

Capturing Relationships Based on Structure Similarity for Self-describing Scientific Data Formats

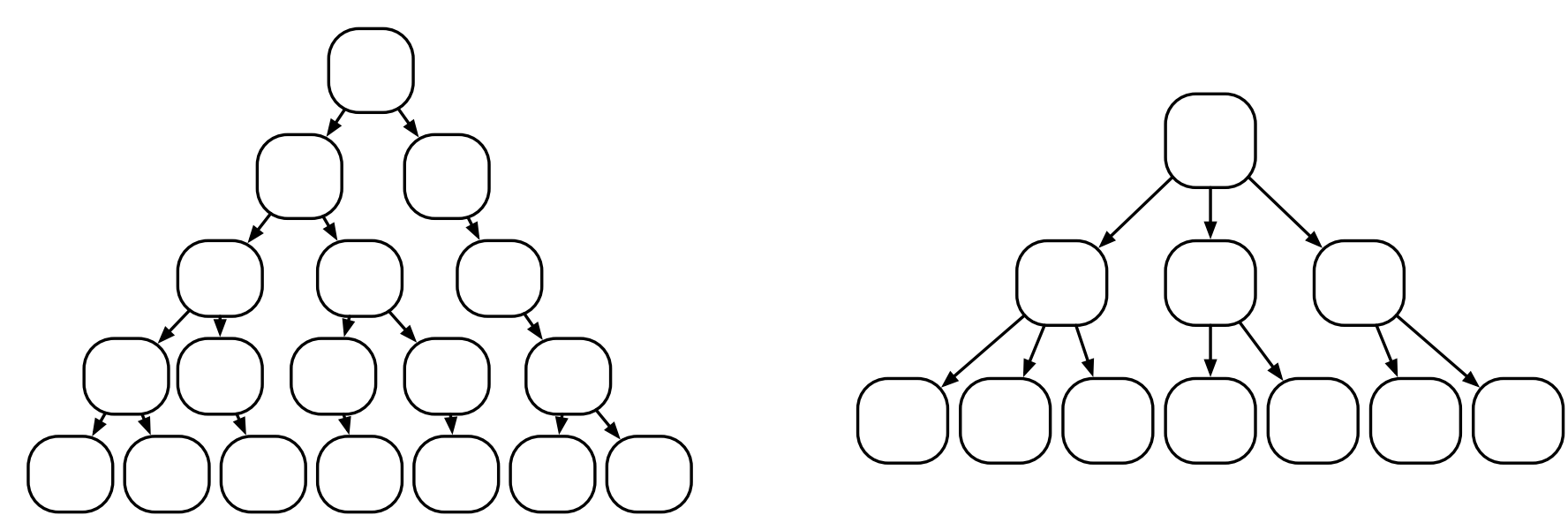
Chenxu Niu[†], Wei Zhang[†], Suren Byna⁺, Yong Chen[†]
[†] Texas Tech University, ⁺ Lawrence Berkeley National Laboratory

Introduction

- Many self-describing files are typically treated in **isolated** manner.
- There is no internal metadata related to other files.
- It is hard to capture relationships among scientific files.
- It is difficult to apply current linked-data principles in the context of modern scientific files.
- There is no standard specification outlining concrete relationships for self-describing files.

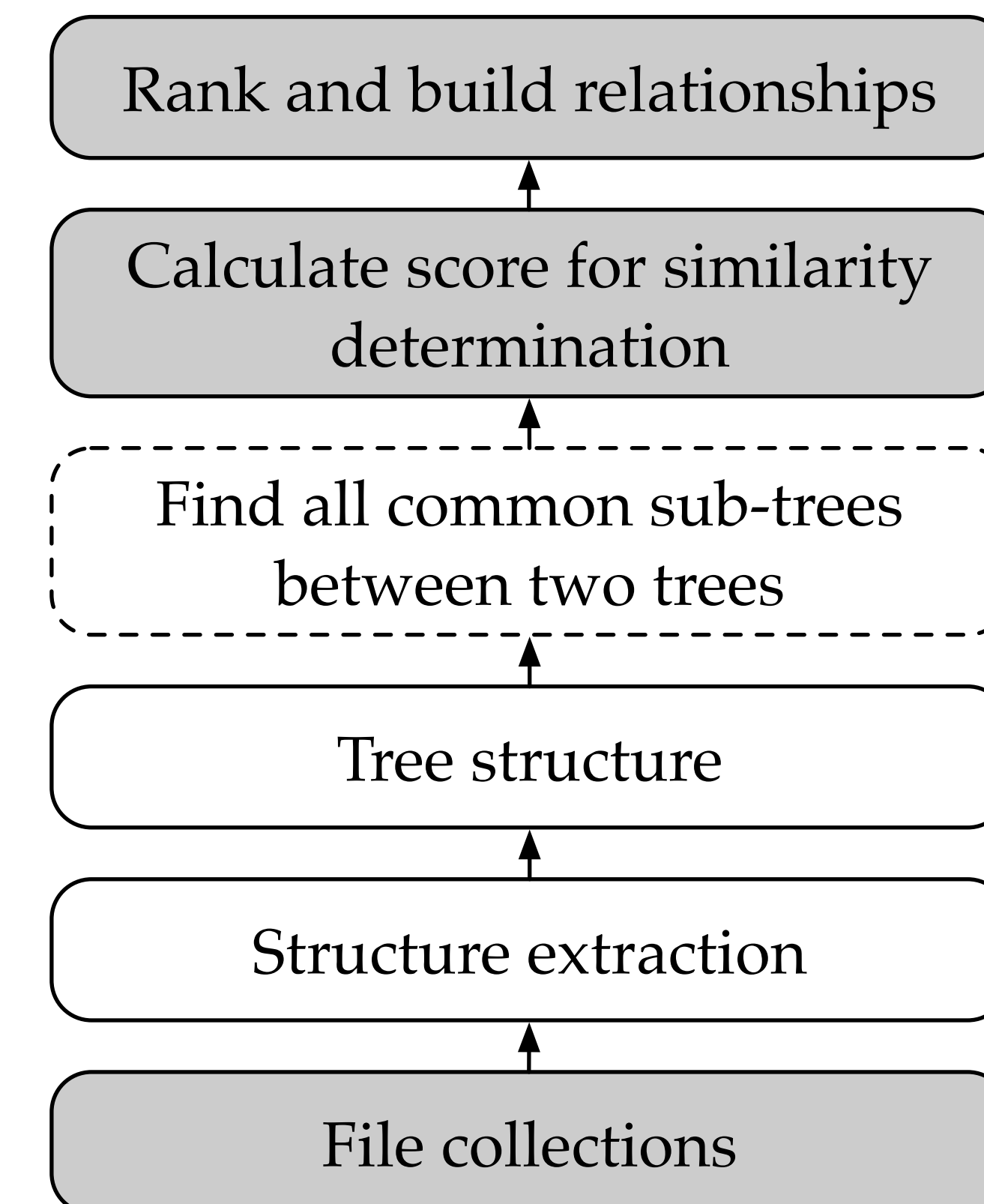
Background

- Many self-describing data formats – such as HDF 5^[1] and netCDF^[2] – have become increasingly popular formats for storing experimental, observational and simulation datasets.
- In self-describing data files, the internal metadata can be seen as a collection of attributes, which are used to describe groups and data objects.
- There are two typical structures found within scientific file: hierarchical structure (tree-like) and flat structure (at most 3 level).



Methods

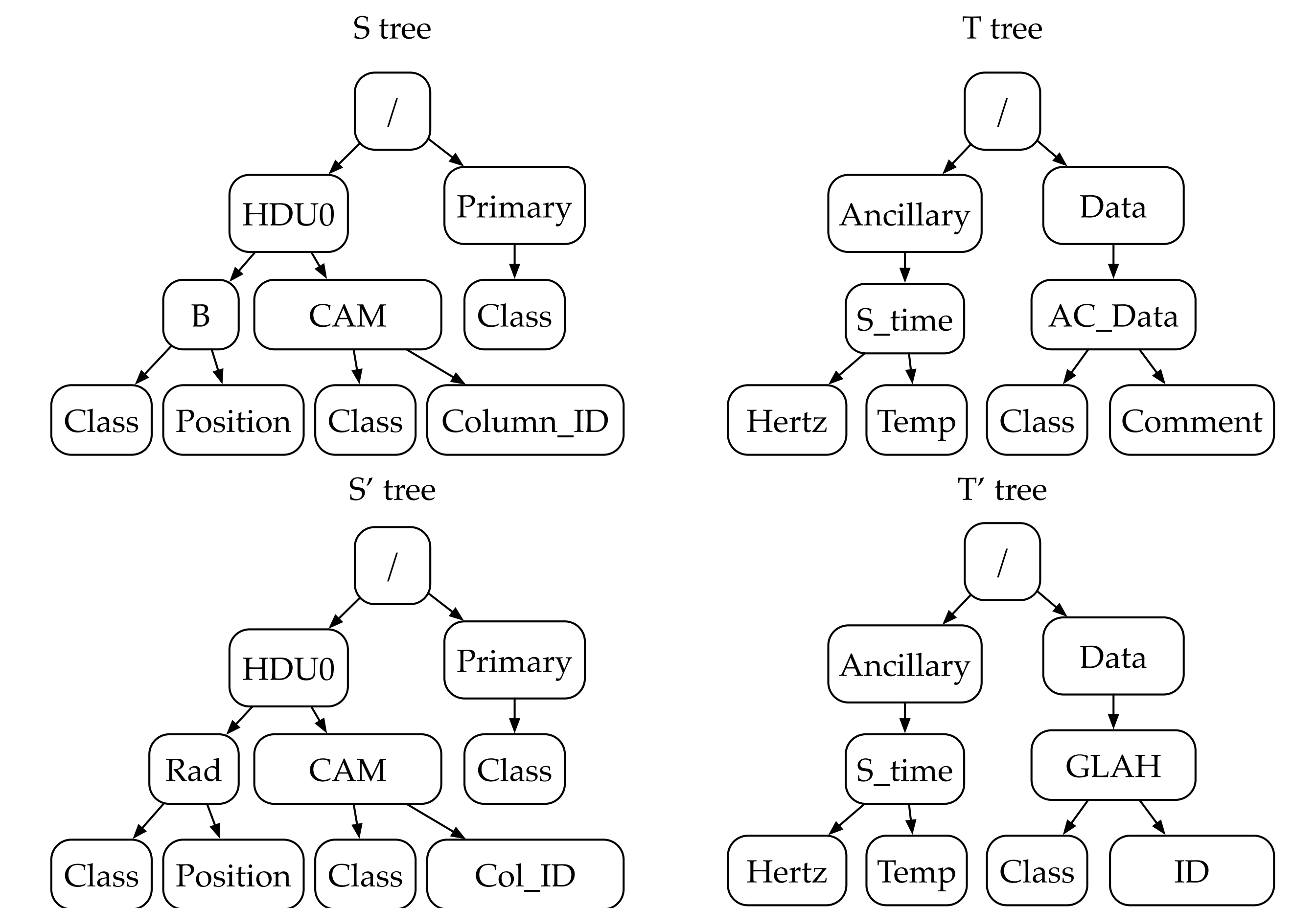
➤ Overview of our design:



- First, we recursively scan the groups, data objects and their attached metadata attributes of each sub-directory from file collections.
- Tree structure has been extracted with groups and data objects as parent nodes and attribute keys as leaf nodes.
- Find all common sub-trees to represent the common structure of these two scientific files
- Calculate “similarity scores” of any two scientific files from a set of files.
- Rank the files and build relationships.

Relationship capturing process

- **Suffix tree on the original trees:** a suffix string is used to support common tree discovery between two files. For example, for S tree, the suffix string is /(HDU0)(B)(Class)#(Pos)(CAM)(Class)#(Col ID)###(Primary)(Class). Suffix string is labelled with pairs of integers (starting position and length) representing sub strings. The concatenation of the labels on a path from the root to an attribute leaf represents the starting position of the suffix string.
- **Mark the attribute leaf nodes:** mark all the nodes of the suffix tree which have at least one red leaf node and one blue leaf node as a descendant.
- **Find the deepest marked nodes:** compute the value $d(l)$ for each blue leaf node of the first suffix tree. First, we attempt to find the deepest marked node u along the path from the root to the leaf node l . If it exists, we take $d(l)$ as the length of $w(u)$. Otherwise, we set 0 to $d(l)$. Then we compute the value $len(l)$ as the length of one leaf for each red leaf node of A .
- **Mark common leaf nodes:** For the leaf node l in A , if $len(l) \leq deep(l)$ (depth of node l), the corresponding sub-tree B is a common sub-tree of S and T .
- **Find common sub-trees:** recursively perform the finding deepest marked nodes and find all possible sub-trees for any two scientific files.
- **Calculate percentage:** given a base file and a set of target files, we will find all common sub-trees between the base file and any target file and calculate the structure percentage of the base file.



Suffix String(S) = /(HDU0)(B)(Class)#(Pos)##(CAM)(Class)#(Col_ID)###(Primary)(Class)

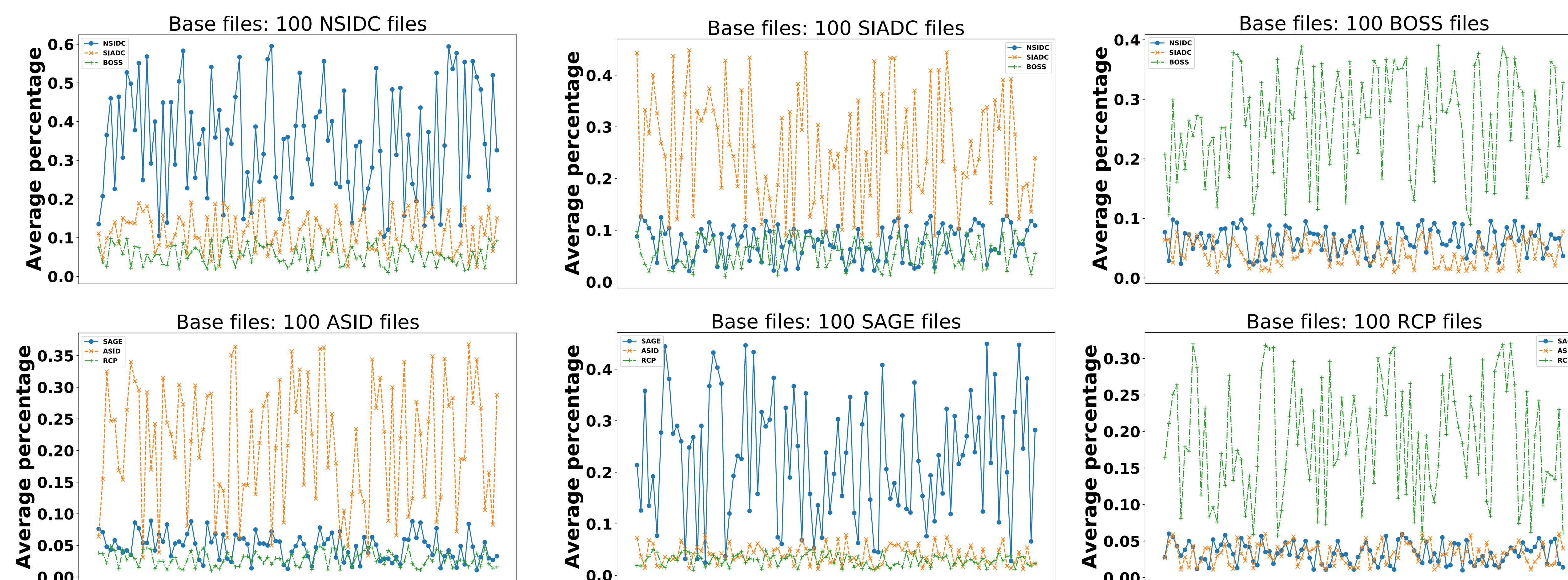
Suffix String(S') = /(HDU0)(B)(Class)#(Pos)##(CAM)(Class)#(Col_ID)###(Primary)(Class)

Suffix String(T) = /(Ancillary)(S_time)(Hertz)#(Temp)###(Data)(AC_Data)(Class)#(Com)

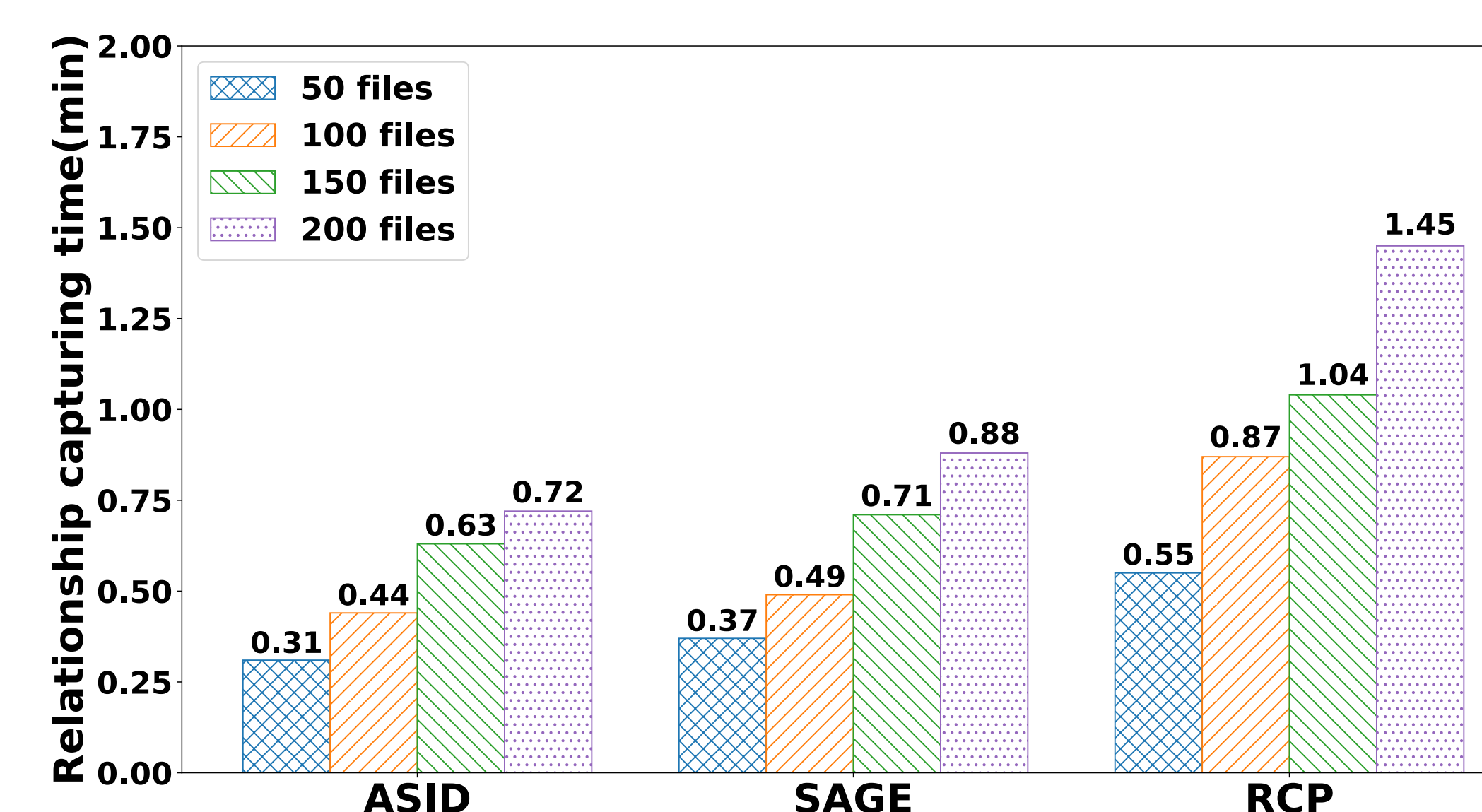
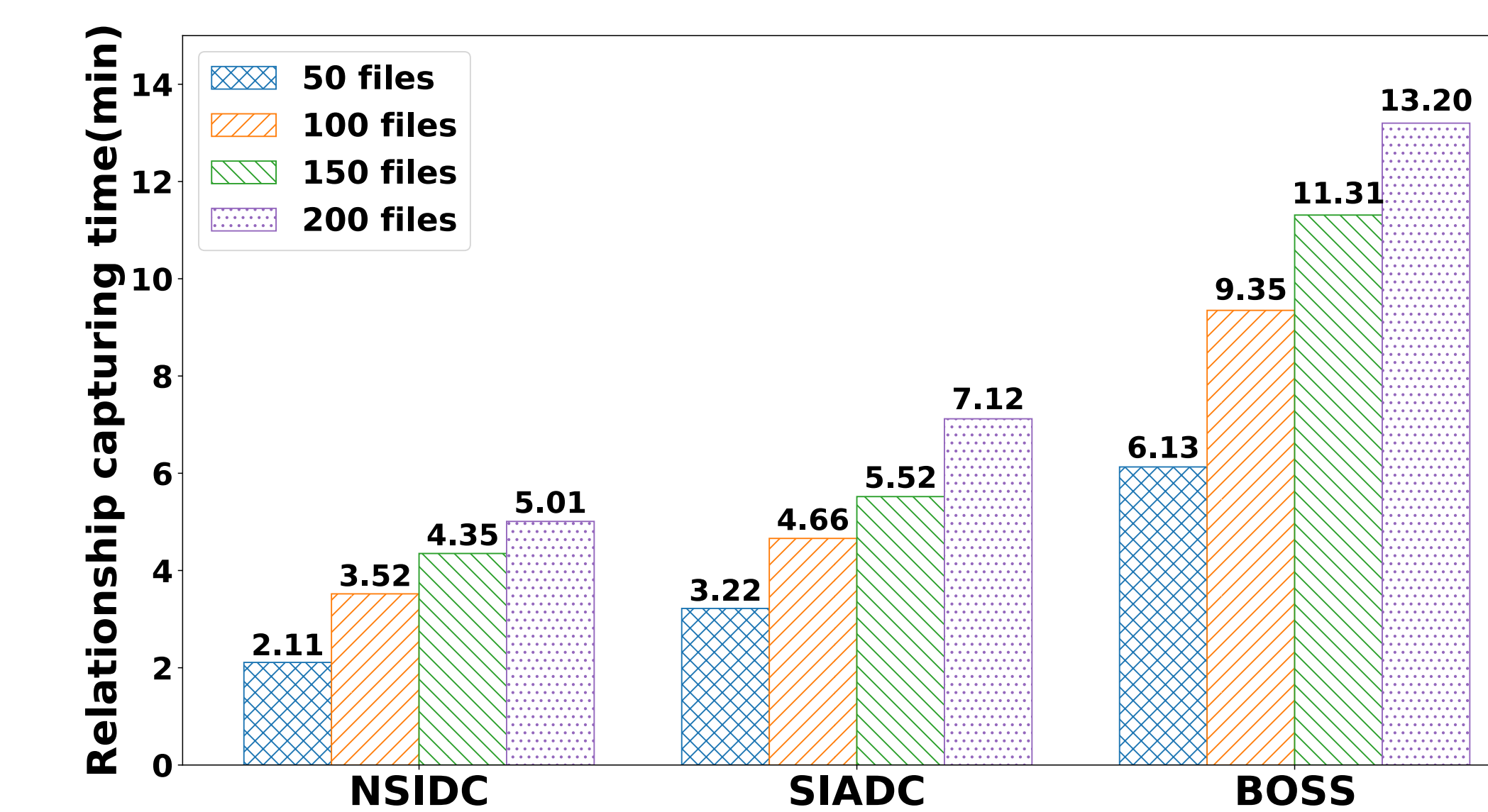
Suffix String(T') = /(Ancillary)(S_time)(Hertz)#(Temp)###(Data)(GLAH)(Class)#(ID)

- **Relationship capturing:** after the structure similarity determination, we can capture a novel relationship based on similarity in order to link self-describing data files. The percentage is the similarity “score”, which represents the structure similarity and metadata similarity. We will rank the “score” and link the scientific files according to their structure similarity relationships.

Evaluation



- We performed this experiment with six sets of self-describing files, including 3 sets of HDF5 files and 3 sets of netCDF files. As shown in the six figures, our method effectively differentiates certain files from a set of files if they are originally from the same organization or share similar structure and attribute keys. For example, if we select a NSIDC file as a base file, the average percentages from NSIDC files are much higher than SIADC and BOSS files. In addition, the experimental results prove the generality of our method, which indicates that this method can be applied in both hierarchical and flat structures.



- As shown in the two figures above, our method proved to be efficient for HDF5 files and netCDF files.

Conclusion & Future Exploration

- In this research, we have explored a methodology to capture relationships based on structure similarity determination for self-describing files.
- We demonstrate the potential of this approach through experiments of real-world scientific datasets.
- We will continue improving the efficacy of the relationship capturing methodology and develop a better storage principle for linked scientific files.

Acknowledgement

- We are thankful to the anonymous reviewers for their valuable feedback. This research is supported in part by the National Science Foundation under grant CCF-1409946, CCF-1718336, and CNS-1817094.
- This work is supported in part by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

References:

- [1] M. Folk, A. Cheng, and K. Yates, “HDF5: A File Format and I/O Library for High Performance Computing Applications,” SC 1999.
- [2] R. Rew and G. Davis, “NetCDF: An Interface For Scientific DataAccess,” IEEE computer graphics and applications, vol. 10, no. 4, pp.76–82, 1990.

