Mapper

```
class MAPPER
    method MAP(doc id, doc d)
        find year number from doc d and save it as YEAR
        for all word w in doc d do
            EMIT(word w, count 1_YEAR)
```

Reducer

```
class REDUCER
    method INITIALIZE()
        DICT <- new DICTIONARY
        DICT_BY_YEAR <- new DICTIONARY
    method REDUCE(word w, counts [c1, c2, …….])
        for all count c_YEAR in counts [c1, c2, …….] do
            split c_YEAR
            combine w and YEAR
            ADD_TO_DICT_BY_YEAR(word w_YEAR, count c)
            ADD_TO_DICT(word w, count c)
    method ADD_TO_DICT(word w, count c)
        if word w not in DICT
            add it to dictionary DICT[w] = c
        else
            update DICT[w] <- DICT[w] + c
    method ADD_TO_DICT_BY_YEAR(word w_YEAR, count c)
        if word w not in DICT_BY_YEAR
            add it to dictionary DICT_BY_YEAR[w_YEAR] = c
        else
            update DICT_BY_YEAR[w_YEAR] <- DICT[w_YEAR] + c
```

```
method CALCULATE_STD()
    for each word w in DICT do
        avg <- sum(list of values for word w) / 231.0
        create a temporary dictionary TEMP_DICT to store count of a word by its
        year using DICT_BY_YEAR
        sum <- 0
        for w_YEAR in TEMP_DICT do
            partial_square = square(TEMP_DICT[w_YEAR] - AVG)
            sum = sum + partial_square
        std_dev = sqrt(sum/4.0)
        EMIT(word w, avg, std_dev)
```