Approximate String Joins in a Database (Almost) for Free

Erratum

Luis Gravano
Columbia University
gravano@cs.columbia.edu

Panagiotis G. Ipeirotis
Columbia University
pirot@cs.columbia.edu

H. V. Jagadish
University of Michigan
jag@eecs.umich.edu

Nick Koudas
AT&T Labs–Research
koudas@research.att.com

S. Muthukrishnan
AT&T Labs–Research
muthu@research.att.com

Divesh Srivastava
AT&T Labs–Research
divesh@research.att.com

1 SQL Expression

In [GIJ+01a, GIJ+01b] we described how to use q-grams in an RDBMS to perform approximate string joins. We also showed how to implement the approximate join using plain SQL queries. Specifically, we described three filters, *count filter*, *position filter*, and *length filter*, which can be used to execute efficiently the approximate join. The intuition behind the *count filter* was that strings that are similar have many q-grams in common. In particular, two strings s_1 and s_2 can have up to $\max\{|s_1|, |s_2|\} + q - 1$ common q-grams. When $s_1 = s_2$, they have exactly that many q-grams in common. When s_1 and s_2 are within edit distance k, they share at least $(\max\{|s_1|, |s_2|\} + q - 1) - kq$ q-grams, since kq is the maximum number of q-grams that can be affected by k edit distance operations.

We implemented *count filter* in the HAVING clause of the SQL statement in Figure 1. String pairs without enough q-grams in common are filtered out from the result. Unfortunately, this implementation of the *count filter* is problematic when kq is greater than or equal to $\max\{|s_1|, |s_2|\} + q - 1$. In this case, two strings can be within edit distance k and still not share any q-grams. In such a case, the SQL statement in Figure 1 will fail to identify s_1 and s_2 as being within edit distance k, since there will be no q-grams from this string pair to join and count. Hence, in this case the result returned by the Figure 1 query is incomplete and suffers from "false negatives," in contrast to our claim to the contrary in [GIJ+01a, GIJ+01b].

In general, the string pairs that are omitted are pairs of short strings. Even when these strings match within small edit distance, the match tends to be meaningless (e.g., "IBM" matches "ACM" within edit distance 2). However, when it is absolutely necessary to have no false negatives, we can make the appropriate modifications to the SQL query in Figure 1 so that it produces the correct results. Since the false negatives are only pairs of short strings, we can join all pairs of these small strings, using only the *length filter*, and UNION the result with the result of the SQL query described in [GIJ+01a, GIJ+01b]. We list the modified query in Figure 2.

2 Experimental Results

We now experimentally measure the number of false negatives from which the query in [GIJ+01a, GIJ+01b] (Figure 1) can suffer. For the experiments we use the same three data sets that we used in [GIJ+01a]. To measure the number of false negatives, we focus on the differences between the Figure 1 and Figure 2 queries. First, we compute *NewPairs*, the number of tuples for which the *edit_distance* predicate is checked in Figure 2 but not in Figure 1. This number indicates the increase in the set of candidate string pairs with respect to Figure 1. Then, we measure the number of string pairs in *NewPairs* that are actual true positives (i.e., are within the given edit distance threshold k). This number, which we denote as *Missed*, is the number of false negatives *not* returned from the original SQL query in Figure 1 (i.e., *Missed* string pairs should have been included in the candidate set but were not). A large fraction of the string pairs in *Missed*,

```
SELECT     R1.A0, R2.A0, R1.Ai, R2.Aj
FROM       R1, R1AiQ, R2, R2AjQ
WHERE      R1.A0 = R1AiQ.A0 AND R2.A0 = R2AjQ.A0 AND
           R1AiQ.Qgram = R2AjQ.Qgram AND
           R1AiQ.Pos - R2AjQ.Pos ≤ k AND R2AjQ.Pos - R1AiQ.Pos ≤ k AND
           LEN(R1.Ai) - LEN(R2.Aj) ≤ k AND LEN(R2.Aj) - LEN(R1.Ai) ≤ k
GROUP BY   R1.A0, R2.A0, R1.Ai, R2.Aj
HAVING     COUNT(*) ≥ LEN(R1.Ai)+q-1 - k*q AND
           COUNT(*) ≥ LEN(R2.Aj)+q-1 - k*q AND
           edit_distance(R1.Ai, R2.Aj, k)
```

Figure 1: The SQL query as described in [GIJ+01a, GIJ+01b]. This SQL query might have some false negatives.

```
SELECT     R1.A0, R2.A0, R1.Ai, R2.Aj
FROM       R1, R1AiQ, R2, R2AjQ
WHERE      R1.A0 = R1AiQ.A0 AND R2.A0 = R2AjQ.A0 AND
           R1AiQ.Qgram = R2AjQ.Qgram AND
           R1AiQ.Pos - R2AjQ.Pos ≤ k AND R2AjQ.Pos - R1AiQ.Pos ≤ k AND
           LEN(R1.Ai) - LEN(R2.Aj) ≤ k AND LEN(R2.Aj) - LEN(R1.Ai) ≤ k AND
           (LEN(R1.Ai)+q-1 > k*q OR LEN(R2.Aj)+q-1 > k*q)
GROUP BY   R1.A0, R2.A0, R1.Ai, R2.Aj
HAVING     COUNT(*) ≥ LEN(R1.Ai)+q-1 - k*q AND
           COUNT(*) ≥ LEN(R2.Aj)+q-1 - k*q AND
           edit_distance(R1.Ai, R2.Aj, k)
UNION ALL
SELECT     R1.A0, R2.A0, R1.Ai, R2.Aj
FROM       R1, R2
WHERE      LEN(R1.Ai)+q-1 ≤ k*q AND
           LEN(R2.Aj)+q-1 ≤ k*q AND
           LEN(R1.Ai) - LEN(R2.Aj) ≤ k AND LEN(R2.Aj) - LEN(R1.Ai) ≤ k AND
           edit_distance(R1.Ai, R2.Aj, k)
```

Figure 2: The SQL query that has no false negatives.

however, are *trivial* matches, involving two short strings of length k or less, with edit distance equal to the length of the longer string. We denote as M_{Triv} the number of *Missed* string pairs that are *trivial* matches and as $M_{NonTriv}$ the number of *Missed* string pairs that are *non-trivial* matches.

The experimental results for the data sets R_1, R_2, and R_3 used in [GIJ+01a], are reported in Tables 1, 2, and 3, respectively. The column *Real* contains the number of real matches within the given edit distance threshold k, for each data set, and the column $Real_{NonTriv}$ contains the number of real matches within the given edit distance threshold k, excluding trivial matches. The column labeled $\frac{Missed}{NewPairs}$ shows the percentage of the new pairs (generated by the new sub-query in Figure 2 but without the *edit_distance* checks) that are actual true positives. When this percentage is high, then most of the *NewPairs* are real matches. When this percentage is low the *NewPairs* set contains many false positives, which means that we waste CPU time to filter out the false positives from the new candidate set. The $\frac{M_{NonTriv}}{NewPairs}$ value is the percentage of the *NewPairs* that are actual, non-trivial matches. We can observe that this number is rarely larger than 10% and never larger than 20%, supporting the hypothesis that a large percentage of the *NewPairs* are either string pairs that do not match within the given edit distance threshold, or are trivial matches.

The column titled $\frac{Missed}{Real}$ shows the percentage of the string pairs that the query of Figure 1 does not report as candidates although they are real matches, with respect to the total number of matches. For data set R_1 there are

```
SELECT     R1.A0, R2.A0, R1.Ai, R2.Aj
FROM       R1, R1AiQ, R2, R2AjQ
WHERE      R1.A0 = R1AiQ.A0 AND R2.A0 = R2AjQ.A0 AND
           R1AiQ.Qgram = R2AjQ.Qgram AND
           R1AiQ.Pos - R2AjQ.Pos ≤ k AND R2AjQ.Pos - R1AiQ.Pos ≤ k AND
           LEN(R1.Ai) - LEN(R2.Aj) ≤ k AND LEN(R2.Aj) - LEN(R1.Ai) ≤ k AND
           (LEN(R1.Ai)+q-1 > k*q OR LEN(R2.Aj)+q-1 > k*q)
GROUP BY   R1.A0, R2.A0, R1.Ai, R2.Aj
HAVING     COUNT(*) ≥ LEN(R1.Ai)+q-1 - k*q AND
           COUNT(*) ≥ LEN(R2.Aj)+q-1 - k*q AND
           edit_distance(R1.Ai, R2.Aj, k)
UNION ALL
SELECT     R1.A0, R2.A0, R1.Ai, R2.Aj
FROM       R1, R2
WHERE      LEN(R1.Ai)+q-1 ≤ k*q AND
           LEN(R2.Aj)+q-1 ≤ k*q AND
           LEN(R1.Ai) - LEN(R2.Aj) ≤ k AND LEN(R2.Aj) - LEN(R1.Ai) ≤ k AND
           edit_distance(R1.Ai, R2.Aj, k) AND
           (edit_distance(R1.Ai, R2.Aj, LEN(R1.Ai)-1) OR
            edit_distance(R1.Ai, R2.Aj, LEN(R2.Aj)-1))
```

Figure 3: A modification of the SQL query of Figure 1 that does not have false negatives and does not report "trivial" matches.

almost no false negatives for moderate values of k. For data sets R_2 and R_3, this percentage is substantial (from 24% to 86%). However, many false negatives are "trivial" matches (e.g., "SUN" and "IBM" within edit distance threshold $k = 3$). If we exclude the trivial matches from our calculation, we can see that the number of false negatives is smaller, especially for small edit distance thresholds; we report this percentage in the $\frac{M_{NonTriv}}{Real_{NonTriv}}$ column. For small edit distance thresholds ($k \leq 3$), for all of our data sets and all values of q that we tried, the percentage of false negatives does not exceed 31% of the real matches (excluding trivial matches from *Real*). For larger values of k, this percentage is substantial, indicating that the original query of Figure 1 has many false negatives for large values of k. In this case, the query of Figure 2 should be used instead of the query of Figure 1.

Finally, we should note that the query of Figure 2 reports back a large number of trivial matches. It is possible to avoid trivial matches altogether by adding the appropriate predicates in the SQL query. A modification of the SQL query of Figure 1 that does not have false negatives and does not report "trivial" matches is shown in Figure 3.

Acknowledgements

We thank Rafael Camps for pointing out the problem in the SQL query in Figure 1.

References

[GIJ+01a] Luis Gravano, Panagiotis G. Ipeirotis, H.V. Jagadish, Nick Koudas, S. Muthukrishnan, and Divesh Srivastava. Approximate string joins in a database (almost) for free. In *Proceedings of the 27th International Conference on Very Large Databases (VLDB 2001)*, pages 491–500, 2001.

[GIJ+01b] Luis Gravano, Panagiotis G. Ipeirotis, H.V. Jagadish, Nick Koudas, S. Muthukrishnan, Lauri Pietarinen, and Divesh Srivastava. Using q-grams in a DBMS for approximate string processing. *IEEE Data Engineering Bulletin*, 24(4):28–34, December 2001.

q	k	$Real$	$Real_{NonTriv}$	$NewPairs$	$Missed$	M_{Triv}	$M_{NonTriv}$	$\frac{M_{NonTriv}}{Missed}$	$\frac{Missed}{NewPairs}$	$\frac{M_{NonTriv}}{NewPairs}$	$\frac{Missed}{Real}$	$\frac{M_{NonTriv}}{Real_{NonTriv}}$
	1	3,795,398	3,795,398	0	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	2	4,132,308	4,132,308	0	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	3	4,505,872	4,505,870	2	2	2	0	0.00%	100.00%	0.00%	0.00%	0.00%
	4	4,871,552	4,871,546	6	6	6	0	0.00%	100.00%	0.00%	0.00%	0.00%
1	5	5,460,476	5,460,460	12	12	12	0	0.00%	100.00%	0.00%	0.00%	0.00%
	6	7,189,518	7,189,248	112	112	112	0	0.00%	100.00%	0.00%	0.00%	0.00%
	7	12,624,402	12,622,992	404	404	404	0	0.00%	100.00%	0.00%	0.00%	0.00%
	8	29,397,534	29,389,258	1,432	1,432	1,432	0	0.00%	100.00%	0.00%	0.00%	0.00%
	9	78,855,260	78,756,624	6,746	6,746	6,746	0	0.00%	100.00%	0.00%	0.01%	0.00%
	10	215,001,624	213,796,068	58,688	58,688	58,688	0	0.00%	100.00%	0.00%	0.03%	0.00%
	1	3,795,398	3,795,398	0	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	2	4,132,308	4,132,308	2	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	3	4,505,872	4,505,870	16	2	2	0	0.00%	12.50%	0.00%	0.00%	0.00%
	4	4,871,552	4,871,546	1,908	10	6	4	40.00%	0.52%	0.21%	0.00%	0.00%
2	5	5,460,476	5,460,460	172,426	186	16	170	91.40%	0.11%	0.10%	0.00%	0.00%
	6	7,189,518	7,189,248	10,779,410	2,344	266	2,078	88.65%	0.02%	0.02%	0.03%	0.03%
	7	12,624,402	12,622,992	278,069,300	40,896	1,358	39,538	96.68%	0.01%	0.01%	0.32%	0.31%
	8	29,397,534	29,389,258	561,750,494	589,640	7,836	581,804	98.67%	0.10%	0.10%	2.01%	1.98%
	9	78,855,260	78,756,624	634,950,404	5,714,562	86,472	5,628,090	98.49%	0.90%	0.89%	7.25%	7.15%
	10	215,001,624	213,796,068	660,206,070	34,629,408	1,017,022	33,612,386	97.06%	5.25%	5.09%	16.11%	15.72%
	1	3,795,398	3,795,398	0	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	2	4,132,308	4,132,308	4	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	3	4,505,872	4,505,870	2,006	2	2	0	0.00%	0.10%	0.00%	0.00%	0.00%
	4	4,871,552	4,871,546	2,872,362	22	6	16	72.73%	0.00%	0.00%	0.00%	0.00%
3	5	5,460,476	5,460,460	468,400,624	920	16	904	98.26%	0.00%	0.00%	0.02%	0.02%
	6	7,189,518	7,189,248	1,144,845,996	18,862	270	18,592	98.57%	0.00%	0.00%	0.26%	0.26%
	7	12,624,402	12,622,992	1,303,804,530	311,544	1,410	310,134	99.55%	0.02%	0.02%	2.47%	2.46%
	8	29,397,534	29,389,258	1,368,505,648	3,338,266	8,264	3,330,002	99.75%	0.24%	0.24%	11.36%	11.33%
	9	78,855,260	78,756,624	1,399,120,644	22,887,192	98,276	22,788,916	99.57%	1.64%	1.63%	29.02%	28.94%
	10	215,001,624	213,796,068	1,419,476,148	102,807,398	1,196,944	101,610,454	98.84%	7.24%	7.16%	47.82%	47.53%
	1	3,795,398	3,795,398	0	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	2	4,132,308	4,132,308	14	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	3	4,505,872	4,505,870	220,514	2	2	0	0.00%	0.00%	0.00%	0.00%	0.00%
	4	4,871,552	4,871,546	487,510,518	66	6	60	90.91%	0.00%	0.00%	0.00%	0.00%
4	5	5,460,476	5,460,460	1,267,623,922	2,598	16	2,582	99.38%	0.00%	0.00%	0.05%	0.05%
	6	7,189,518	7,189,248	1,391,143,162	48,902	270	48,632	99.45%	0.00%	0.00%	0.68%	0.68%
	7	12,624,402	12,622,992	1,440,780,396	605,412	1,410	604,002	99.77%	0.04%	0.04%	4.80%	4.78%
	8	29,397,534	29,389,258	1,476,107,110	5,202,198	8,276	5,193,922	99.84%	0.35%	0.35%	17.70%	17.67%
	9	78,855,260	78,756,624	1,504,242,258	29,991,916	98,620	29,893,296	99.67%	1.99%	1.99%	38.03%	37.96%
	10	215,001,624	213,796,068	1,527,976,430	121,562,098	1,205,272	120,356,826	99.01%	7.96%	7.88%	56.54%	56.30%
	1	3,795,398	3,795,398	0	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	2	4,132,308	4,132,308	316	0	0	0	0.00%	0.00%	0.00%	0.00%	0.00%
	3	4,505,872	4,505,870	15,752,622	6	2	4	66.67%	0.00%	0.00%	0.00%	0.00%
	4	4,871,552	4,871,546	1,171,758,984	158	6	152	96.20%	0.00%	0.00%	0.00%	0.00%
5	5	5,460,476	5,460,460	1,347,449,366	5,124	16	5,108	99.69%	0.00%	0.00%	0.09%	0.09%
	6	7,189,518	7,189,248	1,409,803,648	75,602	270	75,332	99.64%	0.01%	0.01%	1.05%	1.05%
	7	12,624,402	12,622,992	1,453,134,042	793,220	1,410	791,810	99.82%	0.05%	0.05%	6.28%	6.27%
	8	29,397,534	29,389,258	1,487,629,140	5,903,036	8,276	5,894,760	99.86%	0.40%	0.40%	20.08%	20.06%
	9	78,855,260	78,756,624	1,516,244,336	31,812,818	98,636	31,714,182	99.69%	2.10%	2.09%	40.34%	40.27%
	10	215,001,624	213,796,068	1,540,377,090	124,962,812	1,205,540	123,757,272	99.04%	8.11%	8.03%	58.12%	57.89%

Table 1: Experimental results for all the combinations of k and q, and for the data set R_1 used in [GIJ$^+$01a].

q	k	$Real$	$Real_{NonTriv}$	$NewPairs$	$Missed$	M_{Triv}	$M_{NonTriv}$	$\frac{M_{NonTriv}}{Missed}$	$\frac{Missed}{NewPairs}$	$\frac{M_{NonTriv}}{NewPairs}$	$\frac{Missed}{Real}$	$\frac{M_{NonTriv}}{Real_{NonTriv}}$
	1	173,576	131,576	42,000	42,000	42,000	0	0.00%	100.00%	0.00%	24.20%	0.00%
	2	323,194	187,084	132,940	132,940	132,940	0	0.00%	100.00%	0.00%	41.13%	0.00%
	3	593,220	306,654	271,358	271,358	271,358	0	0.00%	100.00%	0.00%	45.74%	0.00%
	4	991,530	505,648	441,078	441,078	441,078	0	0.00%	100.00%	0.00%	44.48%	0.00%
1	5	1,499,796	791,298	613,430	613,430	613,430	0	0.00%	100.00%	0.00%	40.90%	0.00%
	6	2,298,476	1,315,406	810,414	810,414	810,414	0	0.00%	100.00%	0.00%	35.26%	0.00%
	7	3,479,050	2,135,002	1,037,272	1,037,272	1,037,272	0	0.00%	100.00%	0.00%	29.81%	0.00%
	8	5,050,996	3,269,630	1,288,138	1,288,138	1,288,138	0	0.00%	100.00%	0.00%	25.50%	0.00%
	9	6,814,550	4,583,428	1,521,814	1,521,814	1,521,814	0	0.00%	100.00%	0.00%	22.33%	0.00%
	10	9,019,310	6,487,164	1,682,778	1,682,778	1,682,778	0	0.00%	100.00%	0.00%	18.66%	0.00%
	1	173,576	131,576	42,000	42,000	42,000	0	0.00%	100.00%	0.00%	24.20%	0.00%
	2	323,194	187,084	297,326	146,870	136,110	10,760	7.33%	49.40%	3.62%	45.44%	5.75%
	3	593,220	306,654	751,338	334,852	286,566	48,286	14.42%	44.57%	6.43%	56.45%	15.75%
	4	991,530	505,648	1,536,862	609,194	485,598	123,596	20.29%	39.64%	8.04%	61.44%	24.44%
2	5	1,499,796	791,298	2,936,618	965,704	706,926	258,778	26.80%	32.88%	8.81%	64.39%	32.70%
	6	2,298,476	1,315,406	4,412,444	1,496,008	977,780	518,228	34.64%	33.90%	11.74%	65.09%	39.40%
	7	3,479,050	2,135,002	6,210,292	2,267,094	1,331,182	935,912	41.28%	36.51%	15.07%	65.16%	43.84%
	8	5,050,996	3,269,630	8,576,844	3,287,400	1,750,054	1,537,346	46.76%	38.33%	17.92%	65.08%	47.02%
	9	6,814,550	4,583,428	11,955,790	4,296,296	2,177,488	2,118,808	49.32%	35.93%	17.72%	63.05%	46.23%
	10	9,019,310	6,487,164	14,829,080	5,386,648	2,459,642	2,927,006	54.34%	36.32%	19.74%	59.72%	45.12%
	1	173,576	131,576	42,000	42,000	42,000	0	0.00%	100.00%	0.00%	24.20%	0.00%
	2	323,194	187,084	460,288	148,330	136,110	12,220	8.24%	32.23%	2.65%	45.90%	6.53%
	3	593,220	306,654	1,370,898	341,066	286,566	54,500	15.98%	24.88%	3.98%	57.49%	17.77%
	4	991,530	505,648	3,347,916	628,946	485,882	143,064	22.75%	18.79%	4.27%	63.43%	28.29%
3	5	1,499,796	791,298	6,229,552	1,014,944	708,498	306,446	30.19%	16.29%	4.92%	67.67%	38.73%
	6	2,298,476	1,315,406	11,744,022	1,613,874	983,022	630,852	39.09%	13.74%	5.37%	70.21%	47.96%
	7	3,479,050	2,135,002	18,813,686	2,519,282	1,343,770	1,175,512	46.66%	13.39%	6.25%	72.41%	55.06%
	8	5,050,996	3,269,630	28,763,848	3,725,948	1,780,606	1,945,342	52.21%	12.95%	6.76%	73.77%	59.50%
	9	6,814,550	4,583,428	40,574,820	5,031,766	2,229,430	2,802,336	55.69%	12.40%	6.91%	73.84%	61.14%
	10	9,019,310	6,487,164	54,319,956	6,576,756	2,529,328	4,047,428	61.54%	12.11%	7.45%	72.92%	62.39%
	1	173,576	131,576	42,000	42,000	42,000	0	0.00%	100.00%	0.00%	24.20%	0.00%
	2	323,194	187,084	626,076	148,490	136,110	12,380	8.34%	23.72%	1.98%	45.94%	6.62%
	3	593,220	306,654	2,388,086	341,880	286,566	55,314	16.18%	14.32%	2.32%	57.63%	18.04%
	4	991,530	505,648	5,445,326	631,318	485,882	145,436	23.04%	11.59%	2.67%	63.67%	28.76%
4	5	1,499,796	791,298	12,312,764	1,021,152	708,498	312,654	30.62%	8.29%	2.54%	68.09%	39.51%
	6	2,298,476	1,315,406	23,233,962	1,626,760	983,070	643,690	39.57%	7.00%	2.77%	70.78%	48.93%
	7	3,479,050	2,135,002	37,411,076	2,545,468	1,344,048	1,201,420	47.20%	6.80%	3.21%	73.17%	56.27%
	8	5,050,996	3,269,630	56,344,454	3,781,478	1,781,312	2,000,166	52.89%	6.71%	3.55%	74.87%	61.17%
	9	6,814,550	4,583,428	76,249,236	5,134,234	2,230,986	2,903,248	56.55%	6.73%	3.81%	75.34%	63.34%
	10	9,019,310	6,487,164	97,345,300	6,754,750	2,531,918	4,222,832	62.52%	6.94%	4.34%	74.89%	65.10%
	1	173,576	131,576	42,000	42,000	42,000	0	0.00%	100.00%	0.00%	24.20%	0.00%
	2	323,194	187,084	814,292	148,546	136,110	12,436	8.37%	18.24%	1.53%	45.96%	6.65%
	3	593,220	306,654	3,367,170	342,092	286,566	55,526	16.23%	10.16%	1.65%	57.67%	18.11%
	4	991,530	505,648	9,181,906	632,416	485,882	146,534	23.17%	6.89%	1.60%	63.78%	28.98%
5	5	1,499,796	791,298	20,818,238	1,023,810	708,498	315,312	30.80%	4.92%	1.51%	68.26%	39.85%
	6	2,298,476	1,315,406	37,877,128	1,631,530	983,070	648,460	39.75%	4.31%	1.71%	70.98%	49.30%
	7	3,479,050	2,135,002	59,342,300	2,554,194	1,344,048	1,210,146	47.38%	4.30%	2.04%	73.42%	56.68%
	8	5,050,996	3,269,630	83,019,146	3,799,738	1,781,366	2,018,372	53.12%	4.58%	2.43%	75.23%	61.73%
	9	6,814,550	4,583,428	106,674,884	5,165,518	2,231,122	2,934,396	56.81%	4.84%	2.75%	75.80%	64.02%
	10	9,019,310	6,487,164	128,942,148	6,809,926	2,532,140	4,277,786	62.82%	5.28%	3.32%	75.50%	65.94%

Table 2: Experimental results for all the combinations of k and q, and for the data set R_2 used in [GIJ$^+$01a].

q	k	$Real$	$Real_{NonTriv}$	$NewPairs$	$Missed$	M_{Triv}	$M_{NonTriv}$	$\frac{M_{NonTriv}}{Missed}$	$\frac{Missed}{NewPairs}$	$\frac{M_{NonTriv}}{NewPairs}$	$\frac{Missed}{Real}$	$\frac{M_{NonTriv}}{Real_{NonTriv}}$
	1	318,412	145,930	172,482	172,482	172,482	0	0.00%	100.00%	0.00%	54.17%	0.00%
	2	839,294	273,426	556,420	556,420	556,420	0	0.00%	100.00%	0.00%	66.30%	0.00%
	3	1,746,038	548,196	1,149,188	1,149,188	1,149,188	0	0.00%	100.00%	0.00%	65.82%	0.00%
	4	3,086,422	1,023,992	1,911,676	1,911,676	1,911,676	0	0.00%	100.00%	0.00%	61.94%	0.00%
1	5	4,830,528	1,775,476	2,718,060	2,718,060	2,718,060	0	0.00%	100.00%	0.00%	56.27%	0.00%
	6	7,339,750	3,049,472	3,653,118	3,653,118	3,653,118	0	0.00%	100.00%	0.00%	49.77%	0.00%
	7	10,706,196	4,885,022	4,698,766	4,698,766	4,698,766	0	0.00%	100.00%	0.00%	43.89%	0.00%
	8	14,915,558	7,413,576	5,800,648	5,800,648	5,800,648	0	0.00%	100.00%	0.00%	38.89%	0.00%
	9	19,643,000	10,496,124	6,753,400	6,753,400	6,753,400	0	0.00%	100.00%	0.00%	34.38%	0.00%
	10	25,125,768	14,388,044	7,672,738	7,672,738	7,672,738	0	0.00%	100.00%	0.00%	30.54%	0.00%
	1	318,412	145,930	172,482	172,482	172,482	0	0.00%	100.00%	0.00%	54.17%	0.00%
	2	839,294	273,426	1,231,646	599,672	565,868	33,804	5.64%	48.69%	2.74%	71.45%	12.36%
	3	1,746,038	548,196	3,126,906	1,350,216	1,197,842	152,374	11.29%	43.18%	4.87%	77.33%	27.80%
	4	3,086,422	1,023,992	6,272,900	2,456,124	2,061,764	394,360	16.06%	39.15%	6.29%	79.58%	38.51%
2	5	4,830,528	1,775,476	10,900,832	3,883,228	3,050,572	832,656	21.44%	35.62%	7.64%	80.39%	46.90%
	6	7,339,750	3,049,472	16,515,200	5,888,154	4,276,582	1,611,572	27.37%	35.65%	9.76%	80.22%	52.85%
	7	10,706,196	4,885,022	23,049,308	8,539,646	5,790,568	2,749,078	32.19%	37.05%	11.93%	79.76%	56.28%
	8	14,915,558	7,413,576	30,375,440	11,809,908	7,440,204	4,369,704	37.00%	38.88%	14.39%	79.18%	58.94%
	9	19,643,000	10,496,124	38,253,780	15,301,746	9,039,830	6,261,916	40.92%	40.00%	16.37%	77.90%	59.66%
	10	25,125,768	14,388,044	46,261,714	19,215,912	10,584,566	8,631,346	44.92%	41.54%	18.66%	76.48%	59.99%
	1	318,412	145,930	172,482	172,482	172,482	0	0.00%	100.00%	0.00%	54.17%	0.00%
	2	839,294	273,426	1,906,738	603,204	565,868	37,336	6.19%	31.64%	1.96%	71.87%	13.65%
	3	1,746,038	548,196	5,510,940	1,365,232	1,197,842	167,390	12.26%	24.77%	3.04%	78.19%	30.53%
	4	3,086,422	1,023,992	11,969,684	2,503,204	2,062,430	440,774	17.61%	20.91%	3.68%	81.10%	43.04%
3	5	4,830,528	1,775,476	20,849,418	3,996,560	3,055,052	941,508	23.56%	19.17%	4.52%	82.74%	53.03%
	6	7,339,750	3,049,472	31,825,658	6,135,084	4,290,182	1,844,902	30.07%	19.28%	5.80%	83.59%	60.50%
	7	10,706,196	4,885,022	44,377,782	9,028,002	5,820,656	3,207,346	35.53%	20.34%	7.23%	84.33%	65.66%
	8	14,915,558	7,413,576	58,428,208	12,646,180	7,500,938	5,145,242	40.69%	21.64%	8.81%	84.79%	69.40%
	9	19,643,000	10,496,124	74,166,464	16,668,518	9,144,752	7,523,766	45.14%	22.47%	10.14%	84.86%	71.68%
	10	25,125,768	14,388,044	91,561,364	21,313,836	10,733,672	10,580,164	49.64%	23.28%	11.56%	84.83%	73.53%
	1	318,412	145,930	172,482	172,482	172,482	0	0.00%	100.00%	0.00%	54.17%	0.00%
	2	839,294	273,426	2,589,824	603,538	565,868	37,670	6.24%	23.30%	1.45%	71.91%	13.78%
	3	1,746,038	548,196	8,461,780	1,366,886	1,197,842	169,044	12.37%	16.15%	2.00%	78.29%	30.84%
	4	3,086,422	1,023,992	18,021,920	2,507,834	2,062,430	445,404	17.76%	13.92%	2.47%	81.25%	43.50%
4	5	4,830,528	1,775,476	30,715,838	4,007,488	3,055,052	952,436	23.77%	13.05%	3.10%	82.96%	53.64%
	6	7,339,750	3,049,472	45,733,078	6,156,078	4,290,278	1,865,800	30.31%	13.46%	4.08%	83.87%	61.18%
	7	10,706,196	4,885,022	63,416,166	9,070,164	5,821,174	3,248,990	35.82%	14.30%	5.12%	84.72%	66.51%
	8	14,915,558	7,413,576	83,958,582	12,731,644	7,501,922	5,229,722	41.08%	15.16%	6.23%	85.36%	70.54%
	9	19,643,000	10,496,124	107,514,520	16,818,032	9,146,744	7,671,288	45.61%	15.64%	7.14%	85.62%	73.09%
	10	25,125,768	14,388,044	133,415,504	21,550,282	10,737,420	10,812,862	50.18%	16.15%	8.10%	85.77%	75.15%
	1	318,412	145,930	172,482	172,482	172,482	0	0.00%	100.00%	0.00%	54.17%	0.00%
	2	839,294	273,426	3,362,746	603,650	565,868	37,782	6.26%	17.95%	1.12%	71.92%	13.82%
	3	1,746,038	548,196	11,577,860	1,367,286	1,197,842	169,444	12.39%	11.81%	1.46%	78.31%	30.91%
	4	3,086,422	1,023,992	24,058,376	2,509,734	2,062,430	447,304	17.82%	10.43%	1.86%	81.32%	43.68%
5	5	4,830,528	1,775,476	39,919,958	4,011,442	3,055,052	956,390	23.84%	10.05%	2.40%	83.04%	53.87%
	6	7,339,750	3,049,472	59,260,650	6,163,286	4,290,278	1,873,008	30.39%	10.40%	3.16%	83.97%	61.42%
	7	10,706,196	4,885,022	82,603,916	9,083,098	5,821,174	3,261,924	35.91%	11.00%	3.95%	84.84%	66.77%
	8	14,915,558	7,413,576	110,015,936	12,756,214	7,501,982	5,254,232	41.19%	11.59%	4.78%	85.52%	70.87%
	9	19,643,000	10,496,124	140,903,946	16,857,876	9,146,876	7,711,000	45.74%	11.96%	5.47%	85.82%	73.47%
	10	25,125,768	14,388,044	174,977,956	21,613,670	10,737,712	10,875,958	50.32%	12.35%	6.22%	86.02%	75.59%

Table 3: Experimental results for all the combinations of k and q, and for the data set R_3 used in [GIJ$^+$01a].