Ji-zhou Luo, Sheng-fei Shi, Hong-zhi Wang, Jian-zhong Li, 2017. FrepJoin: an efficient partition-based algorithm for edit similarity join. *Frontiers of Information Technology & Electronic Engineering*, **18**(10):1499-1510. http://dx.doi.org/10.1631/FITEE.1601347

FrepJoin: an efficient partition-based algorithm for edit similarity join

Key words: String similarity join; Edit distance; Filter and refine; Data partition; Combined frequency vectors

Corresponding author: Ji-zhou Luo
E-mail: luojizhou@hit.edu.cn
ORCID: http://orcid.org/ 0000-0002-3302-3917

Motivation

- String similarity join with edit distance constraints is crucial in manly applications, where near duplicate tuples need be found.
- Existing filter-and-refine algorithms have 3 defects below.
  - ➢generate candidate pairs by enumerating string pairs and cannot catch the dissimilarity between string subsets
  - ➢The signatures of each string are mainly local structures which cannot catch the dissimilarity of strings from a global perspective of view.
  - ➢usually generate a huge amount of candidate pairs
- To address these issues, a novel partition-based algorithm is developed to evaluate edit similarity join by using global information to enumerate a smaller candidate set in a more efficient way by partitioning the dataset into small chunks.

Main idea

- The frequency vectors of strings catch the dissimilarity between strings, and can be exploited in string similarity join.
- In one hand, a new filter is proposed to leverage the statistics to avoid computing edit distances for a noticeable proportion of candidate pairs which survive the existing filters.
- In the other hand, the frequency vectors are used to partition datasets into data chunks with dissimilarity between them being caught easily. An algorithm is designed to accelerate SSJ via the partitioned data.

Method

1. Bias distance on frequency vectors are defined. And we show it is a distance, based on which a new independent filter is designed.
2. The whole alphabet is combined into θ combined characters, and the frequency range of each combined character is split into small intervals. Thus, the strings fall into chunks which are identified by corresponding intervals.
3. The lower bounds of edit distances between any different chunk are estimated with close formulas, which is used to prune out string pairs without enumerating them.

Major results

- The new filter, which is based on frequency vectors, is independent to the existing ones

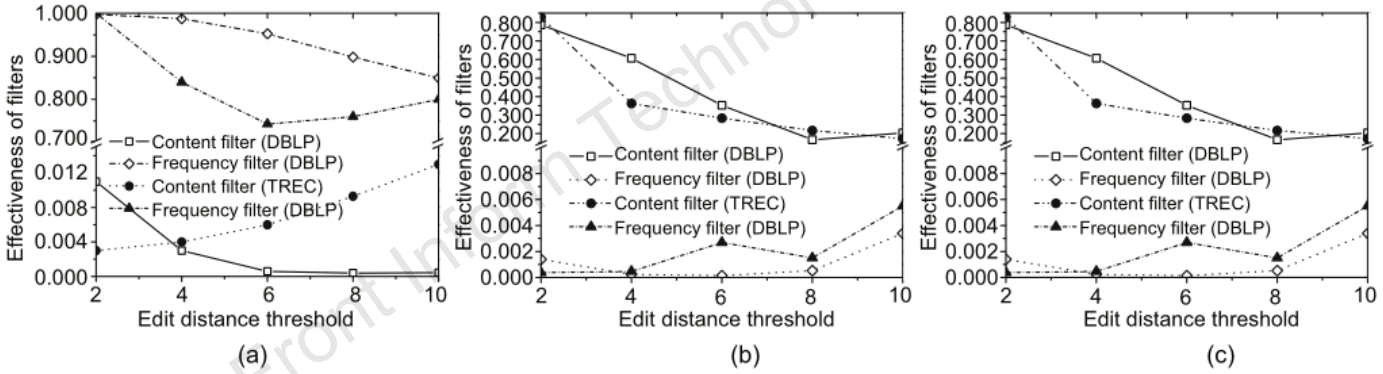

Fig. 1 Independences of frequency filters Ed-Join+FF (a), FF+Ed-Join (b), and Ed-Join(c), with $q = 5$ and $\theta = 3$

Major results (Cont'd)

- Data partition based on the frequency vectors can be done effectively and efficiently

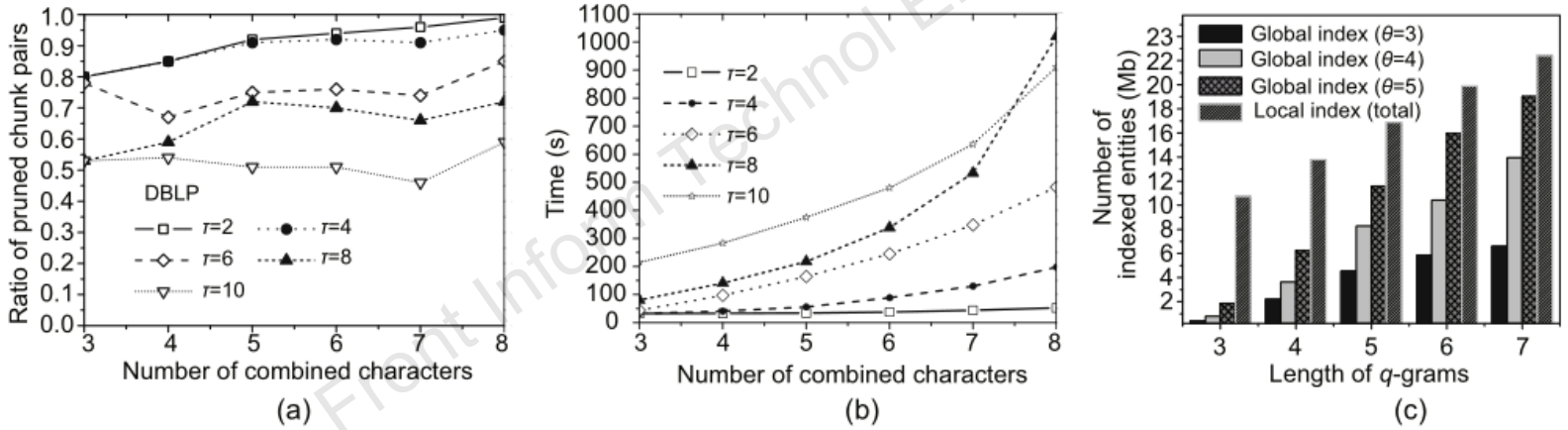

Fig. 2 Performance evaluation of data partitioning: (a) filtering effectiveness in DBLP; (b) time vs. number of characters in DBLP with $q = 5$; (c) number of indexed entities with $\tau = 6$

Major results (Cont'd)

- The new join algorithm runs quiet fast than the existing ones

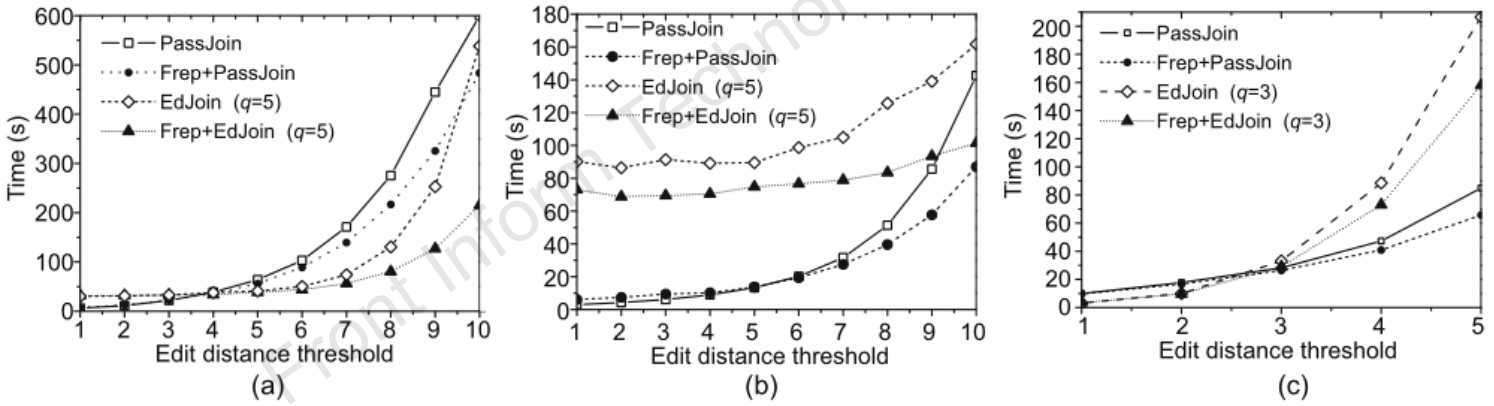

Fig. 3 Performance evaluation of FrepJoin in databases DBLP (a), TREC (b), and AOL Query Log (c) compared with those of Ed-Join and PassJoin

Conclusions

- Frequency vectors of strings can be exploited to improve the efficiency of edit string similarity join.
- Frequency vectors can be used to design a new filter ,which is independent to existing filters.
- Frequency vectors can be used to partition a dataset into data chunks with guaranteed distances, so that a remarkable proportion of candidate pairs can be pruned away without paying to enumerate them
- Experiments on real datasets show the independence of the new filter and the efficiency of the new algorithm.