COMP9313 (18S1) Project 4
DUE ON 09:59 PM 3 JUN, 2018 (SUN)
z5136414 Andrew Wirjaputra

This report will briefly describe the Spark implementation used to find all record pairs in a file with Jaccard similarity higher than the given threshold.

Stage 1: Order Tokens by Frequency

Since the records are not guarantee to be sorted, at first we will need to calculate the frequency of each token in the record collection, which can be done via a simple MapReduce. We will then broadcast the resulting Map for efficient data distribution. We use Map so we can retrieve the frequency of each token directly by specifying the key.

Stage 2: Finding "similar" ID Pairs

In the second stage, we will scan the original input file and extracts the prefix of each record using the token order computed by the first stage. By using individual prefix tokens as keys, we can simply group the result by key to find candidate records for matching. This step called **Prefix Filter**ing is based on the principle that a pair of sorted records can only meet the given threshold if there is a non-zero overlap in their prefixes.

We can further decrease the candidate records by applying **Length Filter**, that is for a given threshold T and a reference record s0, the size of the matching partners must be within the interval T*|s0| and |s0|/T.

While matching two records (s0, s1), when we examine a token from the prefix of s0 and find a match in the prefix of s1, the matching token may be outside the optimal prefix, so we don't need to consider the pair any further. In general, a candidate pair s0, s1 must be considered only if:

$$minoverlap \leq o + min(|s0| - p0, |s1| - p1)$$

$$minoverlap = \frac{T}{1+T}(|s0| + |s1|)$$

where o is the number of matching tokens so far excluding the current match and p0 (p1) is the position of the current match in the prefix of s0 (s1), which starts at 0. This method of filtering is called **Positional Filter**.

This stage will generate record pairs with similarity equal to or higher than threshold.

Stage 3: Remove Duplicates and Sort

The last stage is simply about removing duplicates (different prefix token group can produce same pairs) and sorting (ascending based on the first record and then the second record).