COMP9313 - Project 4

COMP9313 Big Data Management

Project 4 - Similar Join

Hao Fu - z5102511

The main way to optimise the algorithm is using prefix to reduce the input data. Following is the formula to get our prefix array length.

$$|\boldsymbol{R} \cup \boldsymbol{S}| \geq |\boldsymbol{R} \cap \boldsymbol{S}| * \boldsymbol{t} \geq \boldsymbol{max}(\boldsymbol{R}, \boldsymbol{S}) * \boldsymbol{T} \geq \boldsymbol{ceil}(\boldsymbol{max}(\boldsymbol{R}, \boldsymbol{S}) * \boldsymbol{t})$$

Here is the Scala code to get the prefix array.

```
Var list = data.sorted
Var prefix = ceil((s_list.length).toDouble * (1 - t)).toInt + 1
Var pre_list = list.slice(0, prefix)
```

For example, in our project the $\boldsymbol{t} = \mathbf{0.85}$, the input size can be reduce at least **85%**. Assume there are 100M input dataset. The actually input dataset is 15M. But before using prefix array. We need sort the list. The sort func can be any kind of sort algorithm. Here I use the Scala default sort function. In this way, it can avoid suffix array element appear in another prefix array.