Merging a Collection of JSON Documents

Problem

Suppose we have a large number of JSON files with the same structure and about the same type of data. The problem is merge (consolidate) them into one file and reduce duplicates. The main problem is that multiple files may have overlapping pieces of data. We want to remove the duplicates.

A Case Study

Suppose a crawler gathers user comments posted for news articles at various news outlets, e.g., Washington Post. Suppose the crawler gets the data in json format. The crawler may visit a news article several times. Let A be a news article and J_1, …, J_m be successive json files retrieved by the crawler in the m times it visits article A. We expect $J_i \subseteq J_{i+1}$. The names of the files follow the following naming convention:

ArticleID_datetime.txt

Hence, J_i and J_{i+1} have the same ArticleID (assigned automatically to A), but different datetimes, with J_i's datetime being earlier than that of J_{i+1}'s.

An example to show the merging process.

28732_20181116010926.txt

There are two comment nodes (id: **4188537143** and id: **4188547414**)


```
{
        "editableUntil":"2018-11-17T15:07:20",
        "dislikes":0,
        "numReports":0,
        "likes":0,
        "message":"...",
        "id":"4188537143",
        "createdAt":"2018-11-10T15:07:20",
        "author":{....},
        "media":[],
        "isSpam":false,
        "isDeletedByAuthor":false,
        "isDeleted":false,
        "parent":null,
        "isApproved":true,
        "isFlagged":false,
```

"raw_message":"...",
"isHighlighted":false,
"canVote":false,
"thread":"7032906508",
"forum":"mtmpph",
"points":0,
"moderationLabels":[],
"isEdited":true,
"sb":false
},
{
"editableUntil":"2018-11-17T18:09:27",
"dislikes":1,
"numReports":0,
"likes":2,
"message":"...",
*"id":"**4188547414**",*
"createdAt":"2018-11-10T18:09:27",
"author":{....},
"media":[],
"isSpam":false,
"isDeletedByAuthor":false,
"isDeleted":false,
"parent":null,
"isApproved":true,
"isFlagged":false,
"raw_message":"...",
"isHighlighted":false,
"canVote":false,
"thread":"7032906508",
"forum":"mtmpph",
"points":0,
"moderationLabels":[],
"isEdited":true,
"sb":false
}

28732_20181119011419.txt

There are two comment nodes (id: **4188560843** and id: **4188547414**)

{
"editableUntil":"2018-11-17T18:19:21",
"dislikes":0,
"numReports":0,
"likes":0,

```
        "message":"<p>...</p>",
        "id":"4188560843",
        "createdAt":"2018-11-10T18:19:21",
        "author":{...},
        "media":[],
        "isSpam":false,
        "isDeletedByAuthor":false,
        "isDeleted":false,
        "parent":null,
        "isApproved":true,
        "isFlagged":false,
        "raw_message":"...",
        "isHighlighted":false,
        "canVote":false,
        "thread":"7032906508",
        "forum":"mtmpph",
        "points":0,
        "moderationLabels":[],
        "isEdited":true,
        "sb":false
},
{
        "editableUntil":"2018-11-17T18:09:27",
        "dislikes":1,
        "numReports":0,
        "likes":2,
        "message":"...",
        "id":"4188547414",
        "createdAt":"2018-11-10T18:09:27",
        "author":{....},
        "media":[],
        "isSpam":false,
        "isDeletedByAuthor":false,
        "isDeleted":false,
        "parent":null,
        "isApproved":true,
        "isFlagged":false,
        "raw_message":"...",
        "isHighlighted":false,
        "canVote":false,
        "thread":"7032906508",
        "forum":"mtmpph",
        "points":0,
        "moderationLabels":[],
        "isEdited":true,
        "sb":false
```

}

Each comment node has its unique **id** (shown in bold). If two comments from different files have the same ids, then they are one and the same comment.

We want to accomplish two tasks:

1) Keep all unique comment codes.

Thus, we keep three comment nodes: **4188537143**, **4188547414** and ***4188560843***.

2) Update the attribute value for the duplicated nodes based on the json latest document.

Comment node **4188547414** is a duplicated one and two attribute values of this node have changed: *"dislikes"* and *"likes"*. Since 28732_20181119011419.txt is the latest json document, we keep the attribute values as they are in this version of the document, which are *"dislikes":1, "likes":2*.

Target:

The goal is to create a new json document that merges the content of the 2 json documents and removes the duplicate comment nodes. The new file will contain **all** (but **unique**) comment nodes from previous files. Thus, the merge should give us a new file 28732.txt, which has the following content:

{
"editableUntil":"2018-11-17T18:19:21",
"dislikes":0,
"numReports":0,
"likes":0,
"message":" ... ",
*"id":"**4188560843**",*
"createdAt":"2018-11-10T18:19:21",
"author":{...},
"media":[],
"isSpam":false,
"isDeletedByAuthor":false,
"isDeleted":false,
"parent":null,
"isApproved":true,
"isFlagged":false,
"raw_message":"...",
"isHighlighted":false,
"canVote":false,
"thread":"7032906508",
"forum":"mtmpph",
"points":0,
"moderationLabels":[],

```
        "isEdited":true,
        "sb":false
},
{
        "editableUntil":"2018-11-17T18:09:27",
        "dislikes":1,
        "numReports":0,
        "likes":2,
        "message":"...",
        "id":"4188547414",
        "createdAt":"2018-11-10T18:09:27",
        "author":{....},
        "media":[],
        "isSpam":false,
        "isDeletedByAuthor":false,
        "isDeleted":false,
        "parent":null,
        "isApproved":true,
        "isFlagged":false,
        "raw_message":"...",
        "isHighlighted":false,
        "canVote":false,
        "thread":"7032906508",
        "forum":"mtmpph",
        "points":0,
        "moderationLabels":[],
        "isEdited":true,
        "sb":false
},
{
        "editableUntil":"2018-11-17T15:07:20",
        "dislikes":0,
        "numReports":0,
        "likes":0,
        "message":"...",
        "id":"4188537143",
        "createdAt":"2018-11-10T15:07:20",
        "author":{....},
        "media":[],
        "isSpam":false,
        "isDeletedByAuthor":false,
        "isDeleted":false,
        "parent":null,
        "isApproved":true,
        "isFlagged":false,
        "raw_message":"...",
```

"isHighlighted":false,
"canVote":false,
"thread":"7032906508",
"forum":"mtmpph",
"points":0,
"moderationLabels":[],
"isEdited":true,
"sb":false
}

Basic tasks of Merge Function:

1) Duplicates: If a comment node with the same id appears in two json document, then the merge json document will contain only one of them.
    a. Verify whether there are attributes whose values have changed. If so, update the values according to the most recent json document as given by the datetime in the file name.
2) Additions: If a new comment node appears in a newer version of the json document, simply add it to the merged version.
3) Deletes (for 5517 only): Monitor deletions. A deletion appears if a comment node appears in file x, but not in in file y, but the timestamp of x is older than that of y's.

Log Changes 5517 Only:

- Maintain a detail log of those instances where and attributes have different values during merge.
- Create a summary about those changes: e.g., give a histogram that shows the frequency of value change per attribute at the end of a merge job.
- Possible inconsistencies
    - For instance, we expect the entire content of J_i to be in J_{i+1}. This condition may be violated in practice, i.e., some nodes in J_i is not in J_{i+1}. Of course, you will need to include them in the merged file.
    - Keep a detail log about such occurrences, e.g., the pair of files and node id that appears in the file J_i but not in J_{i+1}.

Requirements:

1) Implement a non-parallel version of your solution algorithm.

Data:

You will be given a dataset for implementation and debugging your solution. We will use a different dataset for testing your solution. The json docs will contain a node **cursor**. You can ignore it.