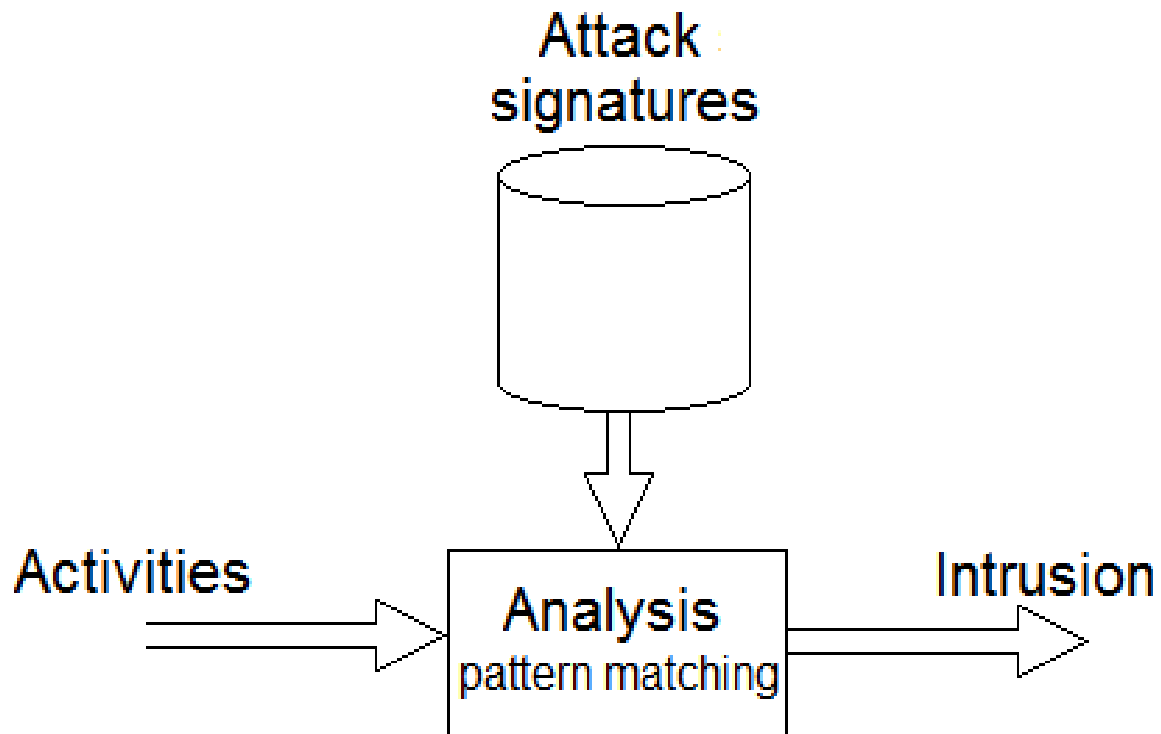


# Approximate search in misuse detection-based IDS by using the q-gram distance

Sverre Bakke

# Outline

- Topic
- Research questions
- q-gram distance
- Approximate search in IDS
- Experiments & results
- Conclusions



A typical misuse detection-based IDS

# Topic (cont.)

## Problem:

- Detects known attacks from a signature database
- Can only find exact matches
- Signature database takes time to search
- Fault-tolerant search can find unknown attacks
- Adding fault tolerant pattern matching adds complexity to the search
  
- Fault-tolerant search is slow!

# Topic (cont.)

- Previous work suggests that the q-gram distance may be used to speed up fault-tolerant document/Internet search
- We wanted to see if this could be applied to intrusion detection

# Research Questions

- How can the so-called q-gram distance be applied in approximate search for intrusion detection?
- How does the q-gram distance compare with other approximate pattern matching algorithms in terms of accuracy and performance?

# q-gram distance

- The q-gram distance is a (pseudo) metric for measuring the distance between two strings
- Can be used to determine if two strings matches each other with less than  $k$  errors.
- Counts occurrences of all the substrings of length  $q$  in two strings and find the difference in the occurrence count between the strings

# q-gram distance (cont.)

- A q-gram is a substring of length q within another string

Examples:

«textstring» contains the following 3-grams ( $q=3$ ):  
tex, ext, xts, tst, str, tri, rin, ing

«textstring» contains the following 2-grams ( $q=2$ ):  
te, ex, xt, ts, st, tr, ri, in, ng

«textstring» contains the following 1-grams ( $q=1$ ):  
t, e, x, t, s, t, r, i, n, g

# q-gram distance (cont.)

- A q-gram profile is a vector containing the occurrence count for all q-grams in a string

Example:

«textstring» contains the following 3-grams:  
[tex=1, ext=1, ... , ing=1]

# q-gram distance (cont.)

- A sliding window abstraction:

<code>this is a string</code>	<b>q-gram</b>	<b>occurrences</b>
<code>this is a string</code>	<code>_a_</code>	1
<code>this is a string</code>	<code>_is</code>	1
<code>this is a string</code>	<code>_st</code>	1
<code>this is a string</code>	<code>a_s</code>	1
<code>this is a string</code>	<code>his</code>	1
<code>this is a string</code>	<code>ing</code>	1
<code>this is a string</code>	<code>is_</code>	2
<code>this is a string</code>	<code>rin</code>	1
<code>this is a string</code>	<code>s_a</code>	1
<code>this is a string</code>	<code>s_i</code>	1
<code>this is a string</code>	<code>str</code>	1
<code>this is a string</code>	<code>thi</code>	1
<code>this is a string</code>	<code>tri</code>	1

# q-gram distance (cont.)

- The q-gram distance between two strings is the L1-distance between their q-gram profiles

$$\sum |x_i - y_i|$$

# q-gram distance (cont.)

## Advantages:

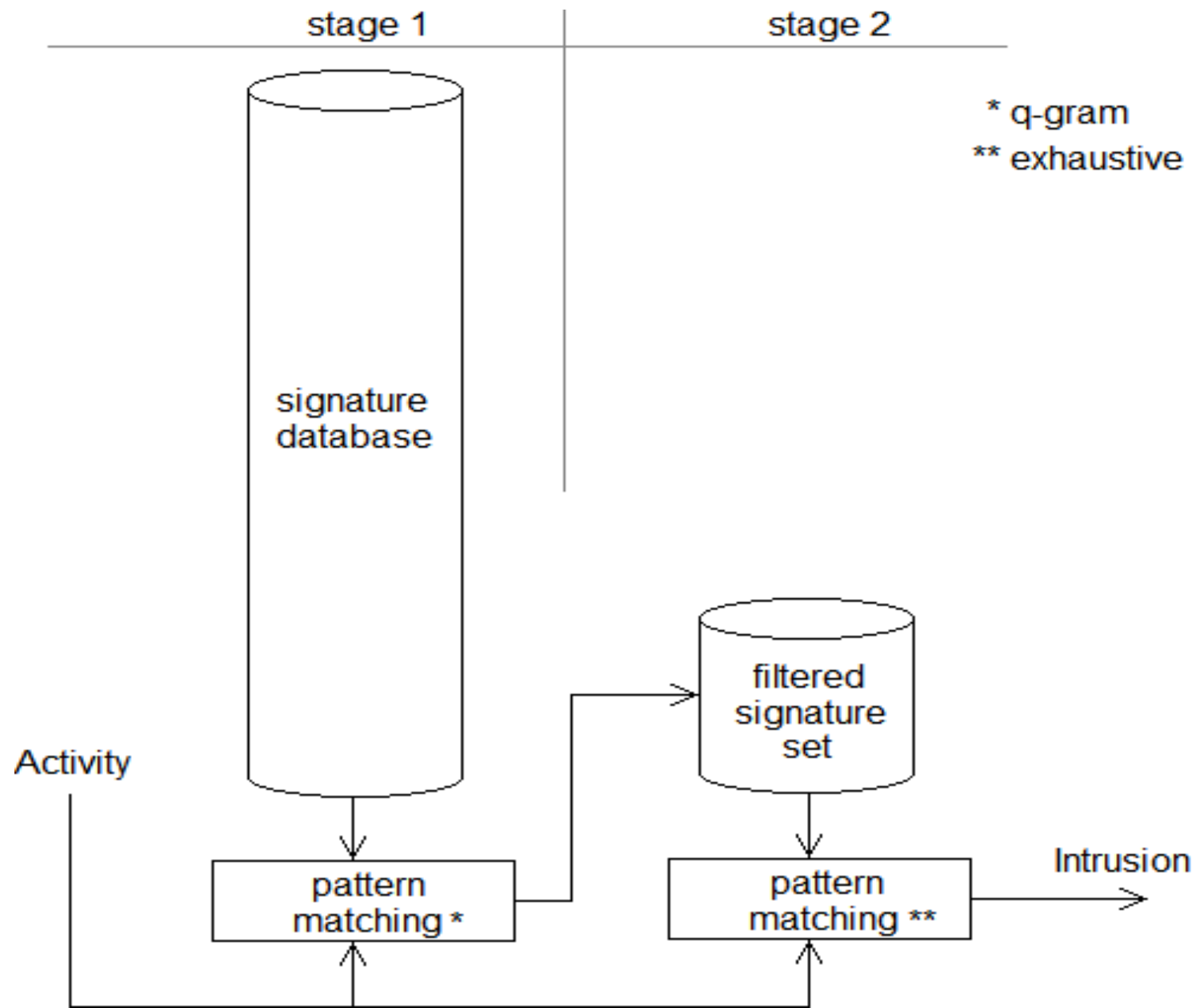
- Linear time complexity  $O(n+m)$ , not  $O(nm)$
- q-gram profiles can be computed at any time

## Disadvantages:

- Only a pseudo-metric
- Can not process strings shorter than length  $q$

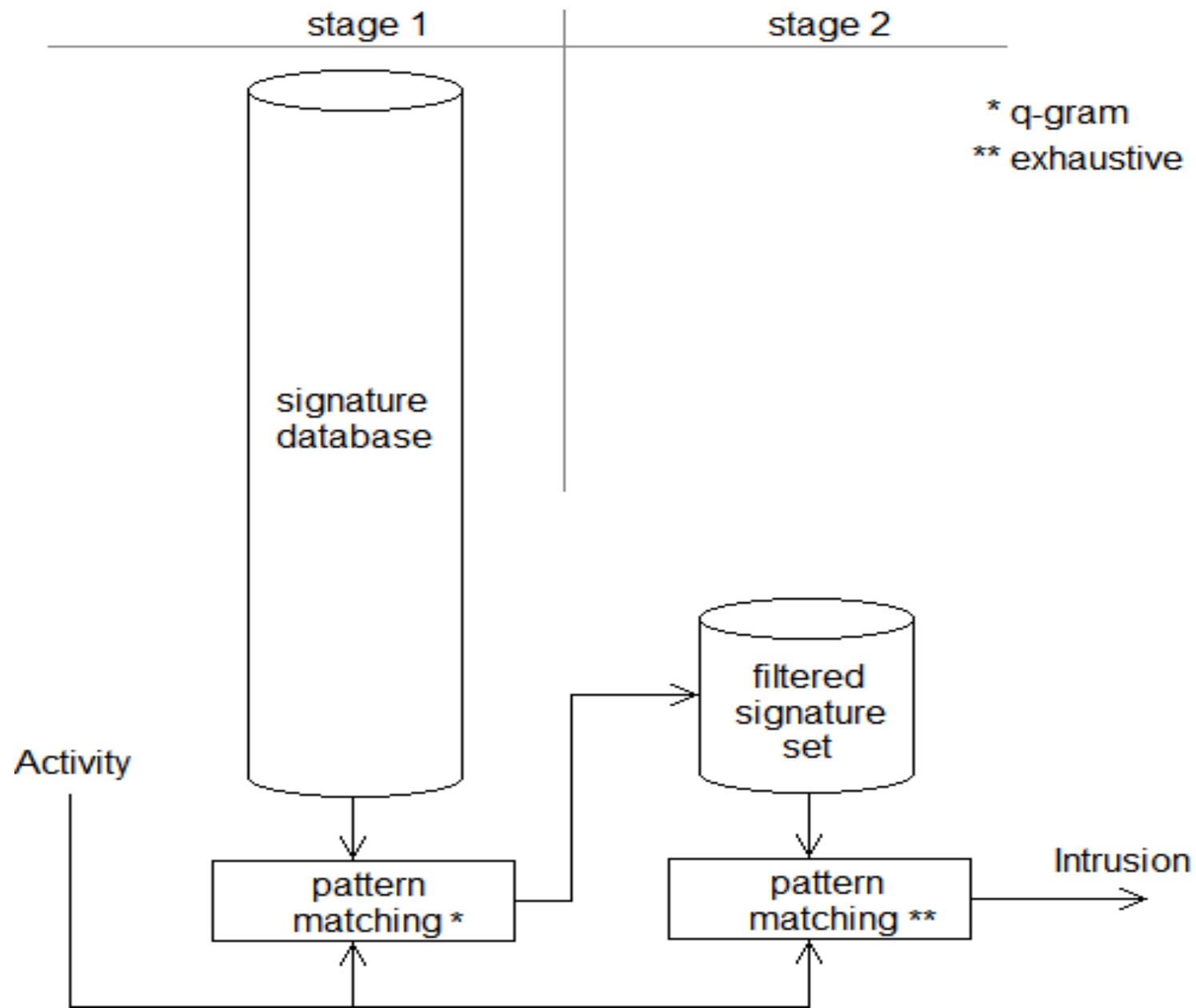
# Approximate Search

- We will use a two-stage search procedure
- q-gram distance used for filtering the dataset in the first stage
- Signatures will only be candidate for finer inspection in the second stage if the distance from the input is less than a given error threshold
- Exhaustive search algorithm is used in the second stage on a reduced dataset
- We focus on the first stage



# Experiments

- Implement the first stage (q-gram distance) and run test data through it
- Use padded SNORT rules (web-misc.rules) as signature database and input data
- More than 43 000 input/rule comparisons
- Look at data reduction, accuracy and performance
- Compare the q-gram distance with the edit distance and the constrained edit distance





# Experiments

Edit distance is the the minimal number of elementary edit operations (substitution, deletion, insertion) needed for transforming one string into another

# Experiments

The constrained edit distance is the edit distance under constraints:

- Maximum number of insertions
- Maximum length of runs of insertions and deletions
- Every substitution is preceded by at most one run of deletions followed by at most one run of insertions

# Experiments

We use the following parameters to the algorithms:

$$q = 1, 2, 3$$

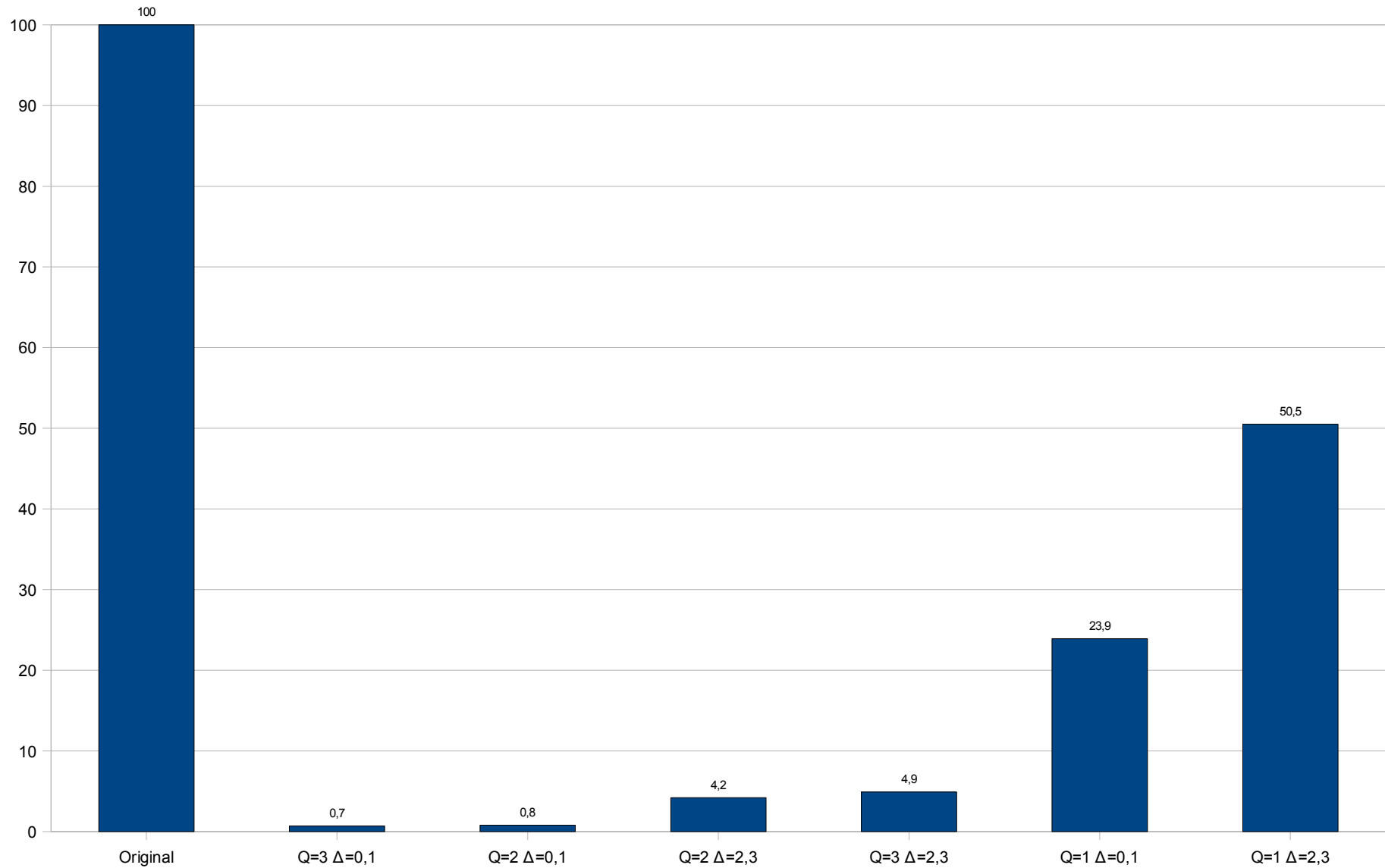
$$F = 1, 2, 3, 4, 5$$

$$\Delta = 0, 1, 2, 3$$

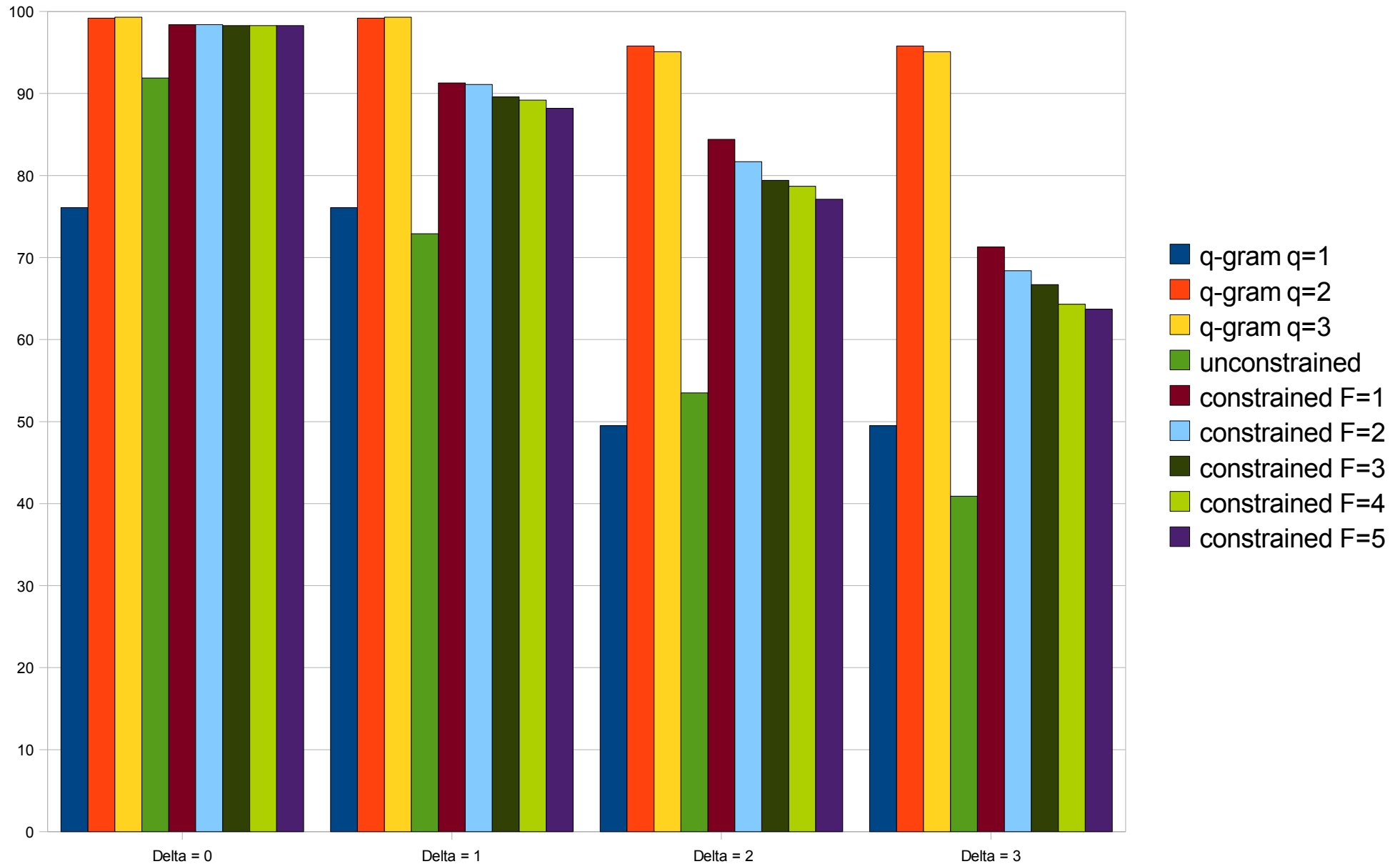
# Reduction Experiment

- See how much data we can remove from the second stage
- Compare each input with all rules
- Count the number of input/rule comparisons that is accepted by our pattern matching

# Reduction Experiment



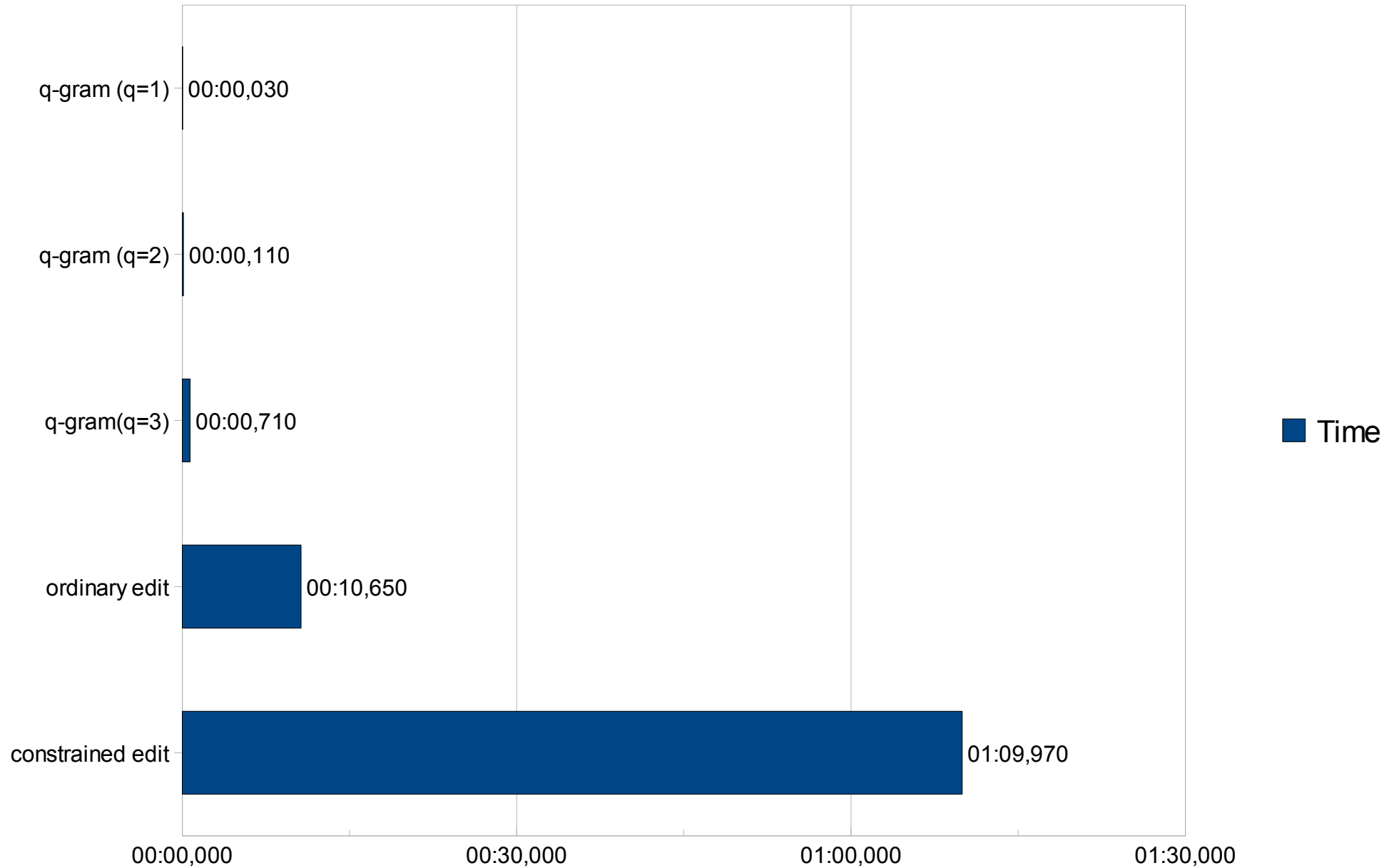
# Reduction Experiment



# Performance Experiment

- Compare the raw performance of the different distance algorithms in the first stage
- Measure the time each algorithm needs to compare all input data with all rules
- Repeat 20 times and use the average time

# Performance Experiment



# Accuracy Experiment

Compare the accuracy of the q-gram distance:

- against the ordinary edit distance
- against the constrained edit distance

The q-gram distance needs to «agree» with the other algorithm for it to be «correct»

Compare all combinations of  $q$ ,  $F$ ,  $\Delta$

Algorithms have their individual  $\Delta$  threshold

# Accuracy Experiment

## q-gram distance vs ordinary edit distance:

48 different combinations of the algorithms parameters

The best case is when they differ in only 6,6% of the input/rule comparisons

The worst case is when they differ in 57,7% of the input/rule comparisons

No apparent pattern in the results

This is not good results!!

#	q	$\Delta_q$	$\Delta_e$	$d_H$	%
1	3	2	0	2858	6,598
2	3	3	0	2858	6,598
3	2	0	0	2896	6,631
4	2	1	0	2896	6,631
5	3	0	0	2921	6,743
6	3	1	0	2921	6,743
7	2	2	0	3246	7,432
8	2	3	0	3246	7,432
9	1	0	0	6990	15,746
10	1	1	0	6990	15,746
11	1	0	1	7760	17,481
12	1	1	1	7760	17,481
13	3	2	1	9531	22,002
14	3	3	1	9531	22,002
15	1	2	3	9564	21,544
16	1	3	3	9564	21,544
17	1	2	2	9816	22,112
18	1	3	2	9816	22,112
19	2	2	1	10214	23,386
20	2	3	1	10214	23,386
21	1	2	1	10351	23,317
22	1	3	1	10351	23,317
23	2	0	1	10978	25,135
24	2	1	1	10978	25,135

# Accuracy Experiment

## q-gram distance vs constrained edit distance:

240 different combinations of the algorithms parameters

- The best case is when they differ in only 0,014% of the input/rule comparisons
- The worst case is when they differ in 48,9% of the input/rule comparisons ( $q=1$ )

The best results are when we use large q-grams and have a low threshold

The q-gram distance can estimate the constrained edit distance for:

- $\Delta_e = 0$  with no more than 0,014% errors
- $\Delta_e = 1$  with no more than 5% errors
- $\Delta_e = 2$  with no more than 8,8% errors
- $\Delta_e = 3$  with no more than 23,4% errors

#	q	$\Delta_q$	F	$\Delta_e$	$d_H$	%
1	3	0	1	0	6	0.014
2	3	1	1	0	6	0.014
3	3	0	2	0	26	0.060
4	3	1	2	0	26	0.060
5	2	0	1	0	34	0.078
6	2	1	1	0	34	0.078
7	3	0	3	0	51	0.118
8	3	1	3	0	51	0.118
9	2	0	2	0	52	0.119
10	2	1	2	0	52	0.119
11	3	0	4	0	65	0.150
12	3	1	4	0	65	0.150
13	2	0	3	0	67	0.153
14	2	1	3	0	67	0.153
15	2	0	4	0	79	0.181
16	2	1	4	0	79	0.181
17	3	0	5	0	90	0.208
18	3	1	5	0	90	0.208
19	2	0	5	0	102	0.234
20	2	1	5	0	102	0.234

# Accuracy Experiment

No algorithms rejected any data that would be a match when using exact search

# Conclusions

- Results indicate that the q-gram distance may be used in some cases for approximate search in IDS, but not a perfect solution for all cases
- Not very good for estimating the edit distance
- May be used to quickly estimate many cases of the constrained edit distance (for large q-grams and low threshold values)
- It does not scale very well with the threshold

Questions?