25th Meeting of the Wiesbaden Group on Business Registers
- International Roundtable on Business Survey Frames

Tokyo, 8 – 11 November 2016

Rr. Nefriana and Said Mirza Pahlevi
Badan Pusat Statistik – Statistics of Indonesia
Session No. 5

Technology

Tuning Statistical Business Register System Matching Feature

Abstract

Badan Pusat Statistics, Statistics – Indonesia, has been developing a Statistical Business Register (SBR) system. It is engined with PHP language and backed by Microsoft SQL Server. One of the prominent features provided by the system is its matching function. The matching function is necessary in the SBR system because there is no single unique business ID in Indonesia; hence the function is used to avoid duplication of business maintained in the SBR system. Currently, the SBR system employs Full Text Search (FTS) technology provided by the Microsoft SQL Server. The FTS ranks top 25 most similar businesses based on the names and addresses of the businesses, and the users/operators decide which one of the companies actually matches the business that is already stored in the SBR system. Unfortunately, we found two main issues in the matching process, namely, the precision of the FTS rank and the system performance for performing matching.

To do the experiment, first, we randomly selected 400 businesses from the register for each experiment. Then, for each selected business we searched top 25 most similar businesses also from the (same) register. Because the selected businesses were not removed from the register (after the sample selection), ideally the business should be positioned at the first rank of the search result. Next, we measured the distance between the rank where the searched companies were supposed to be (the first rank) and the rank we actually got from the search result. We also recorderd the time to run the matching query for each sample company.

This paper describes our approach to solve the two issues by tuning the database architecture, removing stop words from the name and address fields of the businesses, and also translating nonstored query procedure to stored query procedure. Our purpose was to lessen the distance by tuning the database architecture and removing stop words, also to improve query running time by translating query procedure. The database architecture was tuned by building a single database that contains subset variables (variables that used in matching mechanism only; those that used for indexing and those that appear for operators to decide whether two businesses matched or not) from all variables of the businesses. The idea of this single table was to integrate three tables that currently used in our SBR database architecture where businesses reside separately by their type of statistical units (enterprise group, enterprise, and establishment)[1]. Our hypothesis was that by doing this we could make the weight of the searching done by the FTS the same for all type of unit statistics and in the end would improve the precision. To remove the stop words, first we analized what words appear

often in businesses' names and adresses (in Indonesian language). Then, each time our matching query was run, the variables values were already cleaned from those stop words. We found that tuning database architecture and removing stop words are effective to improve the precision of the searching results. Because we also recorded the query time while doing those two experiments, we also found that those two methods also improve its query time performance. Next, we evaluated the impact of translating nonstored query procedure to stored query procedure. The results shows that the database architecture tuning approach makes a positive impact on the performance. Therefore, with these experiments results, we have applied the stop word removal in the SBR system and will use the tuning database architecture and tuning query procedure in the near future.

[1] The reason of this separation is related to our approach on linking module where an enterprise can have another enterprise as its child.