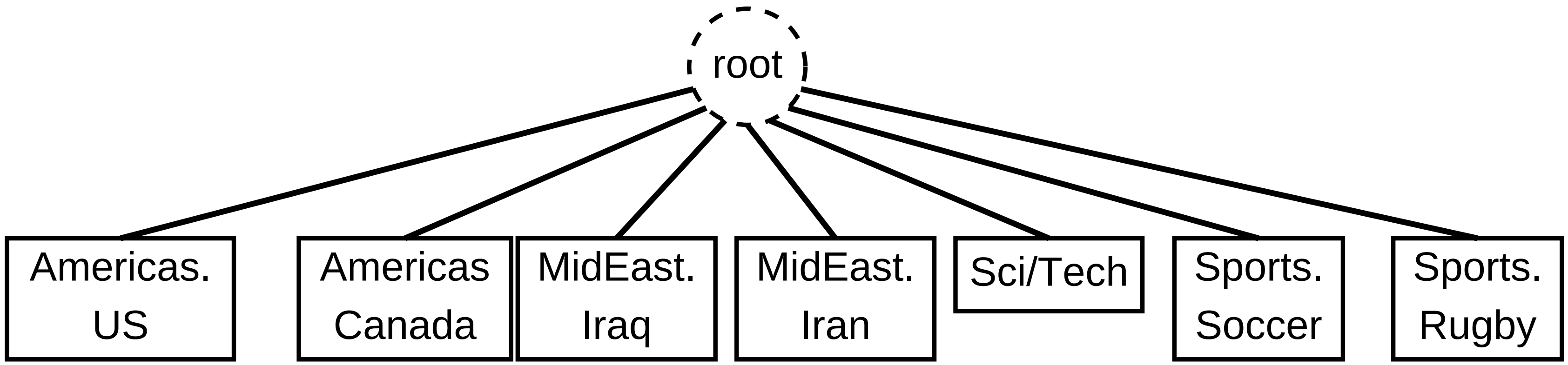
e.g. Threshold = 80.0%

- Not all single-label classifiers can output their "confidence"
- Assumes that all labels are independent
- Difficulty in selecting a good threshold
- Often the threshold encloses way too many labels

Hierarchical Classification (Option 1 - Global)

Uses 1 Problem Transformation method and single-label classifier.

Information about the hierarchy is incorporated into the process.



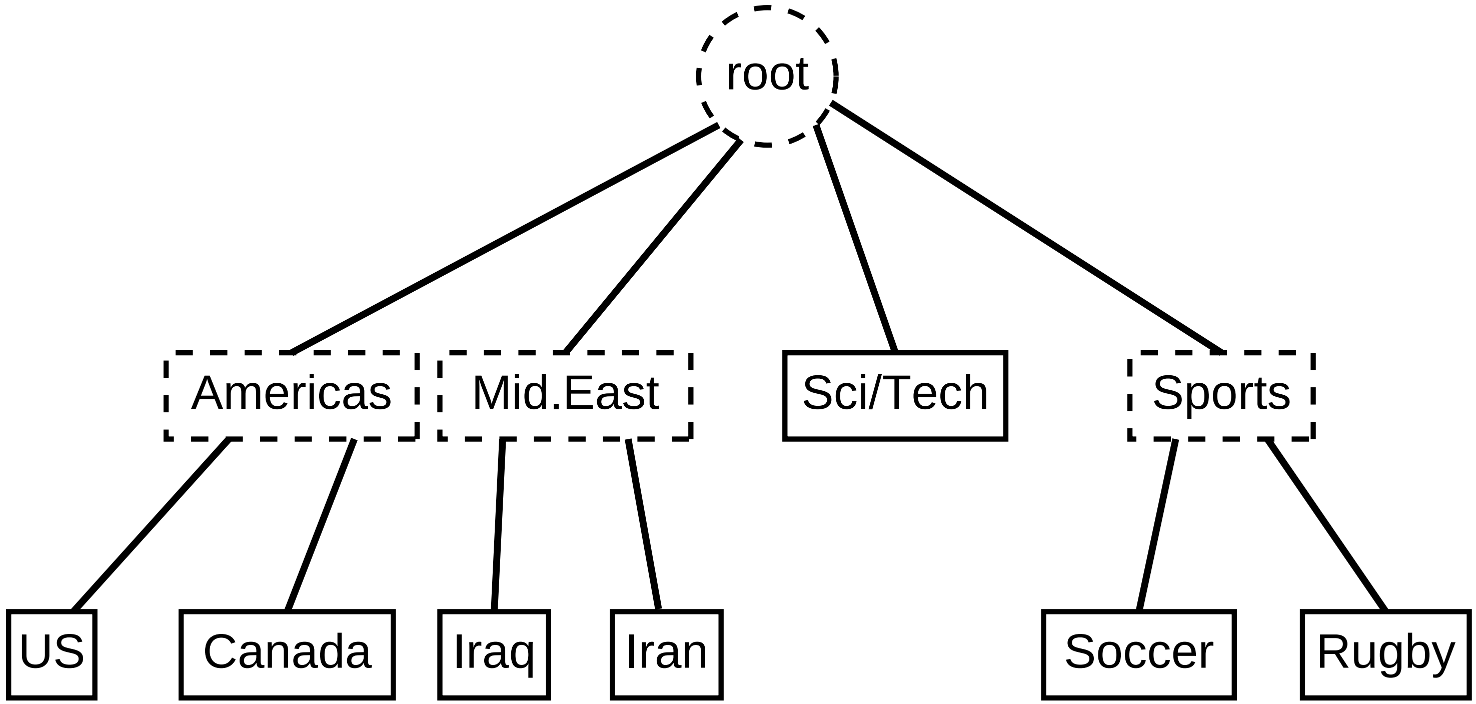
\+ Higher accuracy

– Can run very slow and use up a lot of memory

– Difficult to maintain; inflexible

Hierarchical Classification (Option 2 - Local)

Each internal node with its own Problem Transformation Method.

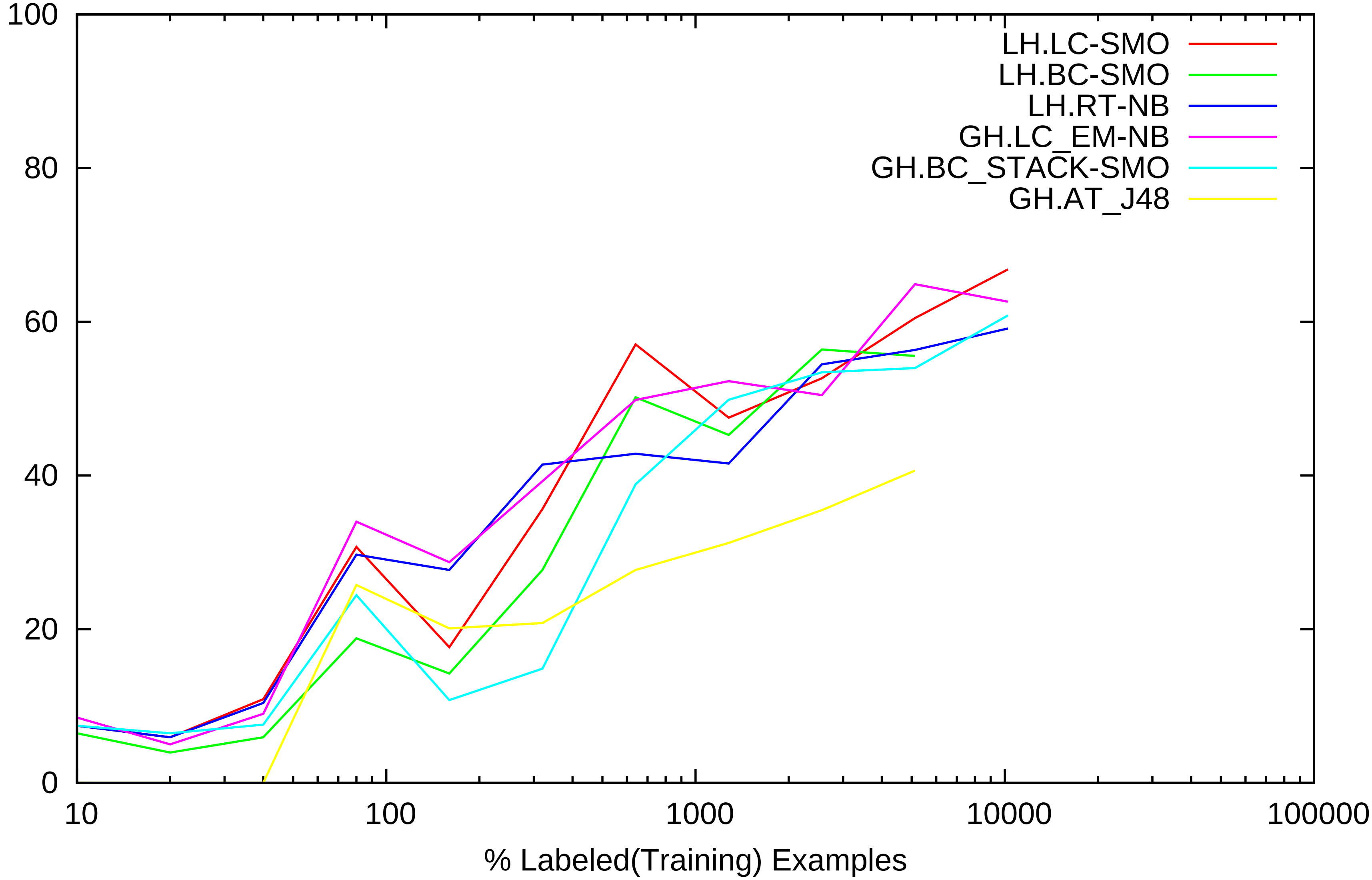


+ Divides up the problem: easy to maintain; efficient; intuitive

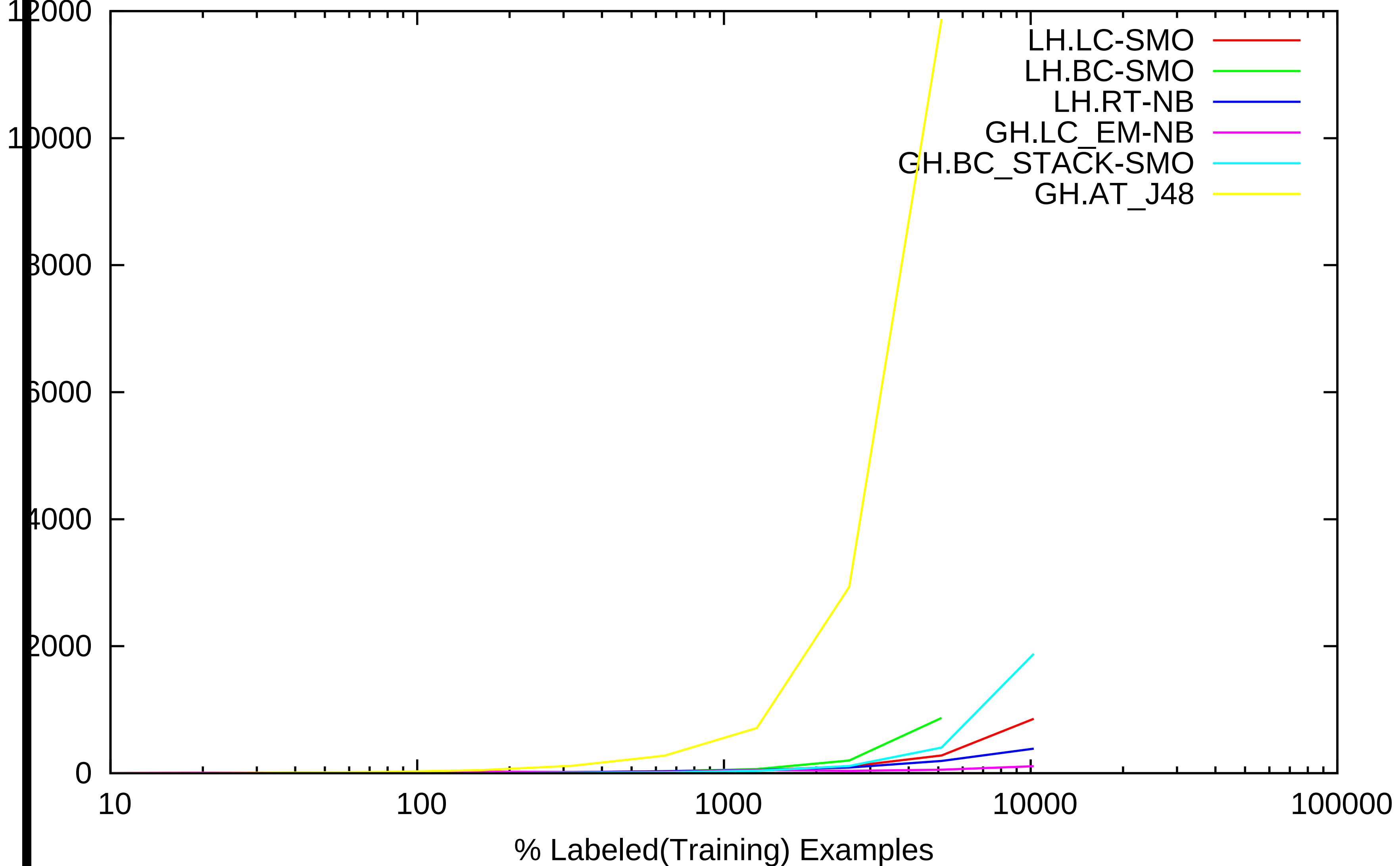
– Error propagation; accuracy unimpressive

– Overhead involved in setting up the hierarchical structure

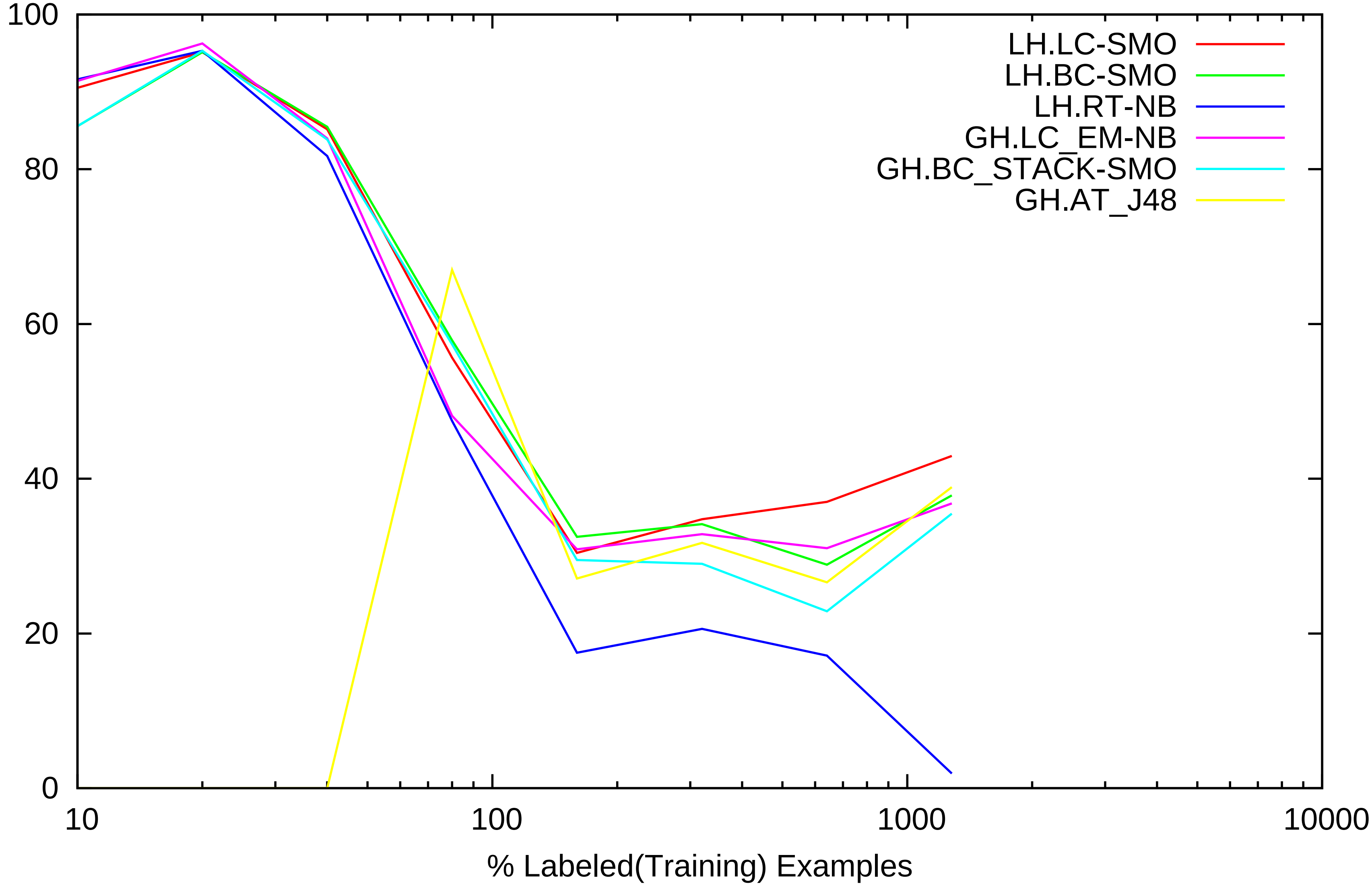
Experiments — 20Newsgroups — Accuracy



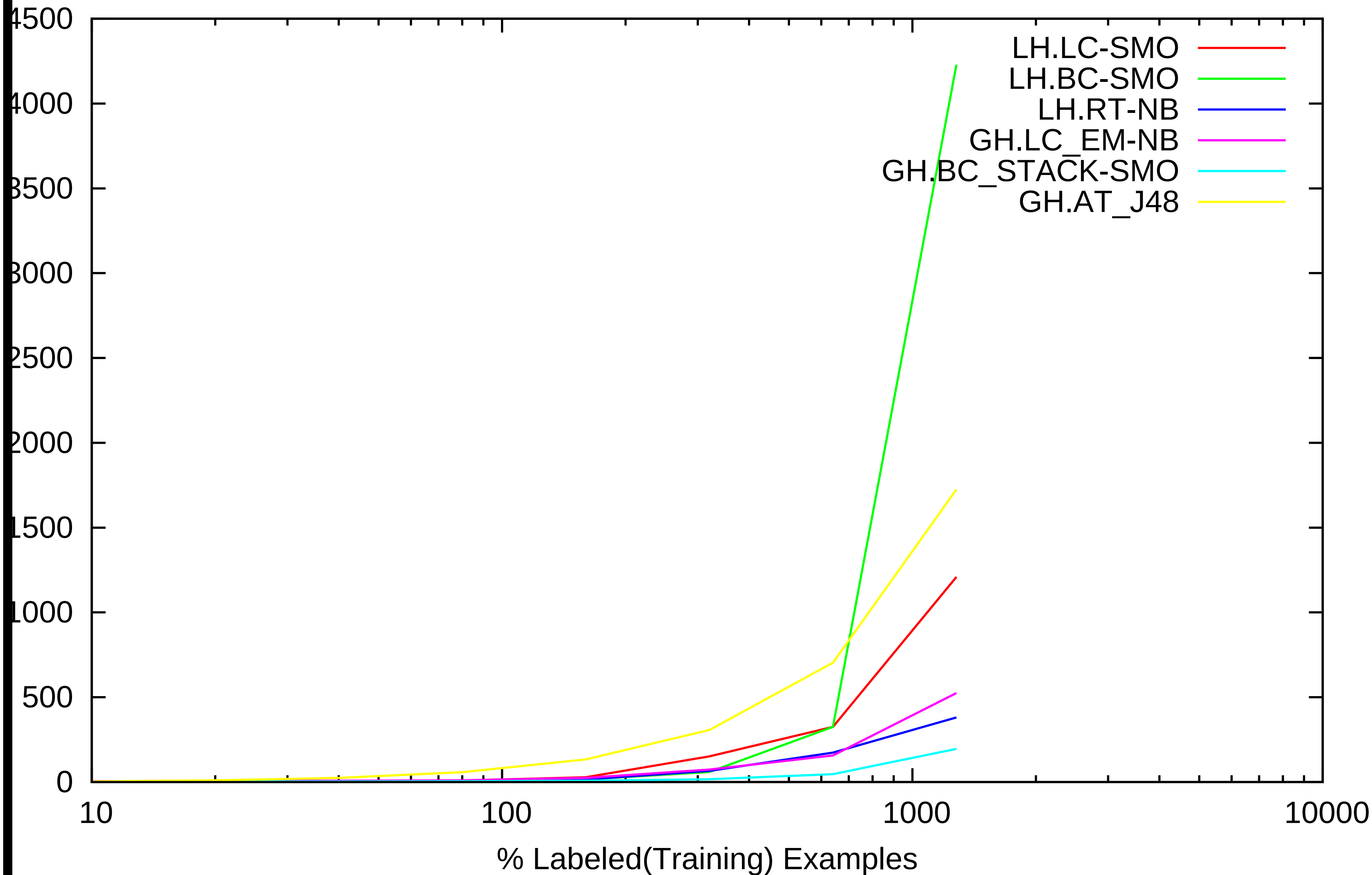
Experiments — 20Newsgroups — Build Time



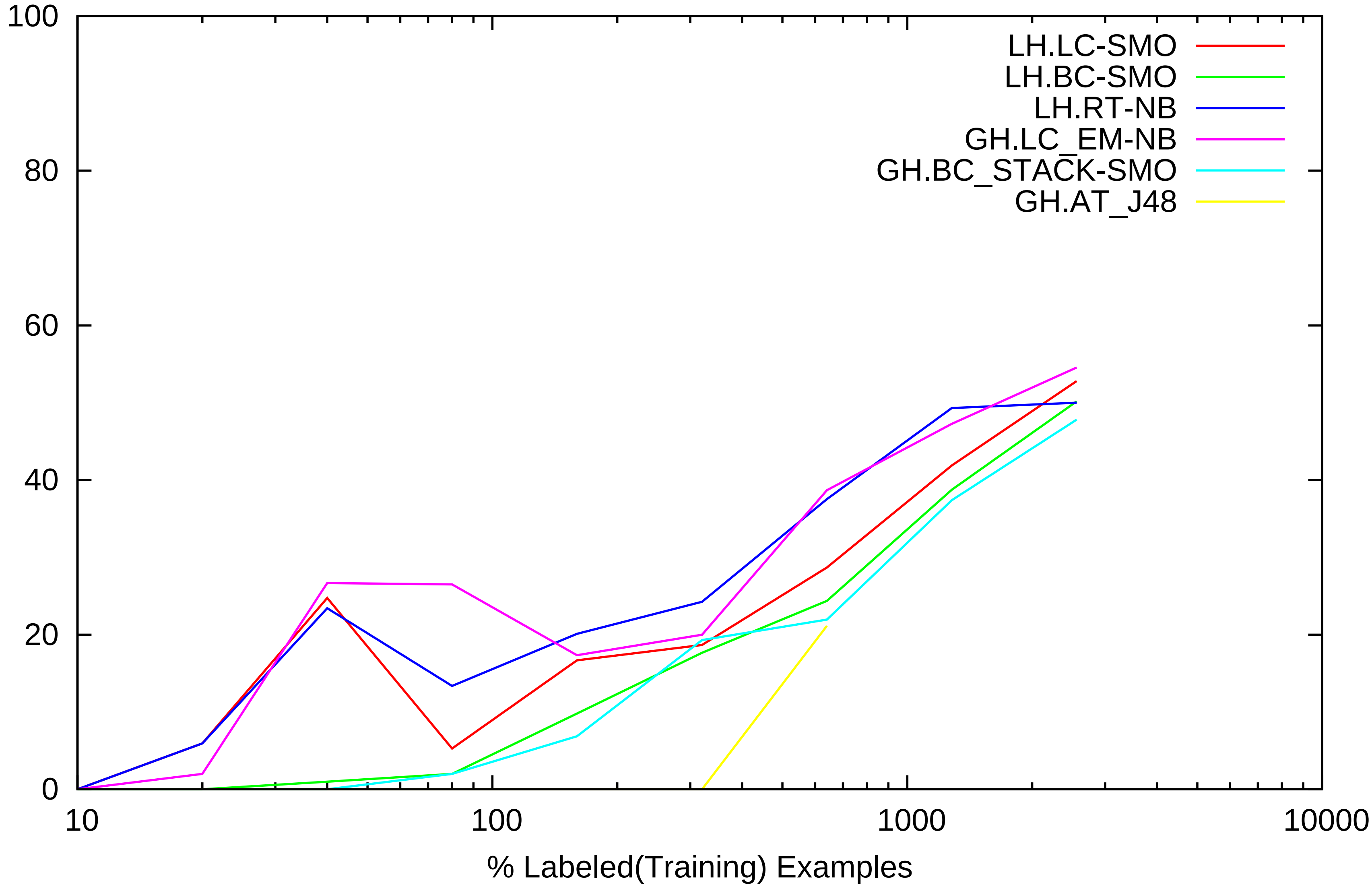
Experiments — Enron — Accuracy



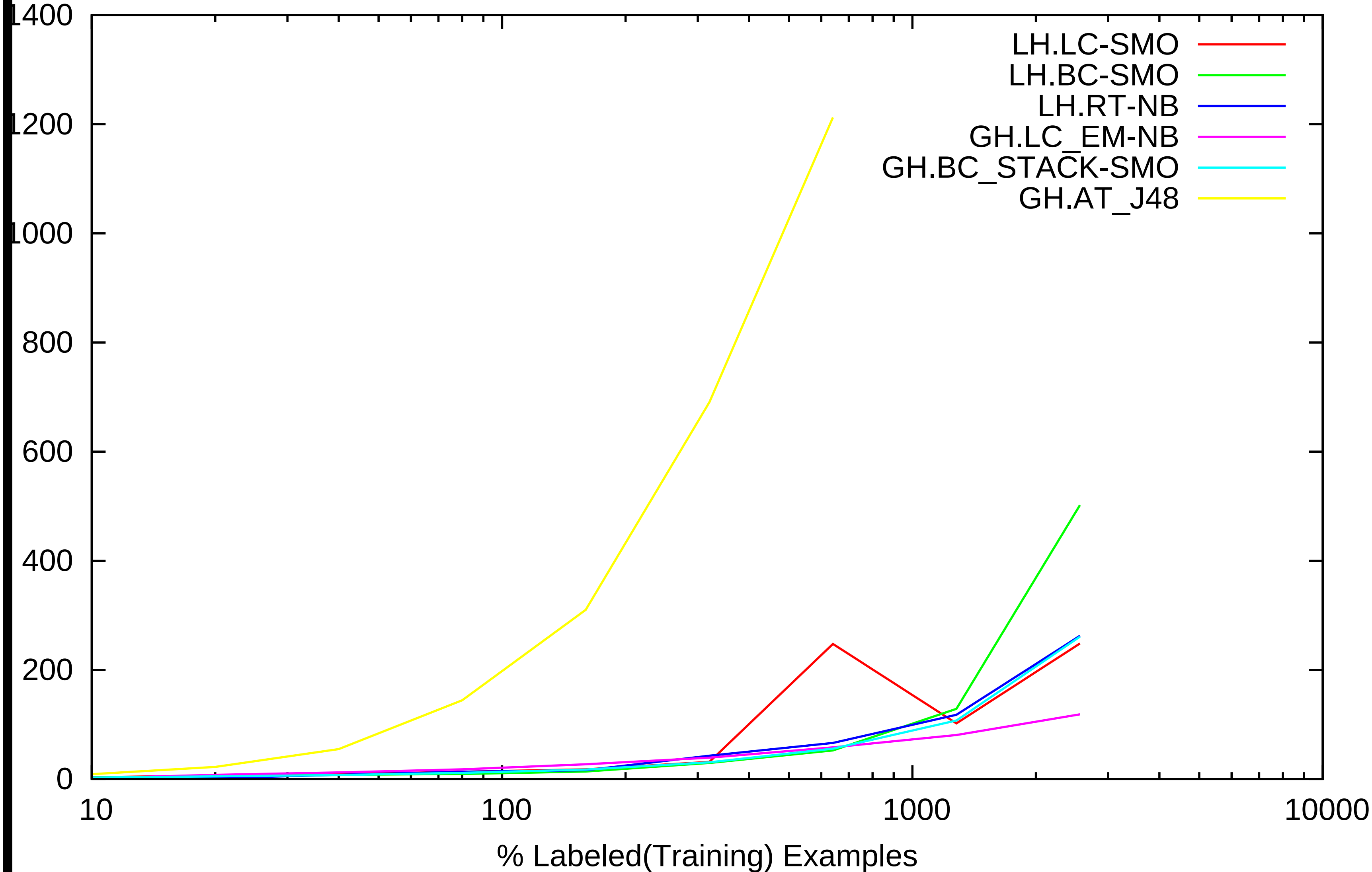
Experiments — Enron — Build Time



Experiments — NewsArticles — Accuracy



Experiments — NewsArticles — Build Time



Initial Conclusions

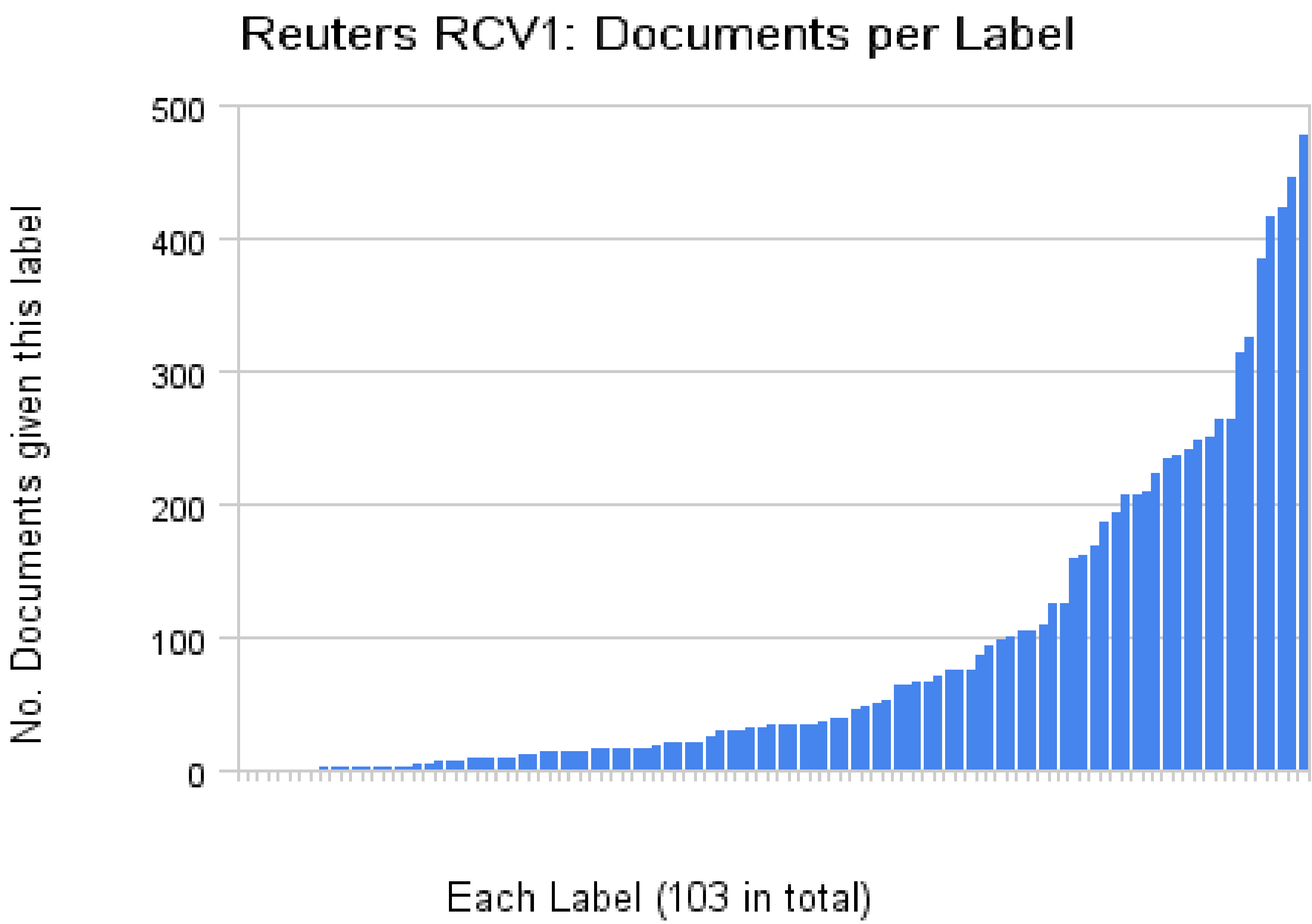
Performance is poor.

- All Problem Transformation methods have significant disadvantages
- Multi-label data is more complex than single-label data
- Multi-label text datasets can be very different, no method best for all
- **On-line data** is invariably susceptible to "Concept Drift"
- . . . but it is very costly to build / rebuild classifiers

Current Work

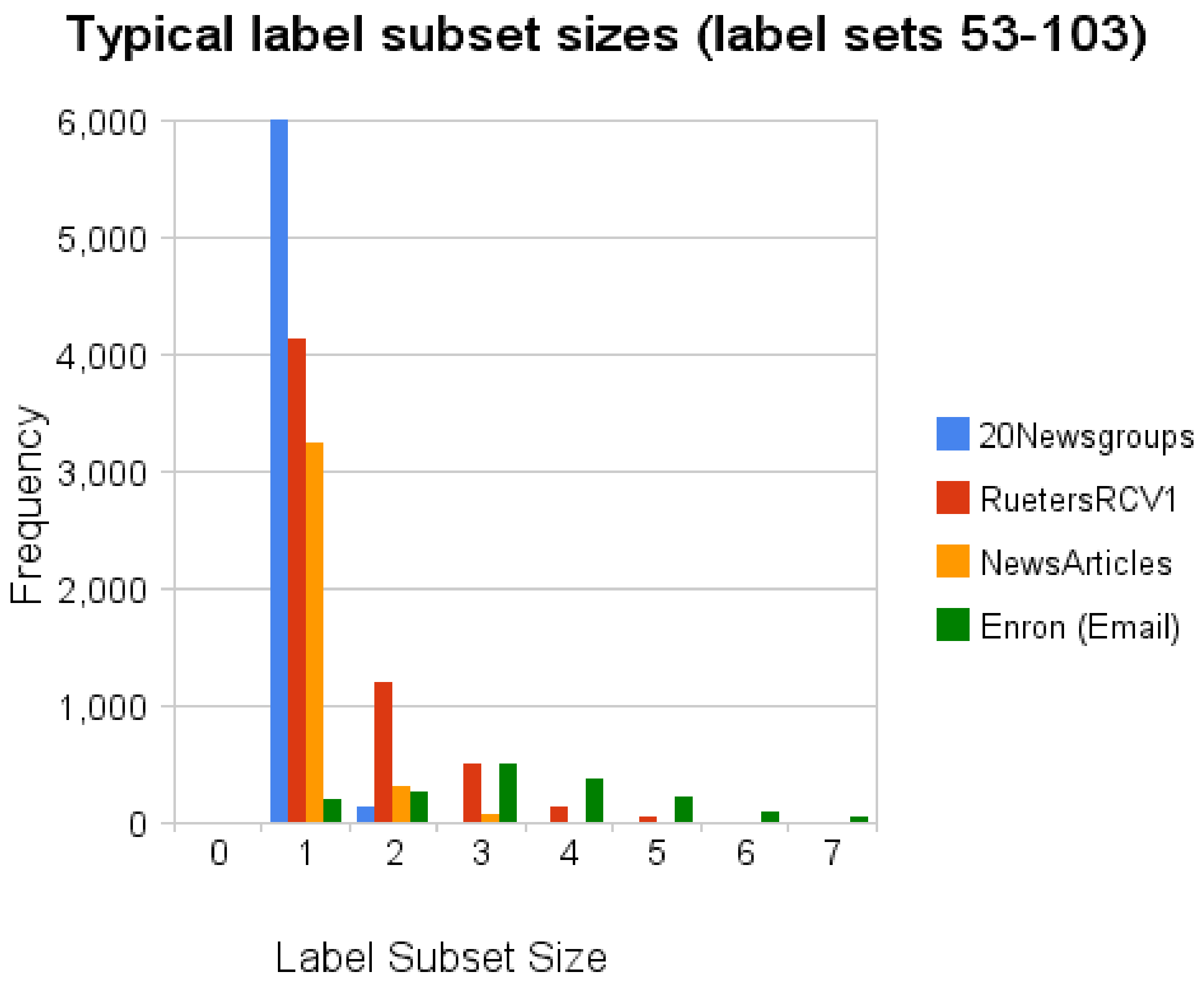
- Analysis and modelling of on-line hierarchical multi-label *text* data
- Analysing the performance/flaws of Problem Transformation methods
- Investigating adaptive and incremental learning methods

"Multi-label-ness": Documents per Label



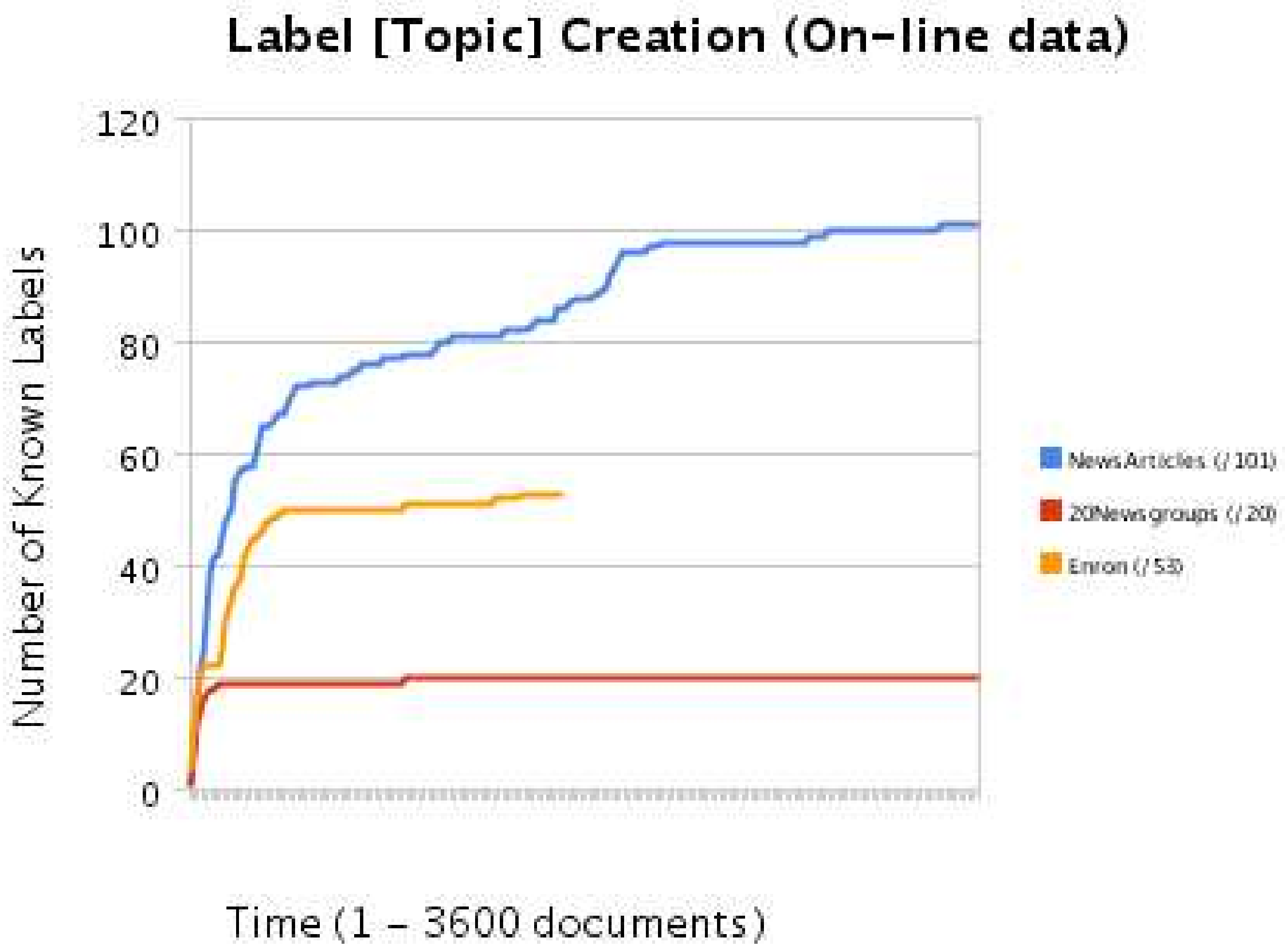
- 80/20 rule. Typically most labels used not used very often.

“Multi-label-ness”: Labels per Documents



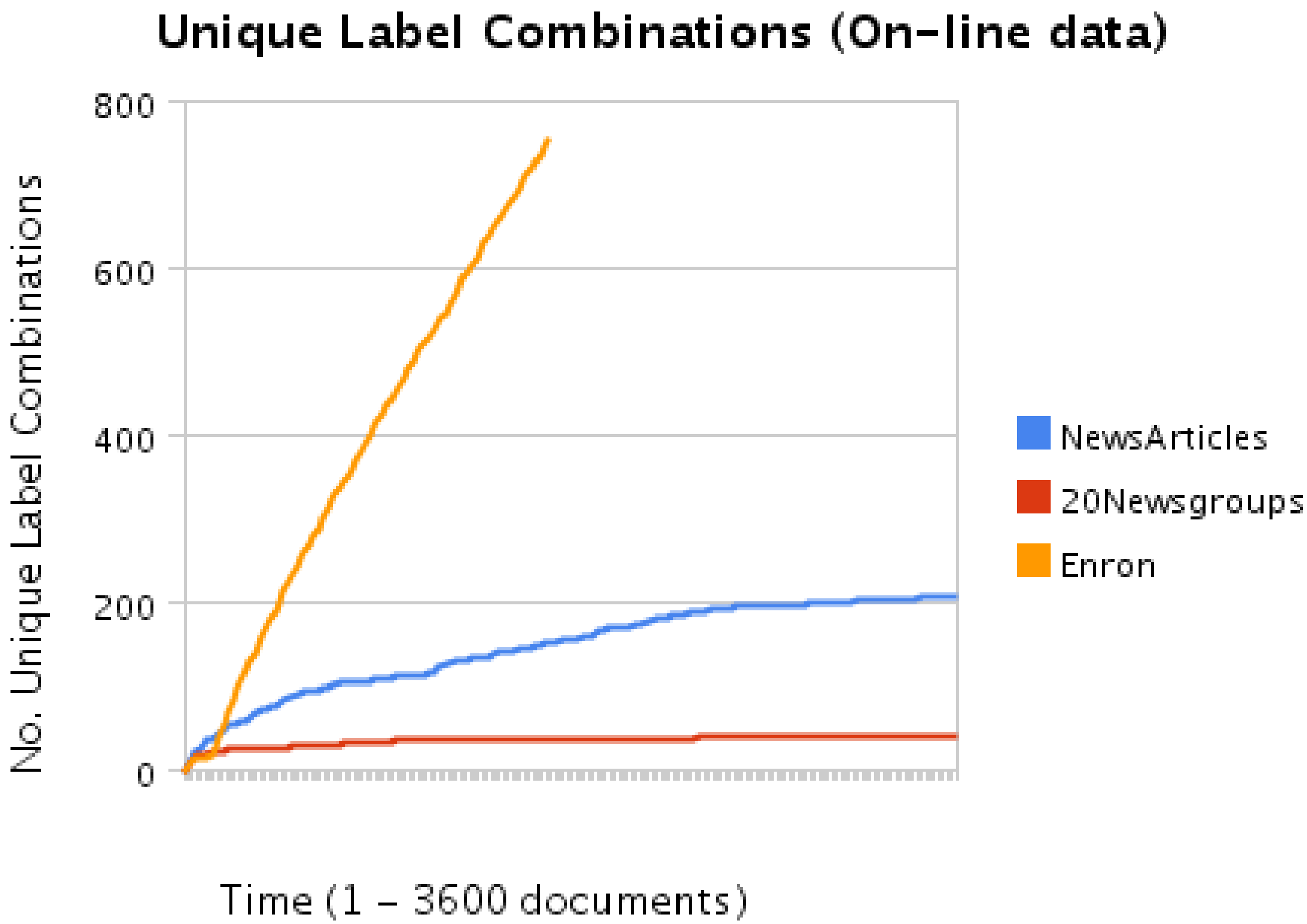
- Most documents have only a few labels.

On-line data: Creation of Labels Over Time



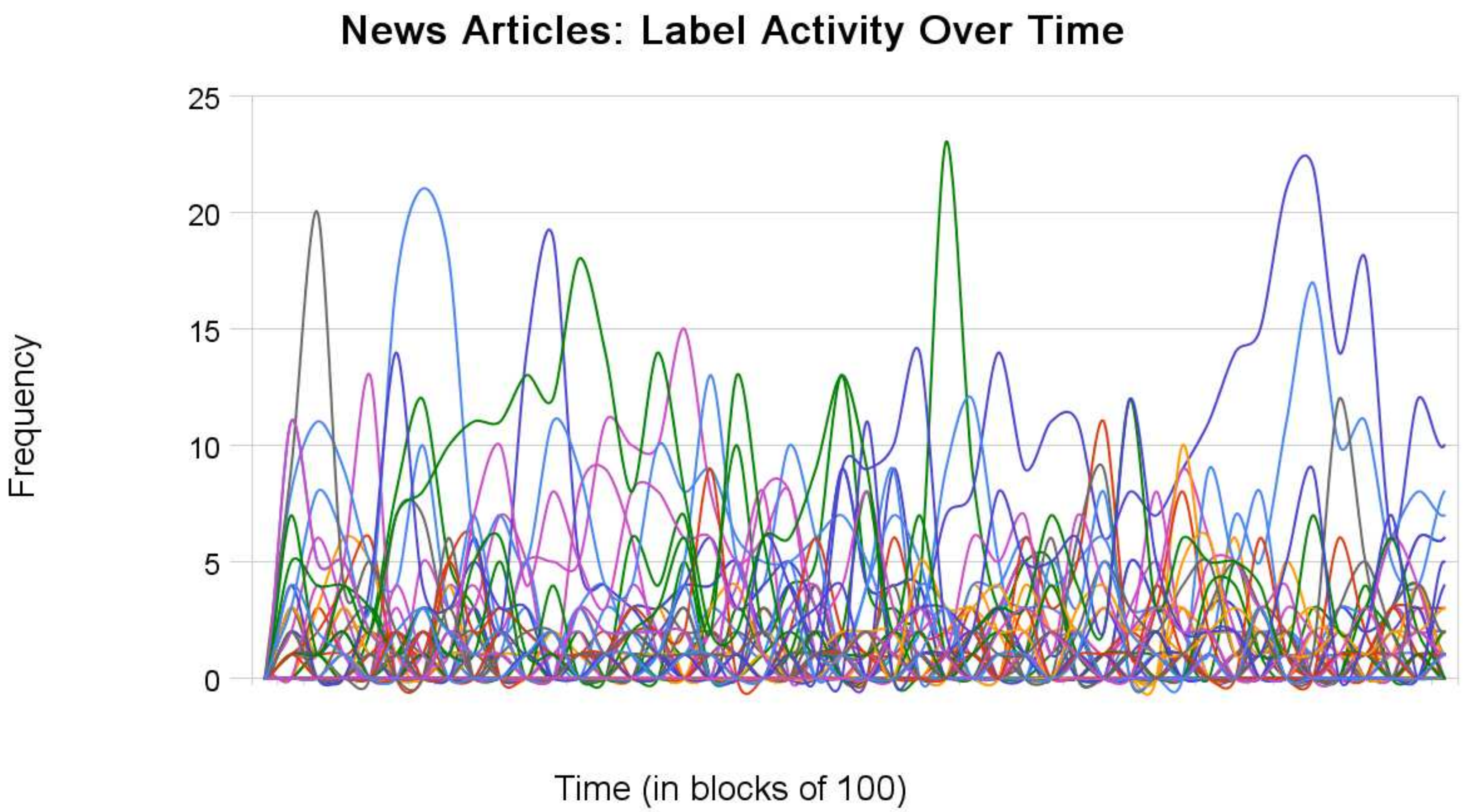
- Most labels are used for the first time (created) very early on.

On-line data: Label Combinations Over Time



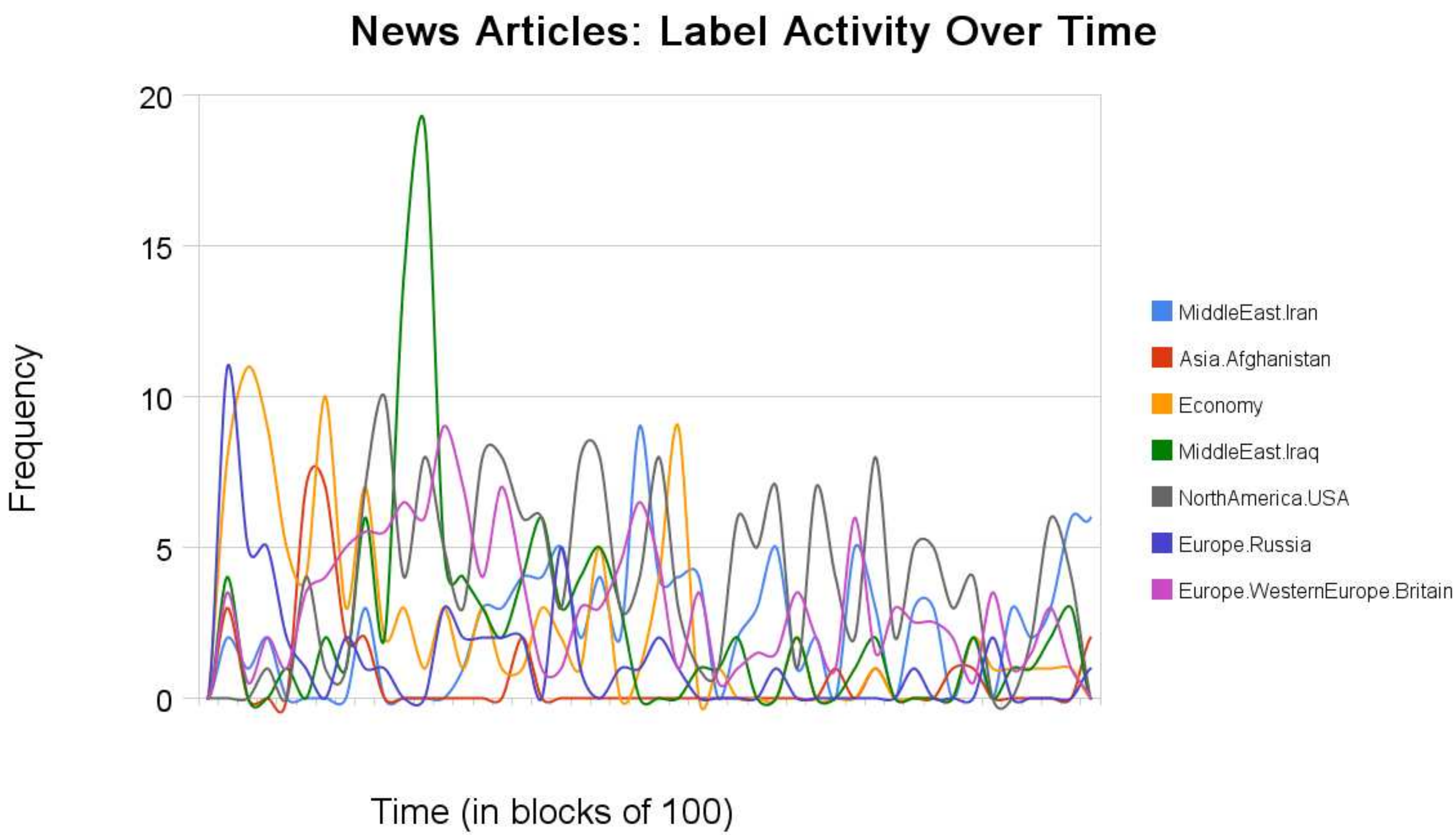
- New label combinations continue to appear for some time.

On-line data*: Label Activity Over Time



- Labels occur and reoccur in "bursts"
- → Topic/"burst" detection*

On-line data*: Label Activity Over Time



- Label often co-occur in bursts.
- Labels may be unused for periods of time

Other Things I found

- Some labels are particularly troublesome
- Some label combinations are particularly troublesome
- Some Problem Transformation methods do better or worse depending on variations of:
  - The length and type of text documents
  - The no. of training examples seen
  - The no. of possible labels it can choose from
  - The no. of unique combinations of those labels
  - Etc.

Future Work

- Continue analysis
- Improve Problem Transformation methods
- Design a novel hierarchical multi-label classification framework, for on-line text data streams, able to adapt to and learn from human interference (manual labelling).

. . . Questions? . . . Comments?