A Space Efficient Solution to the Frequent String Mining Problem for Many Databases*

Adrian Kügel and Enno Ohlebusch

Faculty of Engineering and Computer Sciences, University of Ulm, D-89069 Ulm

In the frequent string mining problem, one is given m databases $\mathcal{D}_1, ..., \mathcal{D}_m$ of strings and searches for strings that fulfill certain frequency constraints. The constraints consist of m pairs of thresholds $(minf_1, maxf_1), ..., (minf_m, maxf_m)$ and one wants to find all strings ϕ that satisfy $minf_i \leq freq(\phi, \mathcal{D}_i) \leq maxf_i$ for all i with $1 \leq i \leq m$, where $freq(\phi, \mathcal{D}_i) = |\{\psi \in \mathcal{D}_i : \phi \text{ is a substring of } \psi\}|$.

Fischer et al. [2] presented an algorithm that solves the frequent string mining problem in linear time under the assumption that the number of databases is treated as a constant. The space consumption of this algorithm, however, is proportional to the total size of all databases. We improve this algorithm in such a way that its space consumption is proportional to the size of the largest database, and it takes linear time regardless of the number of databases. Also, our algorithm is more flexible in the sense that one of several databases can be replaced without having to recalculate everything, that is, intermediate data can be stored on file and be reused.

Algorithm for the Frequent String Mining Problem

- For each database $\mathcal{D}$ from the set of databases $\{\mathcal{D}_1, \ldots, \mathcal{D}_m\}$ do:
  - Construct the enhanced suffix array of all strings in $\mathcal{D}$ and use a modified version of the algorithm of [2] to determine which substrings are relevant, i.e., satisfy the frequency constraint for database $\mathcal{D}$.
  - Store for each suffix ϕ the minimum and maximum length of relevant substrings ψ for which ϕ is the lexicographically smallest suffix which has ψ as a prefix.
- Iteratively calculate the intersection of the relevant substrings of databases $\mathcal{D}_1$ and $\mathcal{D}_2$, then the intersection of the result with the relevant substrings of $\mathcal{D}_3$, and so on. This is done by matching the respective database against the enhanced suffix array of $\mathcal{D}_1$.

References

1. Kügel, A., Ohlebusch, E.: A space efficient solution to the frequent string mining problem. Data Mining and Knowledge Discovery 17(1), 24–38 (August 2008)
2. Fischer, J., Heun, V., Kramer, S.: Optimal string mining under frequency constraints. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 139–150. Springer, Heidelberg (2006)

* This is an extended abstract of an article published in the Data Mining and Knowledge Discovery journal [1].