BLOCKING

1.Blocking Development Procedure

Questions answered in this section:

How did you develop the final blocker? What blocker did you start with? What problems did you see? Then how did you revise it to come up with the next blocker? In short, explain the *development process*, from the first blocker all the way to the final blocker (that you submit in the IPython file).

Step1: Blocking based on movie title:

- Initially we applied an overlap blocker on the movie title with the following specifications: overlap count=2, split on word. The problem with this blocking method is that it eliminated many true positive matches wherein the movie name has only one word. example: "Inception", "Insurgent", "Frozen".
- Next we applied a rule based blocker with edit distance as the similarity measure to match movie titles. This blocking was very restrictive and it blocked true positives. Specifically, for the following types of movie names: "The Avengers (extended edition)" and " The Avengers", "Captain America (Blu-Ray Extended Edition)" and "Captain America".
- Finally, we applied an overlap blocker with overlap size=1 and split based on word. This is working as expected and is not eliminating true positives when checked with the debugger.

Step 2: Blocking based on director name:

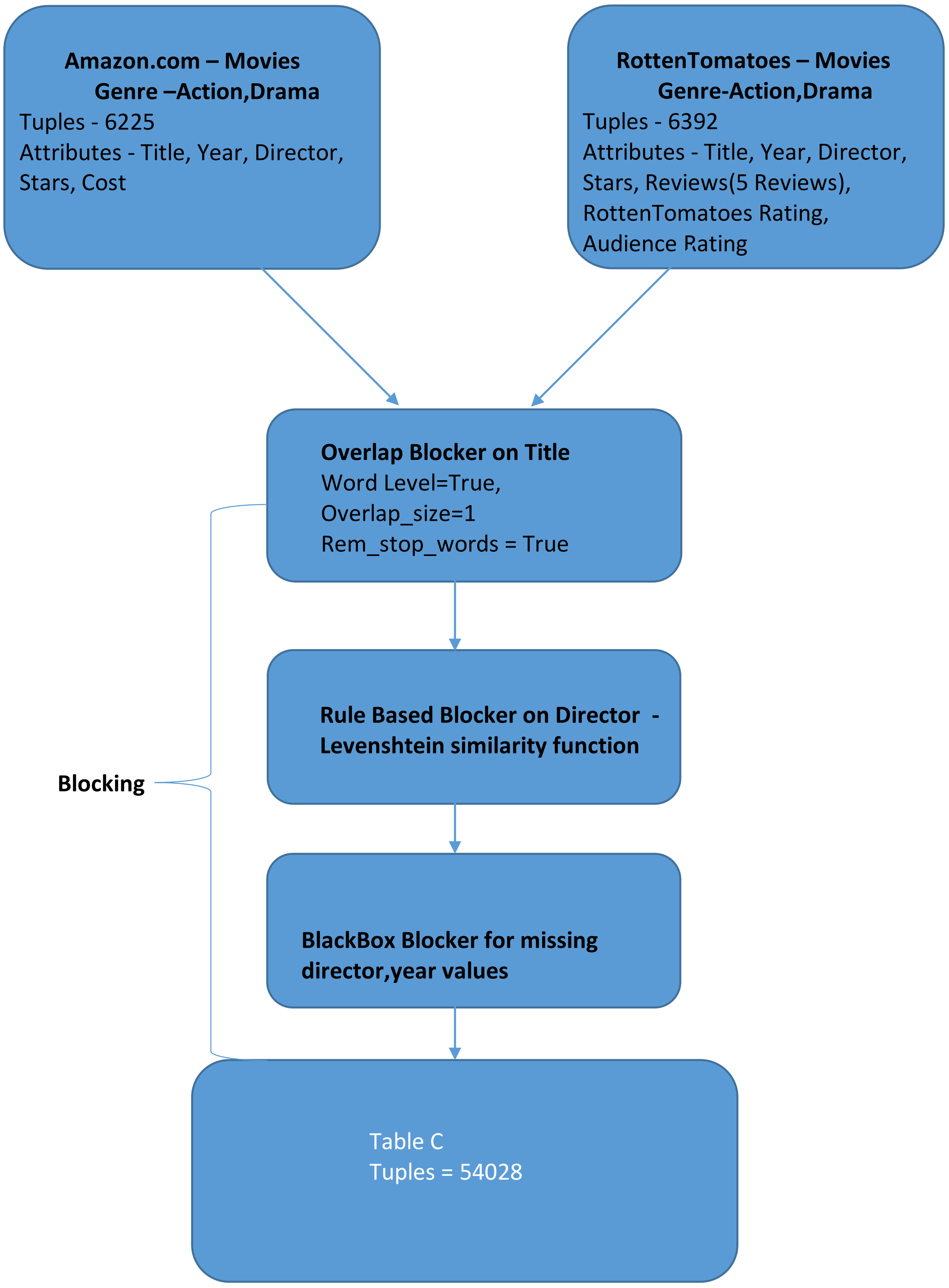
- The first problem here was that a number of tuples from table A (amazon.com) had missing values for the director, year attribute whereas table B has no missing values for the director, year attribute. If we perform overlap, attribute equivalence blocking, then the tuples with missing values for the director attribute would be blocked and would not be present in the final candidate set. This is not correct because the tuples with missing director or year values could still match with corresponding tuples in table B. In order to handle this case correctly we have applied a rule based blocker followed by a black box blocker to handle the missing values.
- To match director names, we have used Levenshtein similarity. We initially set the threshold to 0.8. this seemed to be eliminating many true positive matches. we reduced the threshold to 0.7,0.6 and finally 0.4

  Ex: Lee Jeong-beom, Jeong-beom Lee will match only for threshold 0.4,

- Next we applied a black box blocker to handle the missing values. If the director value is missing but the release year matches, then the tuple is not blocked. If year is also missing, then the tuple is not blocked. If director value is missing but release year does not match, then that tuple is blocked. As there are many tuples in table A with missing director values and missing year values, this black box blocker has been applied in order to reduce the number of obvious non-matches

BLOCKING

Blocking Overview


Amazon.com – Movies
Genre –Action,Drama
Tuples - 6225
Attributes - Title, Year, Director, Stars, Cost
RottenTomatoes – Movies
Genre-Action,Drama
Tuples - 6392
Attributes - Title, Year, Director, Stars, Reviews(5 Reviews), RottenTomatoes Rating, Audience Rating
Overlap Blocker on Title
Word Level=True,
Overlap_size=1
Rem_stop_words = True
Rule Based Blocker on Director - Levenshtein similarity function
BlackBox Blocker for missing director,year values
Table C
Tuples = 54028
Blocking

Final Blocker:

Step 1: Blocking based on movie title
  Blocker: Overlap Blocker
  - overlap size =1
  - split based on space

Step2: Blocking based on director name
  Blocker: Rule Based Blocker using Levenshtein Similarity measure
  - Threshold = 0.4

Step3: Blocking tuples with missing values
  Blocker: Blackbox blocker

```
If Director name is missing
    If Year is missing
        Do not block:
    Else if year matches:
        Do not block:
    Else:
        Block:
```

2.Blocking Debugger Usage and Problems

Did you use the debugger? If so, where in the process? And what did you find? Was it useful, in what way?

- We used the blocker after each blocking phase to determine if we are losing any true positives in the blocking steps.
- After doing blocking by title, we applied the debug blocker and checked the output. No true positives were being eliminated
- After doing rule based blocking on director name, we applied the debug blocker. here. It helped to determine the correct threshold value for Levenshtein similarity measure.
- The debugger helped to identify true positives that are being blocked at each stage, and to fine tune the blocker to prevent elimination of true positives.

3.Blocking Procedure Time Taken

How much time did it take for you to do the whole blocking process?

- We took a total of 3-4 days to complete the blocking process.
- We tried multiple iterations of the blocking process. In each iteration we used a different combination of attribute based blocker, overlap blocker, rule based blocker and black box blocker. After carefully analyzing the results of each of the different blocker combinations we were able to determine which blocker would be best for the given data set.

4.Table Sizes

Report the size of table A, the size of table B, the total number of tuple pairs in the Cartesian product of A and B, and the total number of tuple pairs in the table C.

- Size of table A: 6225
- Size of table B: 6392
- Size of Cartesian product of A and B: 39,790,200
- Size of table C: 54028

5.Additional Cleaning

Did you have to do any cleaning or additional information extraction on tables A and B?

- While blocking based on director names if the director name is missing in the tuple then we are using release year to determine if the tuples match or not. We have performed additional cleaning in order to extract only the release year from the movie release date. The movie release date format is of the following formats: "dd/mm/yy", "month day, year", "year". We are using only the release year to compare tuples. Therefore, we have used a black box blocker to extract only the release year.

6.Issues with Magellan

Did you run into any issues using Magellan (such as scalability?). Provide feedback on Magellan. Is there anything you want to see in Magellan (and is not there)?

- Overlap blocker discards tuples with missing values for selected attribute, we think there should be an option for the user to specify whether tuples with missing values should be discarded or not. This applies to attribute level blockers as well.
- When Magellan was used with iPython for a long time, large amount of the systems RAM was being used.
- It would be useful to be able to call Magellan's in-built similarity functions from a black box blocker as opposed to using third party python packages in the black box blocker.