Web Spam Challenge 2008

Data Analysis School, Moscow, Russia

K. Bauman, A. Brodskiy, S. Kacher, E. Kalimulina,
R. Kovalev, M. Lebedev, D. Orlov, P. Sushin, P. Zryumov,
D. Leshchiner, I. Muchnik

The Data Used

- Web graph
  - Host graph (114K hosts)
  - The full Web graph (105M URLs) wasn't used
- Sample pages – up to 400 (first crawled) pages per host, in WARC format (12M)
- Spam judgments – for ~3.75% of hosts
- Features provided by organizers

The Host Graph

- 114529 hosts
  - 453 hosts labeled as spam (by 2006 and 2007 judgments)
  - 4995 hosts labeled as normal
- Weight of an edge is the number of inter-host links

Pre-computed Feature Vectors

- Two obvious direct features:
  - Number of pages in host
  - Host name length (in bytes)
- Features, proposed in the articles:
  - L. Becchetti, C. Castillo, D. Donato, S. Leonardi, R. Baeza-Yates:
    - "Using Rank Propagation and Probabilistic Counting for Link-Based Spam Detection"
  - C. Castillo, D. Donato, A. Gionis, V. Murdock, F. Silvestri:
    - "Know your Neighbors: Web Spam Detection using the Web Topology"
  - Link-based features (*the list on the next slide*)
    - For the front page and the page with the maximal PageRank
  - Content-based features (*the list on the second slide*)
    - For the front page and the page with the maximal PageRank – plus averages and standard deviations over all host pages

Link-based Features

- Assortativity coefficient (degree / average degree of neighbors)
  - "degree" here is undirected (in-degree+out-degree)
- Average in-degree of out-neighbor pages
- Average out-degree of in-neighbor pages
- Number of in-neighbor pages at distances 1 to 4 (4 features)
- Out-degree
- PageRank
  - in the doc graph with no self-loops, with a damping factor of 0.85, with 50 iterations
- Standard deviation of the PageRank of in-neighbors
- Fraction of out-links that are also in-links
  - a page with no out-links has a value of 0
- Number (approx.) of in-neighbor hosts at distances 1 to 4 (4 features)
- TruncatedPageRank using truncation distances 1 to 4 (4 features)
- TrustRank (obtained using 3,800 hosts from ODP as trusted set)

Content-based Features

- Number of words in the page
- Number of words in the title
- Average word length
- Fraction of anchor text
- Fraction of visible text
- Compression rate of the page
- Top 100, 200, 500, 1000 corpus terms precision (4 features)
- Top 100, 200, 500, 1000 corpus terms recall (4 features)
- Top 100, 200, 500, 1000 query terms precision (4 features)
- Top 100, 200, 500, 1000 query terms recall (4 features)
- Entropy of trigrams
- Independent trigram likelihood

The Challenge Submission Overview

- A boosted vote of few large margin classifiers
  - There were 13 partial classifiers combined
- Voters built by separate groups of features
- Two overall classifiers were built
  - using different training procedures and data

Groups of Features Used

- Host graph analysis (scores extension)
- Distribution of host pages compression rate
- Content features (word frequencies)
- Features readily provided by organizers

Host Graph Analysis

- The features were an elaboration of those in:
  - T. Abou-Assaleh, T. Das, 2007
    - Extention and Propagation of manual Spam scores
- They were extensively reworked due to
  - 10-fold increase of this year’s host graph size
  - Relatively low amount of spam scores available

Compression Rate Features

- GZIP compression rates for every page of a host:
  - Were put into bins: [0, 0.5), [0.5, 1), [1, 1.5) ... [9.5, 10), [10, +)
  - Makes 63 features – three per each of 21 bins:
    - The bin's page count, average compression rate and standard deviation
  - Normalization of the features:
    - for the mean=0, std=1 on our training set

Compression Rate Results

- SVMLight with linear kernel was used
  - The features were normalized (mean = 0, std=1)
- Classification results:
  - **F1 = 27.99% (R = 35.38%, P = 23.15%)**

	Normal	Spam
Predicted Normal	3495	137
Predicted Spam	249	75

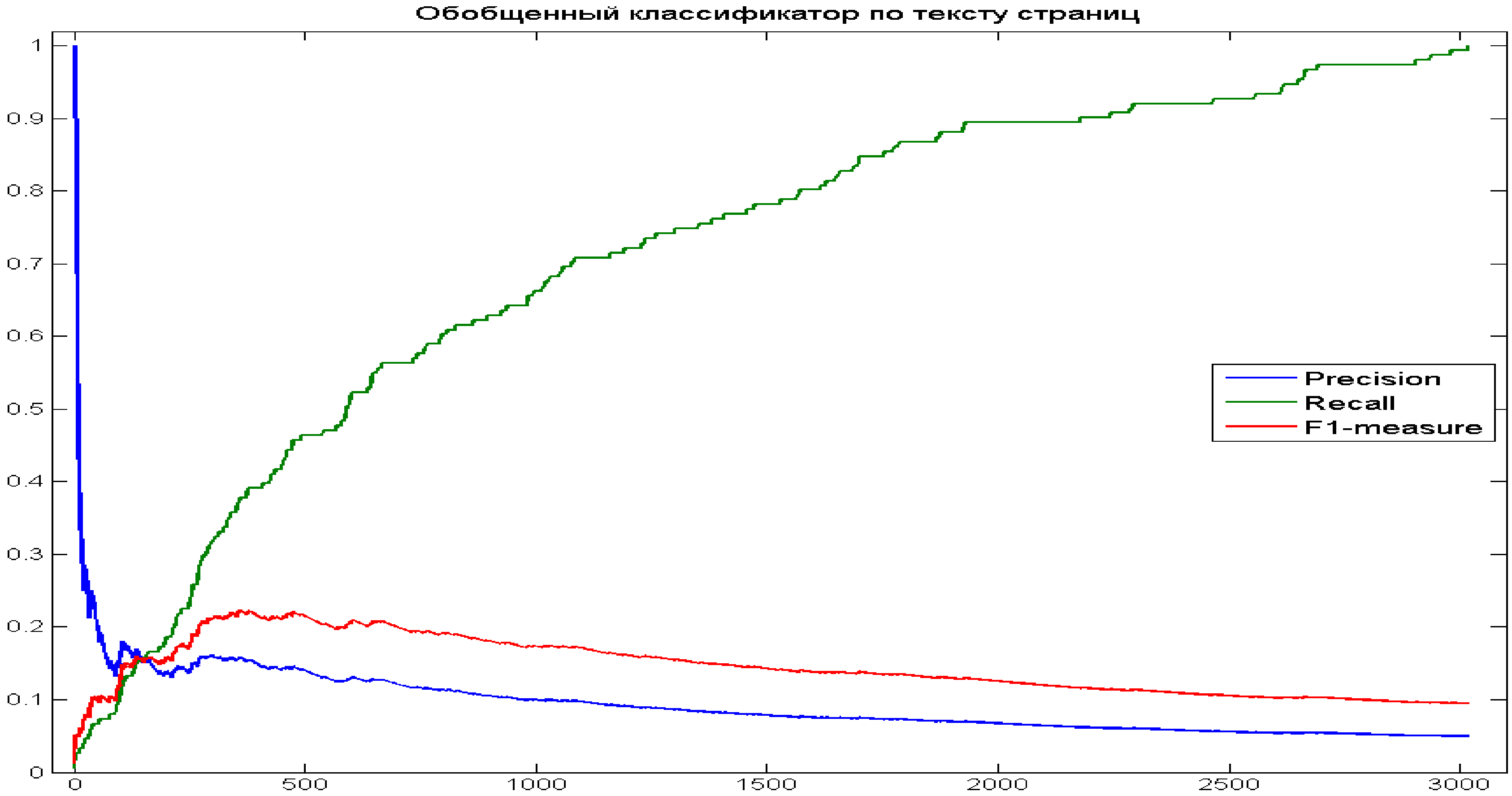
- Just 8.63% of non-labeled hosts classified as spam with these settings

Word Frequency Features

- Words* in <title>, <meta> (keywords, description), <anchor> and <body>
  - Computed the average of *log(1+ wc)/log(1+pl)*
    - where wc – word count, and pl – page length
  - Separately for each word and each tag
  - Also used query log frequencies in word counting for pages
- Feature selection
  - Should be present with at least 10% of either spam or normal hosts
  - A threshold of 75% discriminating power between classes by Student test
- SVMLight with linear kernel used for classification

- * a 'word' is any sequence of letters, numbers and some special symbols
  - **lowercased**
  - **Numbers excluded**
  - **Examples: seed, foo, 123f, $100, бабай**

Classification by Word Frequencies



- Max F1 = 22.35% (R = 39.07%, P = 15.65%)

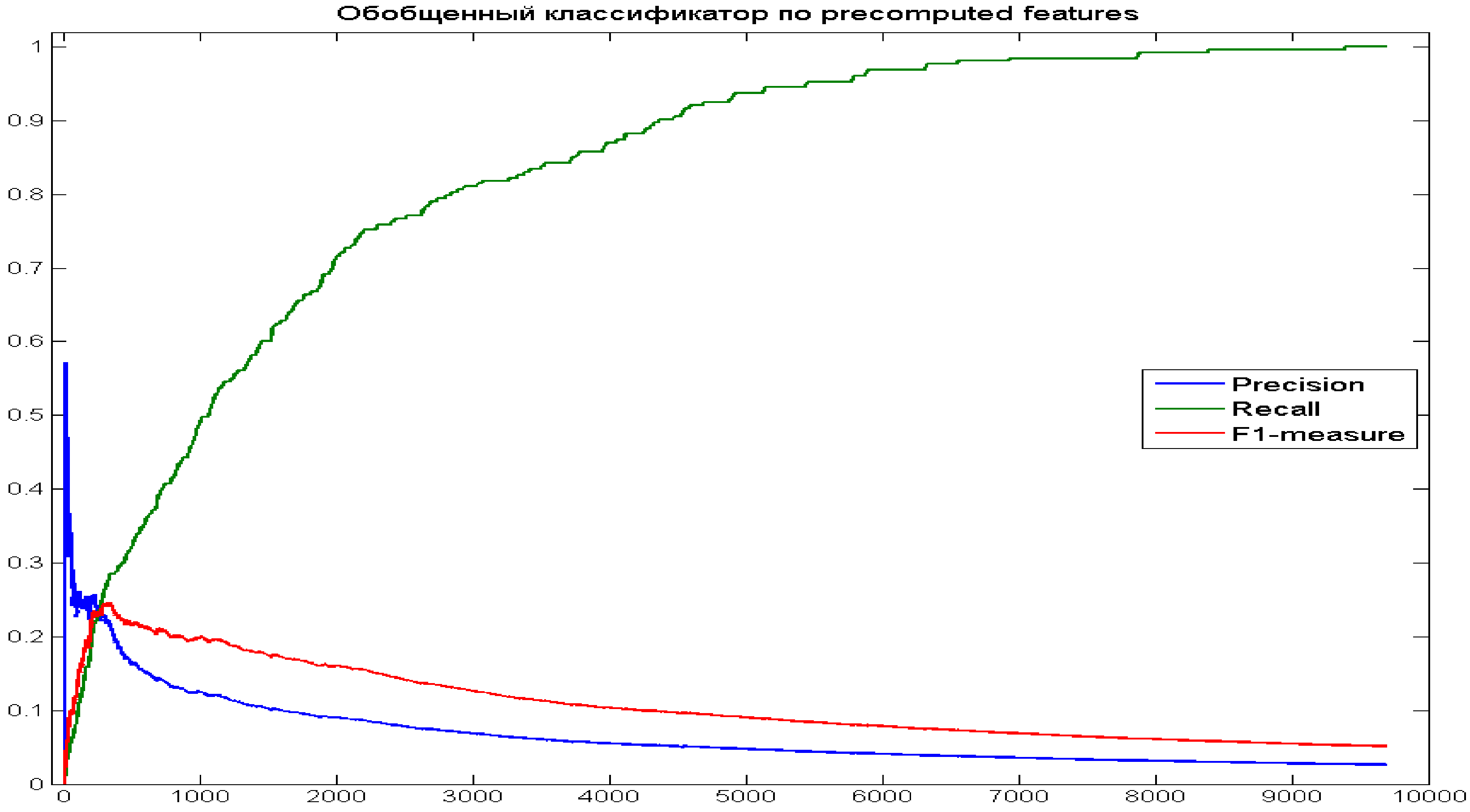
Classification by Pre-computed Features

- Total of 276 features provided
- The training data were made by extending host labels from ones given
  - Total of 9700 hosts
- Three different ways of data normalization were used:
  - 1) normalizing features to (mean=0, std=1)
  - 2) normalizing data vectors to |x|=1
  - 3) the combination of 1) followed by 2)
- One classifier set has been built using Gaussian kernel SVMLight
  - The weight –j was set to 1/40
    - That was the ratio of spam to normal within the training set
- The training set was divided into three equal parts
  - The first one used for SVM training, the other one for kernel gamma parameter tuning, the third one for cross-evaluation
  - The best achieved F1 (for normalization 3) was 0.39 (R=0.6, P=0.29)

Classification by Pre-computed Features

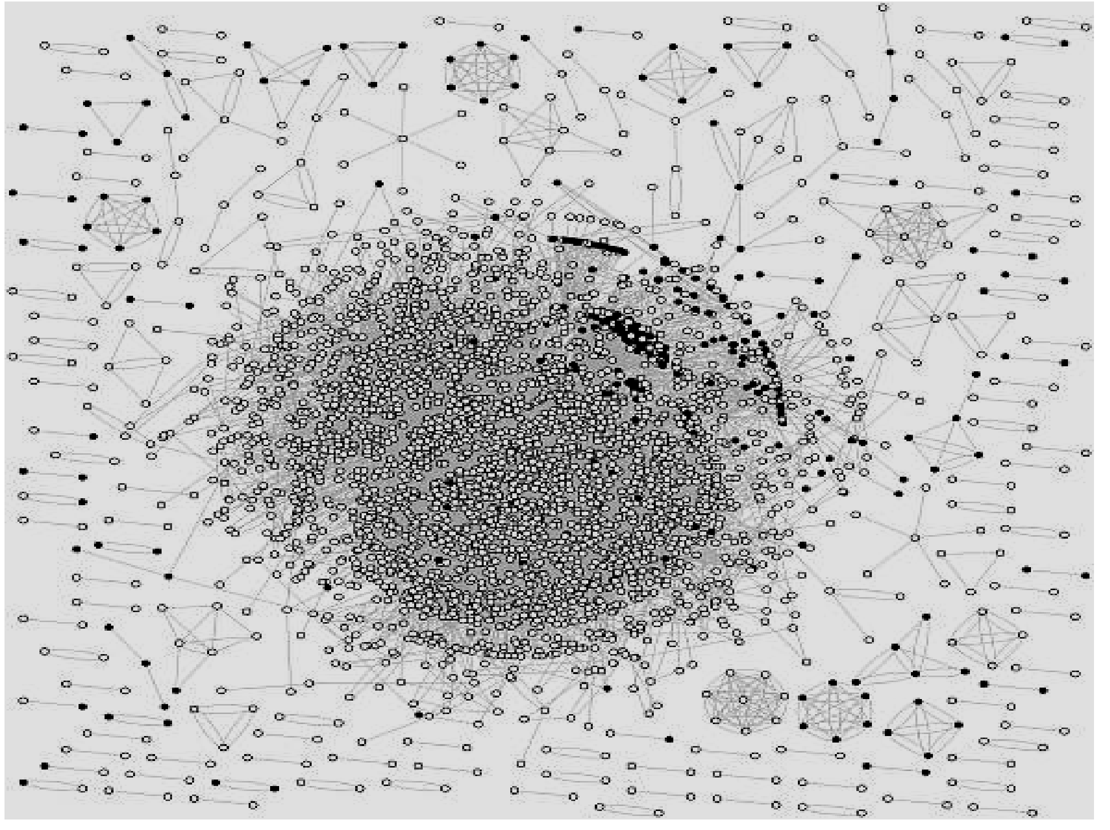
- The other classifier used weighted Linear kernel SVMLight
  - with feature selection
  - with weight of normal class = 0.2
- Feature selection
  - The features correlated at level >0.95 were considered connected
  - Then 276 features yielded 188 connected components
    - Of each connected component a single feature has been taken as a representative
    - The one most correlated with spam judgments is taken
- That set of 188 features achieved F1=22.4 (R=0.23, P=0.22)

Results with Pre-computed Features



- Max F1 = 24.66% (R = 28.46%, P = 21.75%)

Host Graph Structure (an Illustration)



Labeled graph nodes (connected by >100 links) are distributed on plane using “spring” model and the spam ends up together

("Know your Neighbors: Web Spam Detection using the Web Topology")

Host Graph Data

- The original training set:
  - Spam – 229 nodes
  - Normal – 3714 nodes
- The set, extended by last year's judgments:
  - Spam – 453 nodes
  - Normal – 13504 nodes

Scores Extension

- Additional scores taken
  - At least two judges gave the same score
  - Or, hosts that were in “trusted” domain:

    ac.uk, sch.uk, gov.uk, nhs.uk, police.co.uk

Normal Nodes Labeling

- The nodes classified a priori as normal:
  - Those judged as normal
  - Those linked by normal… – the idea was:
    - Spam refers to spam frequently, normal hosts don’t

Spam Labeling (1)

- The nodes classified a priori as spam:
  - Those judged as spam
  - Those linking spam… – the idea was:
    - Spam refers to spam frequently, normal hosts don’t

Spam Labeling (2)

- Two features computed for each node:
  - Overlap:
    - The ratio of bidirectional links to sum of in- and out-links
      - The idea is of link farms detection
  - Variance:
    - Standard deviation in number of out-links with in-neighbors
      - The idea: if it’s small, the graph might be automatically generated
- Thresholds for Overlap и Variance were learned
  - The node is classified as spam by either one

Scores Initialization

Node *x* is assigned a pair *(Bad Score(x) , Good Score(x))*:

- 1) If a node was marked normal, *Good Score(x) = 1*
- 2) If a node was marked spam, *Bad Score(x) = -1*

Make iterations on scores – and consider the pair:

- *(Old Bad Score(x), Old good score(x))*
  - the values on previous interation

Scores Propagation

Then next iteration scores are computed as:

$$\text{Bad Score}(x) = \text{Old Bad Score}(x) + \alpha^{i} \frac{\sum_{\text{y - in neighbor}} \text{Old Bad Score}(y)}{|\text{in neighbors}|}$$

$$\text{Good Score}(x) = \text{Old Good Score}(x) + \alpha^{\text{i}} \frac{\sum_{\text{y - out neighbor}} \text{Old Good Score}(y)}{|\text{out neighbors}|}$$

α is a free parameter here

- we set it to 0.2

The Final Node Classification

After fixed number of iterations we get final values of Bad Score и Good Score, and use it for classification:

$$\beta \times \text{Bad Score} + (1 - \beta) \times \text{Good Score} < 0 \Rightarrow \text{spam}$$

$$\beta \times \text{Bad Score} + (1 - \beta) \times \text{Good Score} > 0 \Rightarrow \text{normal}$$

The value of parameter β is set to = 0.95

Algorithm Parameters were Chosen

- Overlap
  - 0.5
- Threshold on link weight for spam labeling
  - 5000
- Threshold on variance
  - 0.05
- $\alpha = 0.2$
- $\beta = 0.95$

Choosing the Parameters

Two approaches were used:

1. The gradient optimization
2. The mesh search

Starting point:

1. Overlap = 0.1, step= 0.1
2. Link weight threshold = 4000, step = 1000
3. Variance threshold = 0.01, step = 0.01
4. α = 0.1, step = 0.1
5. β = 0.9, step = 0.01

The target functions in parameter choice

- Specificity =
  - Spam correctly classified / Total spam
- Sensitivity =
  - Normal correctly classified / Total normal
- The threshold on specificity was set
  - It was set to 0.4
    - Values in (0.4 – 0.6) were originally tried
  - The sensitivity was maximized

Results for Host Graph 2006

- Results with no cross-validation (on a training set)
  - F1 = 89.93% (R = 100%, P = 81.7%)

	Normal	Spam
Predicted Normal	4797	0
Predicted Spam	151	674

- Results with cross-validation (2-fold)
  - F1 = 52.8% (R = 50.08%, P = 55.83 %)

	Normal	Spam
Predicted Normal	4628	334
Predicted Spam	265	335

Results for Host Graph 2007

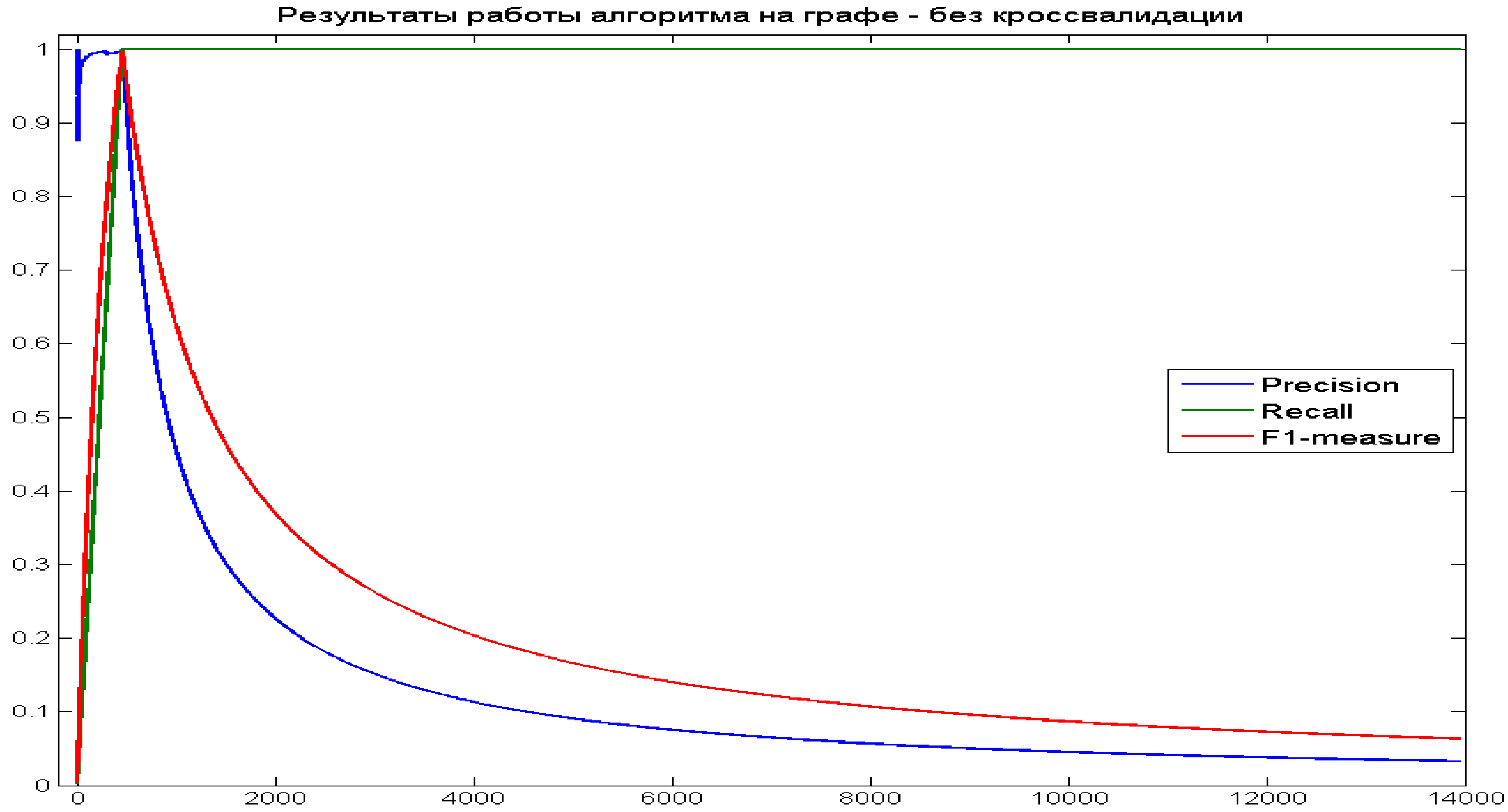
- Results with no cross-validation (on a training set)
  - F1 = 99.67% (R = 100%, P = 99.34%)

	Normal	Spam
Predicted Normal	13501	0
Predicted Spam	3	453

- Results with cross-validation (2-fold)
  - F1 = 21.47% (R = 30.82%, P = 16.47%)

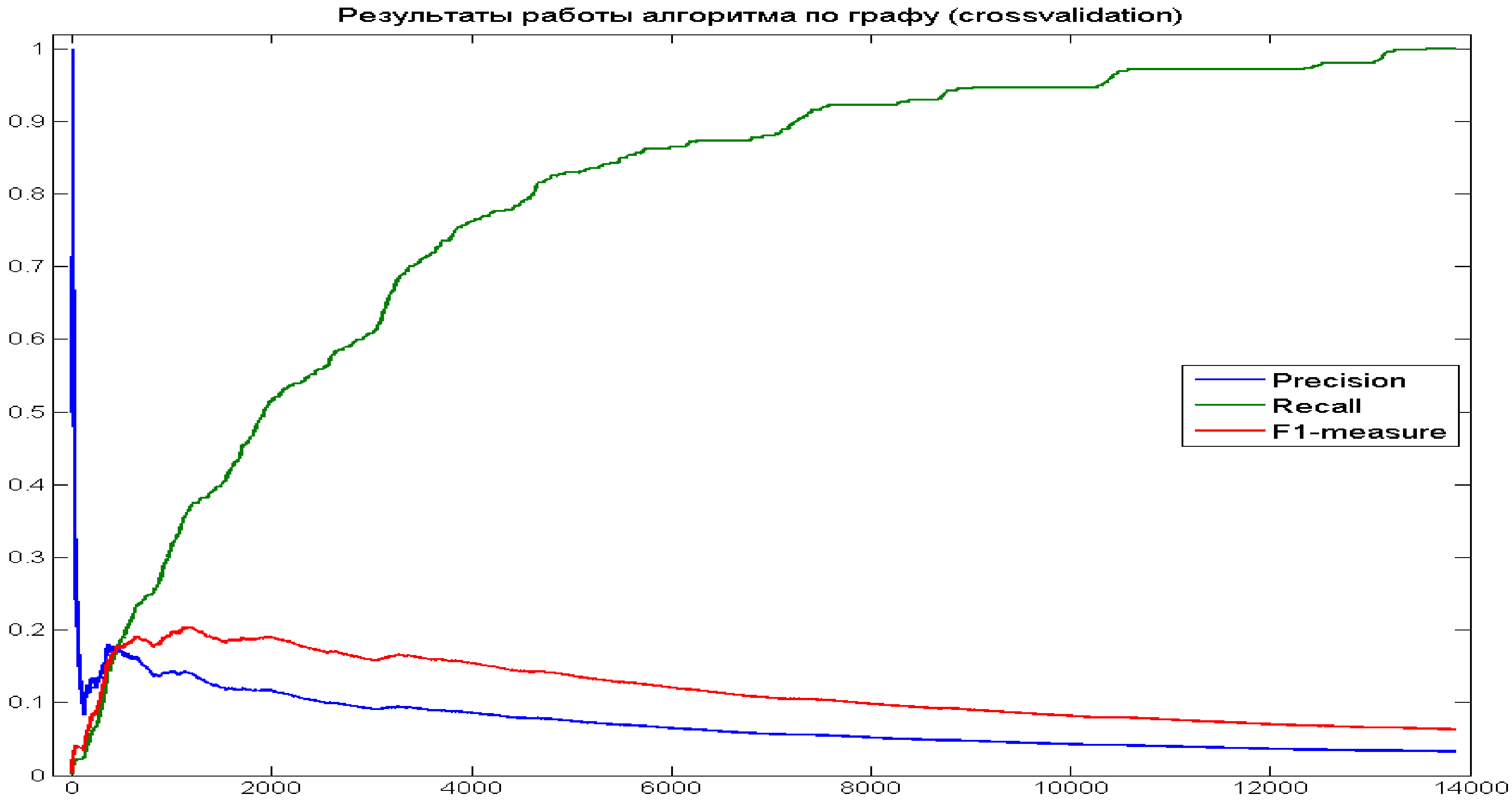
	Normal	Spam
Predicted Normal	12559	312
Predicted Spam	705	139

Results for Host Graph 2007 (no cross-validation)



- Max F1 = 99.67% (R = 100%, P = 99.34%)

Results for Host Graph 2007 (cross-validation)



- Max F1 = 21.47% (R = 30.82%, P = 16.47%)

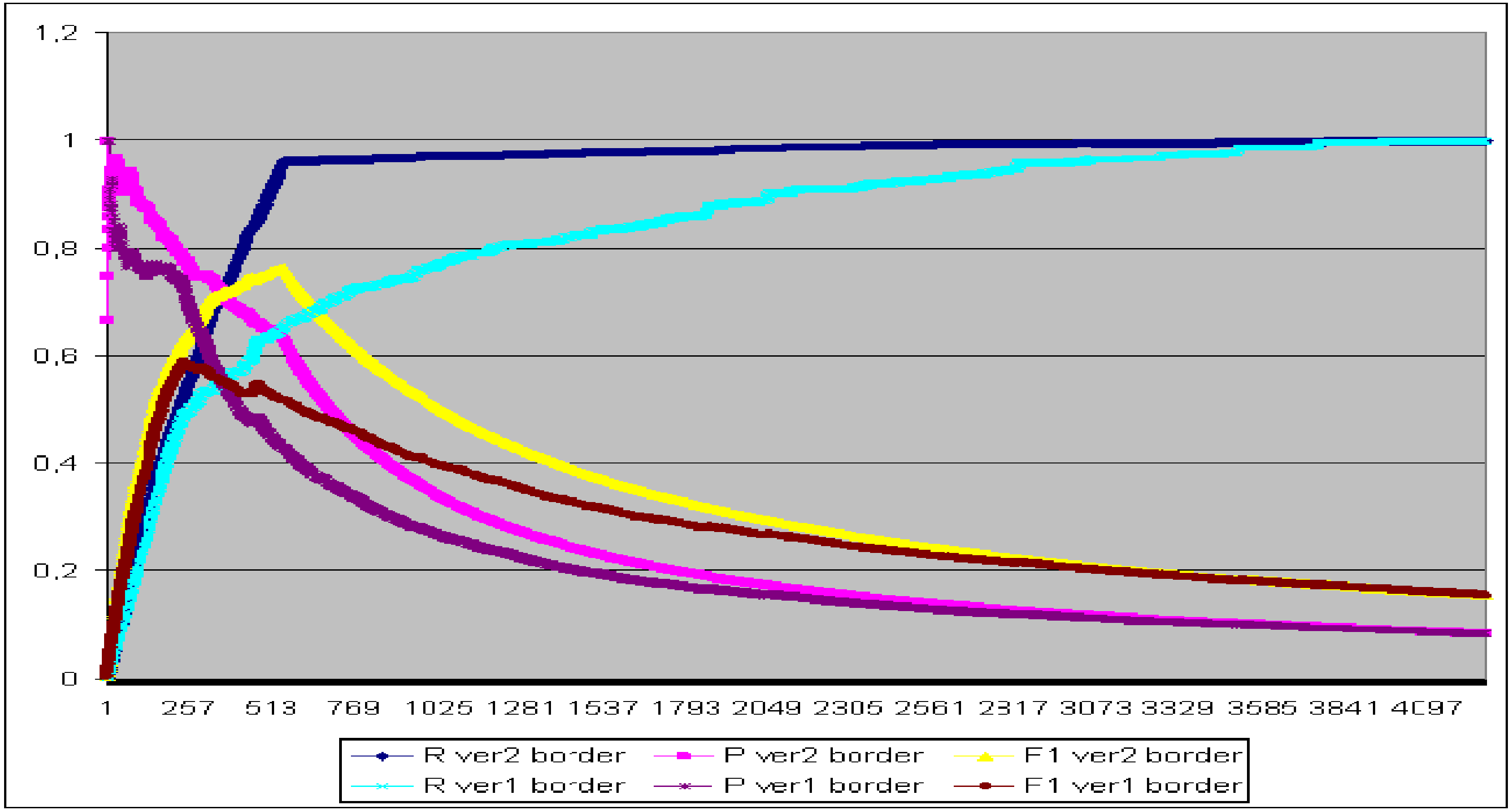
The Final Classifier Training

- The classifier was created by combining weak learners
  - Weak learners obtained by separate groups of features
- Combination was done with the TreeNet software
  - In classification mode, with unit weights
- The partial classifiers were created using:
  - SVM with Linear and Gaussian kernels, Naïve Bayes
    - four SVM and three Naïve Bayes classifiers were built on word frequencies
- There also was a direct graph-based rule
- Discriminant functions were weak learners for TreeNet model
- The F1 measure of stand-alone classifiers did not exceed 39%
  - The combined F1 for spam detection estimated as 67.5% (at R=68.3%, P=66.7%)

The Second Final Classifier Training

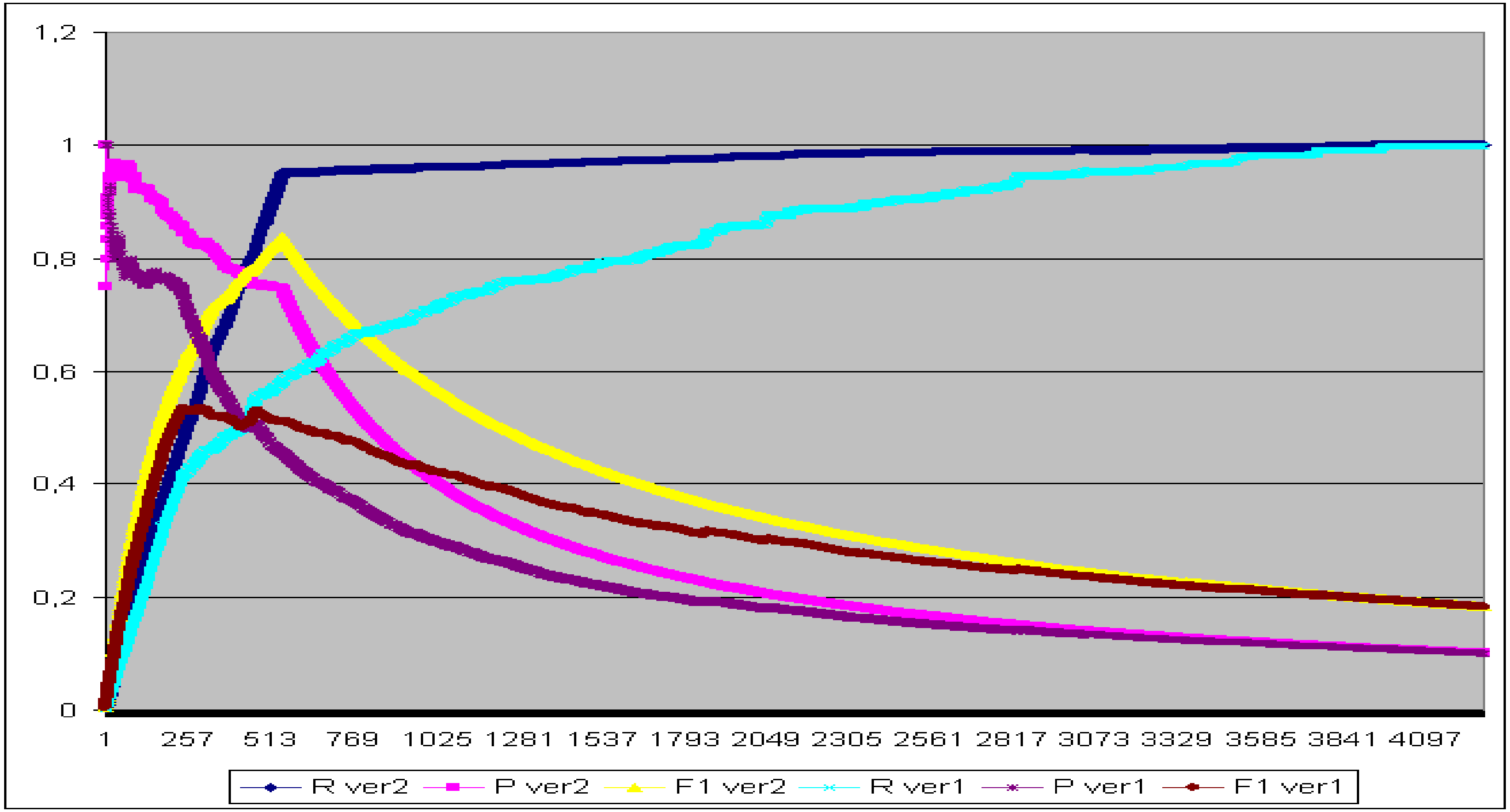
- With the first submission
  - the TreeNet classifier trained on overall spam judgments
    - Obtained with judgments by all judges taken together
- With the second submission
  - 34 separate classifiers were built for judgments of each judge
  - Judges that made more than 100 judgments (we took 34 of them)
  - For four judgment types (borderline, nonspam, spam, unknown)
- The probabilities of each class were computed for each judge
- Weighted sum of 34 probabilities for each of first three classes taken
  - The weights equal to (1-prob("unknown"))
- Then the final spam probability was calculated as
  - (s + 0.5*b)/(s + n + b)
    - Where s, n, b were weighted sums of computed probabilities from all judges
    - For the classes of "spam", "nonspam" and "borderline", respectively

“Borderline” as 0.5 Spam (training)



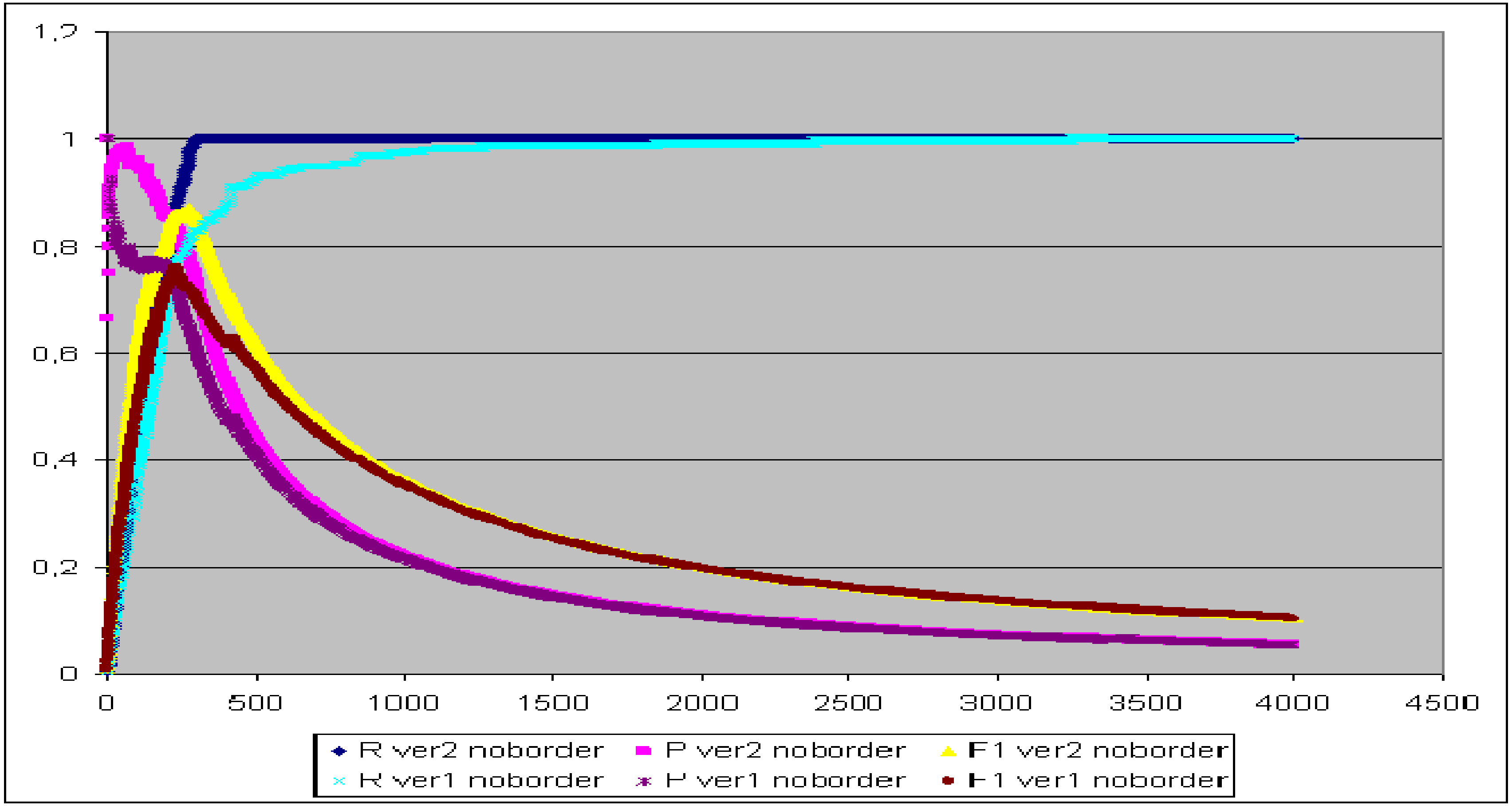
- First Version: Max F1 = 58.9% (R = 49.2%, P = 73.3%)
- 2nd Version: Max F1 = 76.3% (R = 96.3%, P = 63.3%)

"Borderline" as 0.75 Spam (training)



- First Version: Max F1 = 53.5% (R = 45%, P = 65.9%)
- 2nd Version: Max F1 = 83.8% (R = 95.3%, P = 74.8%)

“Borderline” Judgments Ignored (training)



- First Version: Max F1 = 76% (R = 77.9%, P = 74.2%)
- 2nd Version: Max F1 = 87.2% (R = 98.2%, P = 78.4%)

Acknowledgments

- The authors are grateful to several people who have made this work possible…
  - … and especially to:
    - Alexander Melkov
    - Alexei Pyalling
    - Sergey Pevtsov
  - … for their invaluable help and contributions