**Menus Dataset**

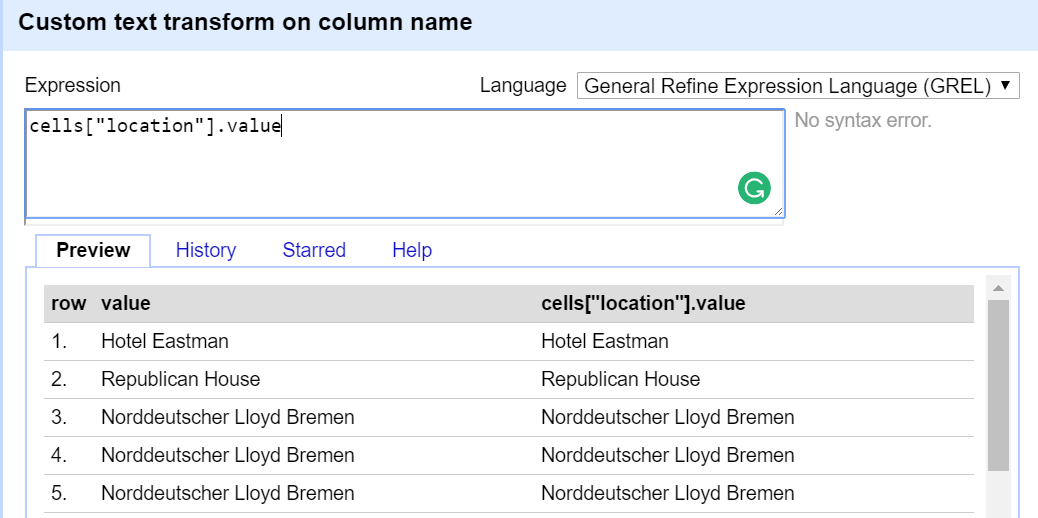
Steps followed

**Column: Id**

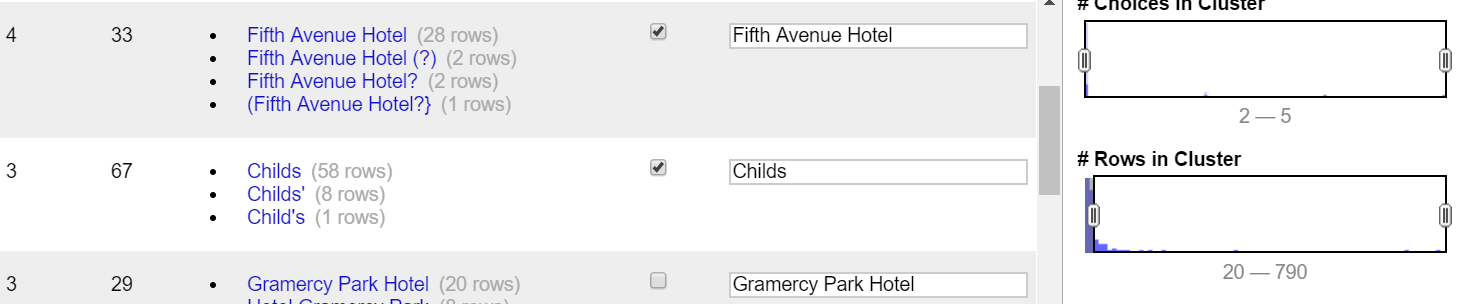
1. Trim
2. Collapse consecutive whitespaces.
3. Transform datatype to number.
4. Check for duplicate values: Used text facet and then sort by count. Found none.
5. Sort values (min: 12463, max: 35526)

**Column: Name**

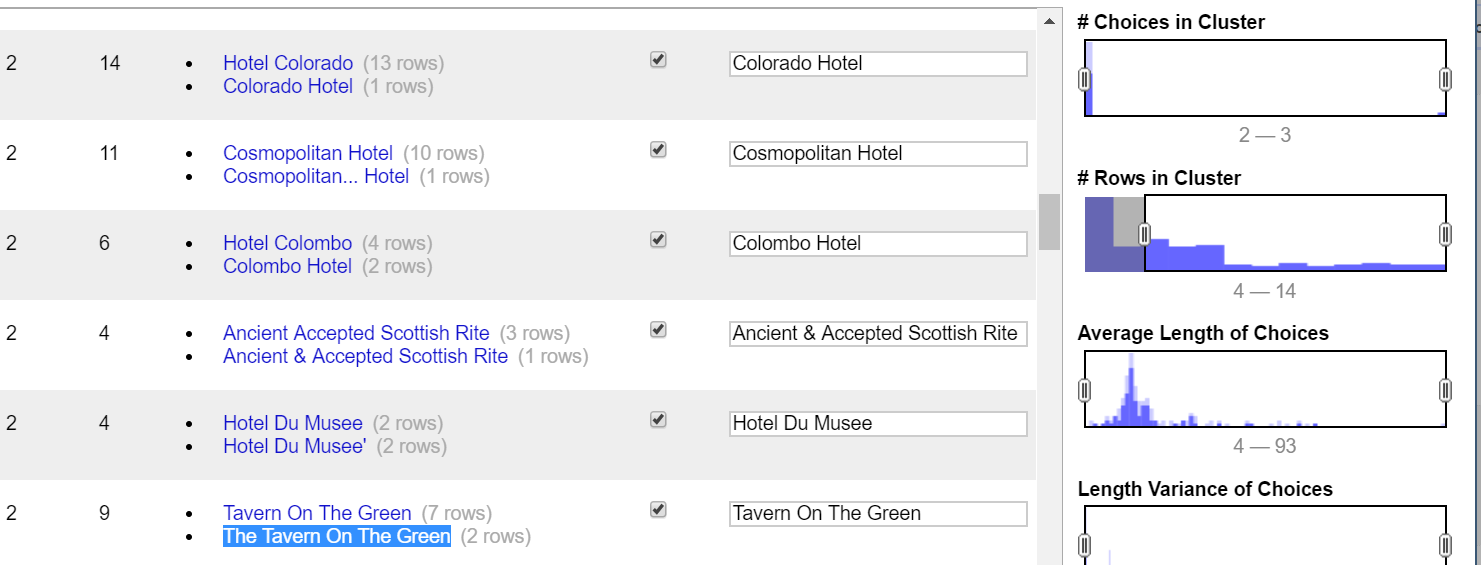
1. Trim
2. Collapse consecutive whitespaces.
3. While checking for the null values in the name column found an interesting fact. The values of the Sponsor and Location columns were identical. To verify this fact used checked a random sample of the Menus data on the [NYPLP Menus](http://menus.nypl.org/menus/) page. The image of the menu card verified that the information was correct. After this performed the below steps for updating the null values.
4. Filtered the null values of the name column.
5. Applied the filter “completed” on the status col using facet. 17453 values were found.
6. Checked the location column for the blank values and found none.
7. Repeated the same step for the menu with status as “under review” and found no rows with null values for the location col.
8. Updated the location values into the name for the null values.



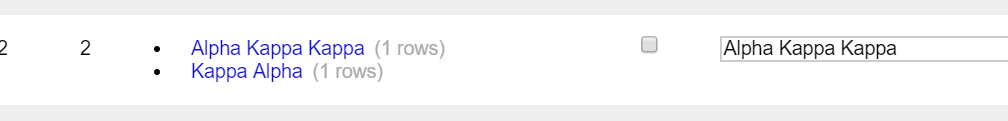
1. Trim
2. Collapse consecutive whitespaces.
3. There were no null values for the name column but “?” represents the null value, hence updated that as a null.
4. Replaced the double quotes in the string, using replace ('"""','') function.
5. Performed clustering on the Column: name using fingerprint function of the key collision method.
6. Performed clustering for the clusters with rows between 20-790.
7. Referred google and Wikipedia for merging the data. For example, childs, child’s and childs’ used Wikipedia.
8. Merged all the hotels in the last for all the hotels mentioned.



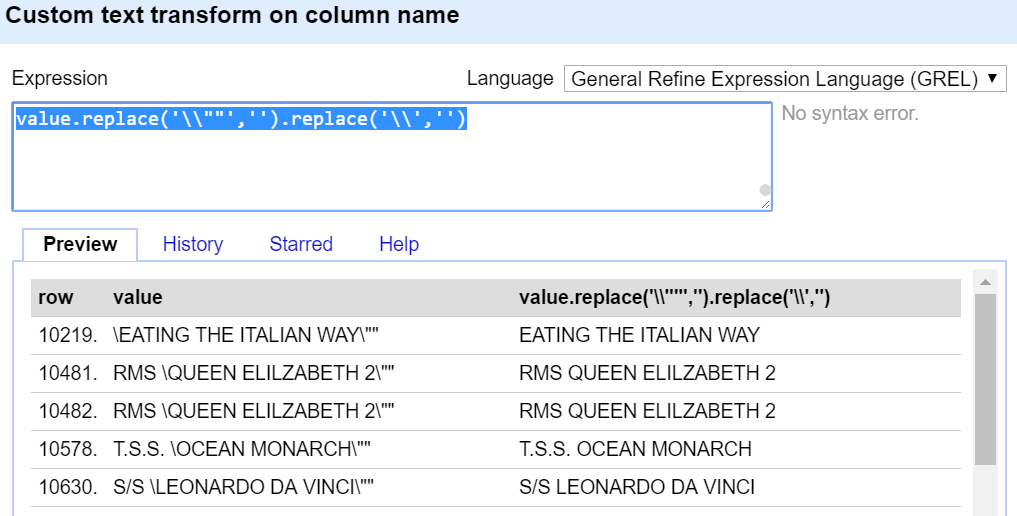
1. Executed the steps again for the clusters with row count of 4-14.



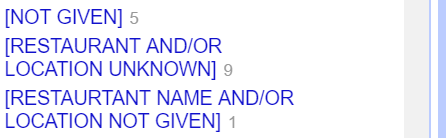
1. Performed the steps again on the data with lesser row count. Didn’t merge the data as below.



1. Replaced “?” using replace function.
2. Removed the “[]” using manual operation from the text.
3. Replaced the values for the text starting with ‘\’ and ending with ‘\””’ using replace function, value.replace('\\""','').replace('\\','')



1. Replaced the below sentinel values to null values. These sentinel values are corrupting the data, hence replaced them with null, so that all garbage and missing values are represented by null.



**Column: Sponsor**

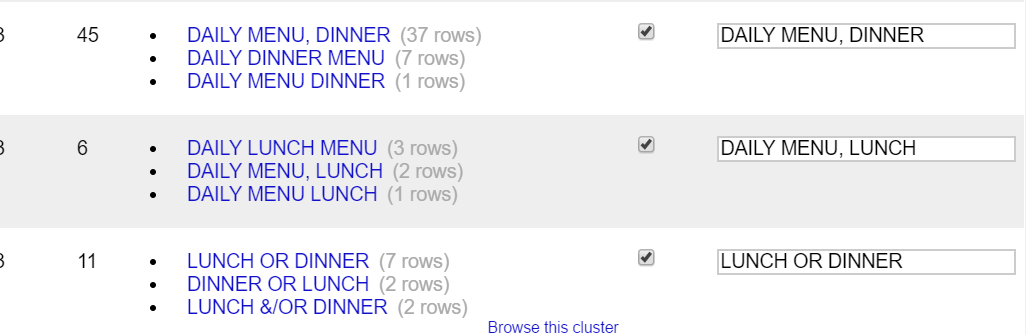
1. Trim whitespaces.
2. Collapse consecutive whitespaces.
3. From the previous cleaning step, I found that it’s better to perform the text to upper case, before clustering.
4. From a random sample, we found that the column name, sponsor and location seems to contain the same values. Hence, using the same logic from the step 3 for the Column: Name, have updated the value of the sponsor column.
5. Filtered the null values of the sponsor column.
6. Used the name column to populate the value into the sponsor column where the value was null.
7. Performed clustering on the column using fingerprint function of the key collision method. Repeated the steps 8, 9 and 10 for the sponsor column as well.
8. Performed further clustering on the column using the n-gram function of the collision method.
9. Replaced the strings containing, ‘?’,’\\’,’\\””’ and?) using replace function as below.

replace('?','').replace('\\""','').replace('\\','').replace('"','').replace('?)','')

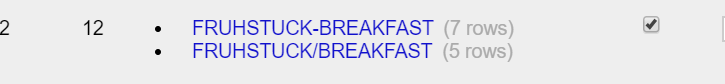
1. Removed the “(“from the start and “)” in the end using the split function as below. value.split('(')[0].split(')')[0].
2. Similarly removed the “[“ and “]” using split function.

**Column: Event**

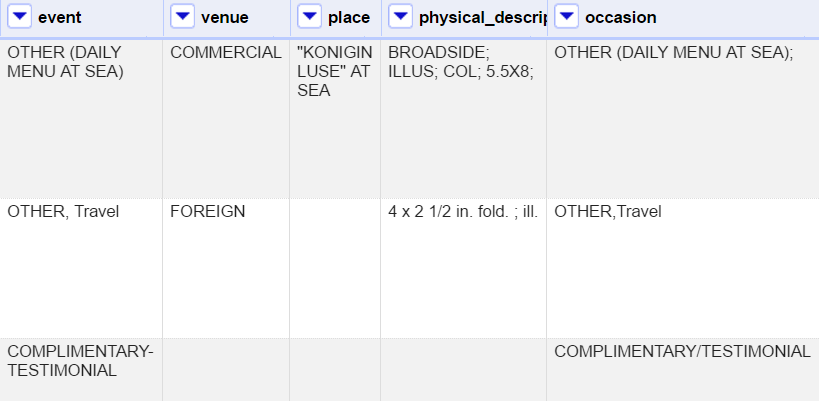
1. Trim whitespaces.
2. Collapse consecutive whitespaces.
3. From the previous cleaning step, I found that it’s better to perform the text to upper case, before clustering.
4. removed the “[“and “]” using split function.
5. Removed “;” using the replace function. Value.replace(‘;’,’’).
6. Used clustering with the fingerprint method of the key collision method.
7. Clustered with the highest no of rows first and then iterated for the lower number.
8. For below values, added the event in the last like lunch or dinner.



1. Used “-“ instead of “/” for concatenating two values.

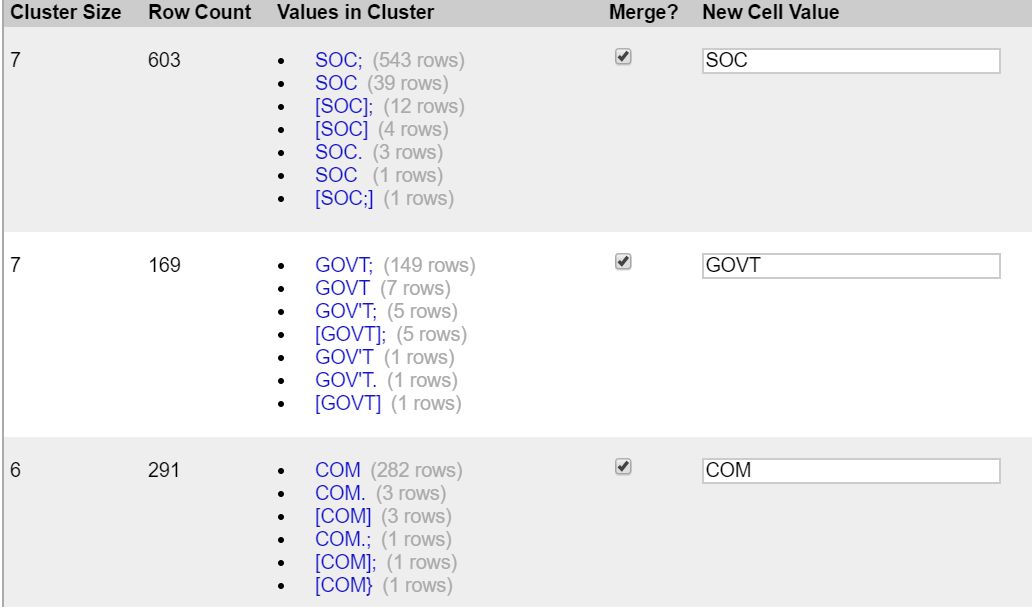


1. Used the n-gram function of the key collision method. And re-did clustering again.
2. For some of the values tried to populate the event value from the occasion. Such as below:



**Column: Venue**

1. Trim whitespaces.
2. Collapse consecutive whitespaces.
3. From the previous cleaning step, I found that it’s better to perform the text to upper case, before clustering.
4. Removed “?” using the replace function.
5. Used Clustering with keycollision method and fingerprint function.



1. After clustering observed that there are multiple values for the same type. For example, EDUC represents short form of EDUCATIONAL Venue, MIL as MILATARY and so on. Hence merged these values to a generic form. The EDUC will be replaced with EDUCATIONAL, MIL with MILITARY etc.

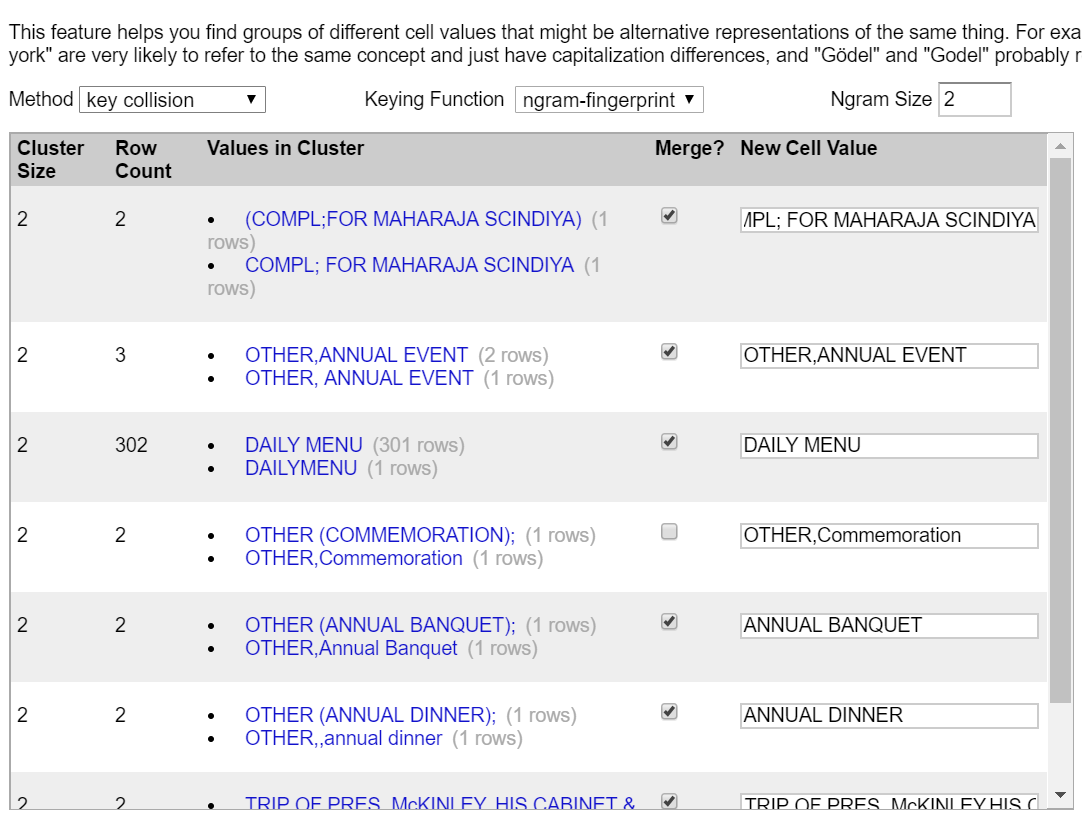


**Column: Occasion**

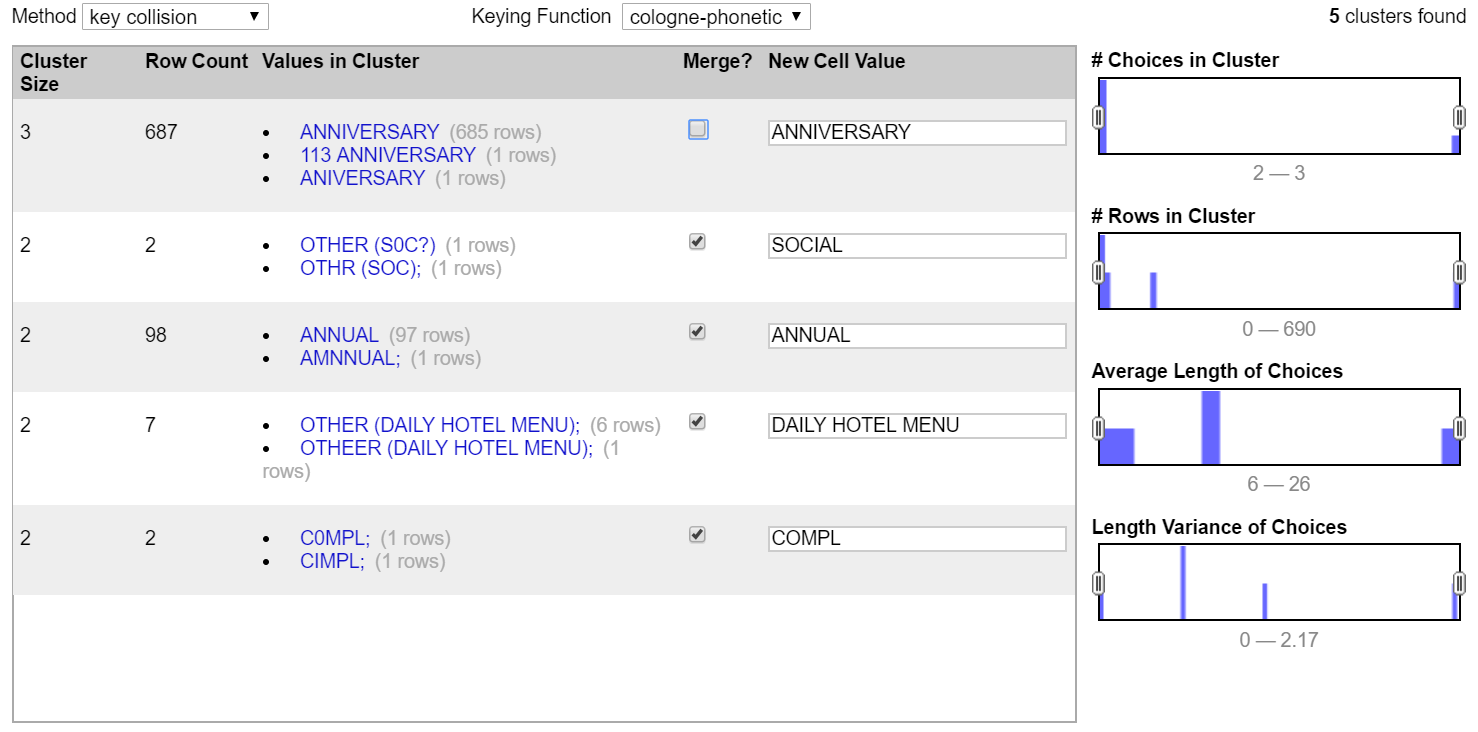
1. Trim whitespaces.
2. Collapse consecutive whitespaces.
3. Removed the word “OTHER” to make the data more consistent.
4. Removed “;,?”using replace function



1. Used Clustering with keycollision method and fingerprint function.
2. Used Clustering with keycollision method and ngram-fingerprint function with Ngram size 2.



1. Used Clustering with keycollision method and cologne-phonetic function



**Column: call\_number**

A code made of letters and numbers assigned to each item in the Library to indicate its location on the library shelves. The call number is required, along (usually) with author and/or title, for the retrieval of library materials. In the old or [Dictionary Catalog](http://legacy.www.nypl.org/research/chss/grd/resguides/lingo.html#dictcat), the call number appears in the upper right-hand corner of the card; it often consists of a few letters of the alphabet. In [CATNYP](http://legacy.www.nypl.org/research/chss/grd/resguides/lingo.html#catnyp), the call number appears at the beginning of the information about the book, [periodical](http://legacy.www.nypl.org/research/chss/grd/resguides/lingo.html#periodical), or other material. [1]

There are two types of call numbers

1. Numeric:

Ex. 1906-017, where the first 4 numbers represent the year and rest of the number represents the shelf.

1. Alphabetic:

Ex. Barrata 60, Soete 180, Zander 140, are the digital libraries and the number represents the shelf.

There are 1562 blank values.

1. Trim whitespaces.
2. Collapse consecutive whitespaces.

[1] <https://www.nypl.org/node/5622>

**Column: keyword** and **Column: call\_number**

These are blank columns

**Column: date**

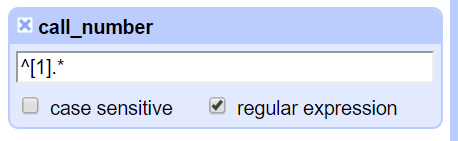
1. Trim whitespaces.
2. Collapse consecutive whitespaces.
3. Using text facet function, these dates are unrealistic.

1. Using the function “facet by blank”

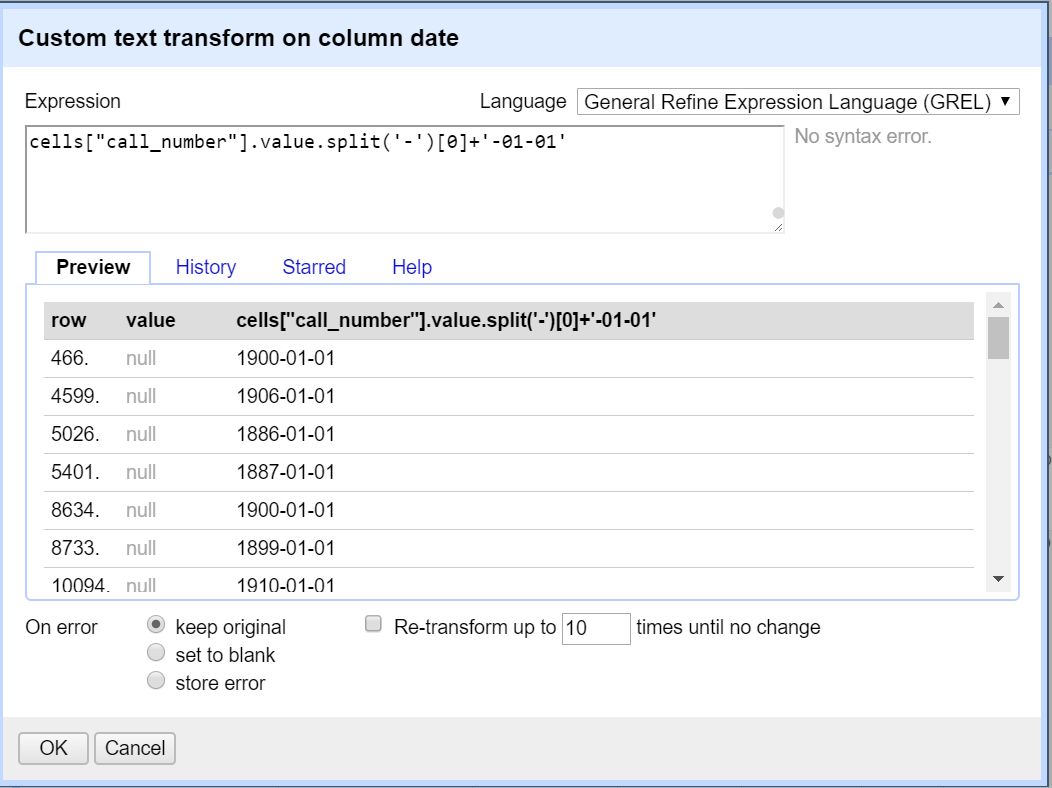


Further, using regular expression “^[1].\*” in the call\_number column to extract the year from it.



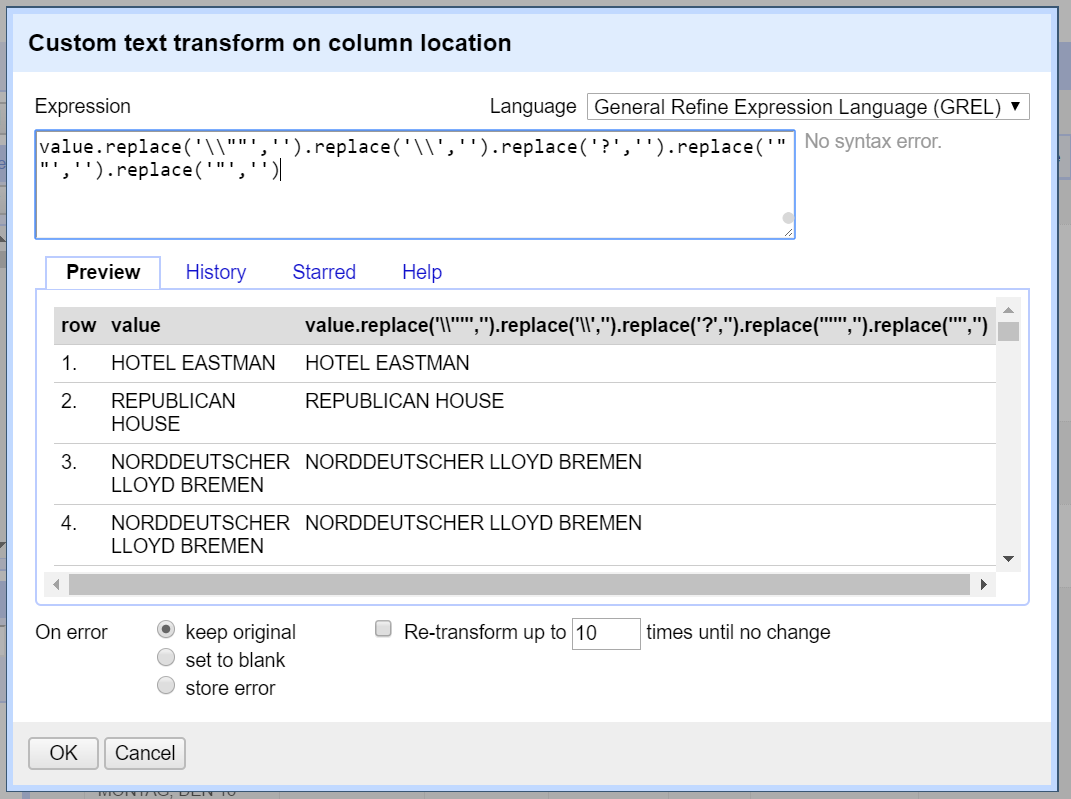
46 blank values were replaced by their respective years and to maintain the date format 01-01 were added to the year using the customer text transform on column date and using the expression

cells["call\_number"].value.split('-')[0]+'-01-01'

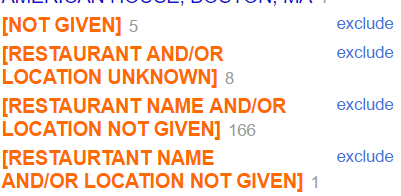


**Column: location**

1. Trim whitespaces.
2. Collapse consecutive whitespaces.
3. Removing (\\””,\\,?,””,”’) using replace function



1. Performed Clustering following the same principles as discussed for the name columns.
2. The below values formed different types of null values, hence changed it to uniform one, i.e. blank. Used the replace function value.replace('[RESTAURANT AND/OR LOCATION UNKNOWN]','').replace('[NOT GIVEN]','').replace('[RESTAURTANT NAME AND/OR LOCATION NOT GIVEN]','')



1. Replaced the “[“and “]” as first and last character, using value.split('[')[0].split(']')[0].

**Column: Location\_type.**

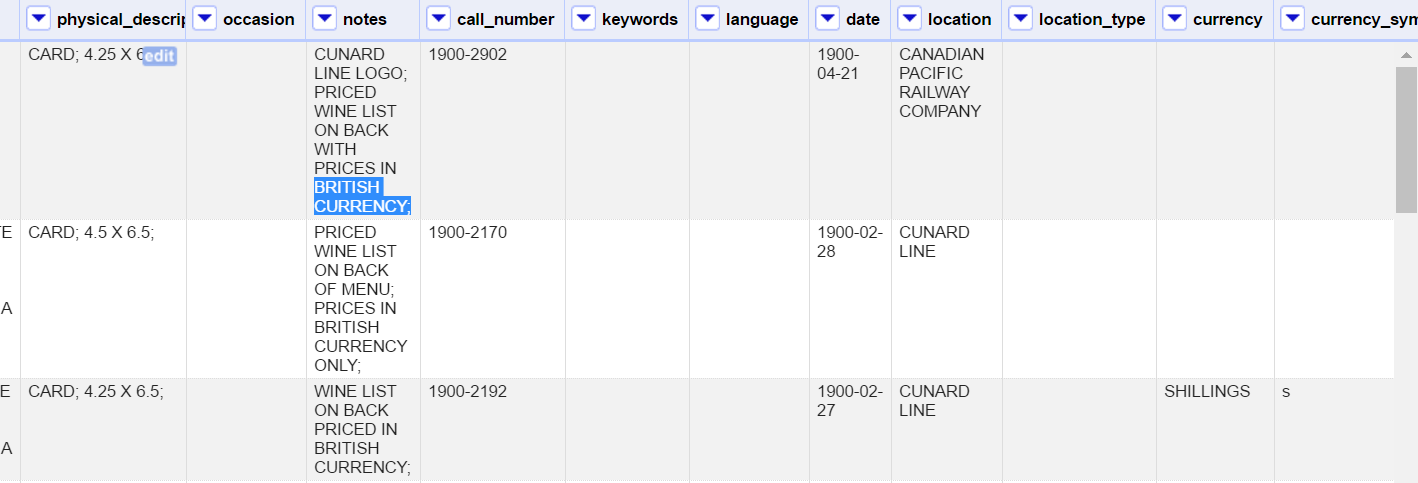
This column doesn’t have any values, hence didn’t perform any cleaning steps.

**Column: Currency:**

1. Performed Trim operation.
2. Collapse consecutive whitespaces.
3. Changed the case to Upper case.
4. For the null values, took some random sample and compared it on the [NYPL menus](https://digitalcollections.nypl.org/items/510d47db-4977-a3d9-e040-e00a18064a99) site to find the price. The menu cards didn’t have the price written on them as well. So couldn’t proceed further.

But using Notes Column we could figure out what currency was used. For example, below rows have notes for BRITISH CURRENCY and a similar row have SHILLINGS as the currency. So, we update these using the notes.

But we were not able to generalize this method for other currencies.



**Column: Currency\_symbol:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Checked the values of the column which are null but have values in the currency column. Later populated those values using that reference.

**Column: Status:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Changed it to upper case.

**Column: page\_count:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Changed it to number data type.

**Column: dish\_count:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Changed it to number data type.

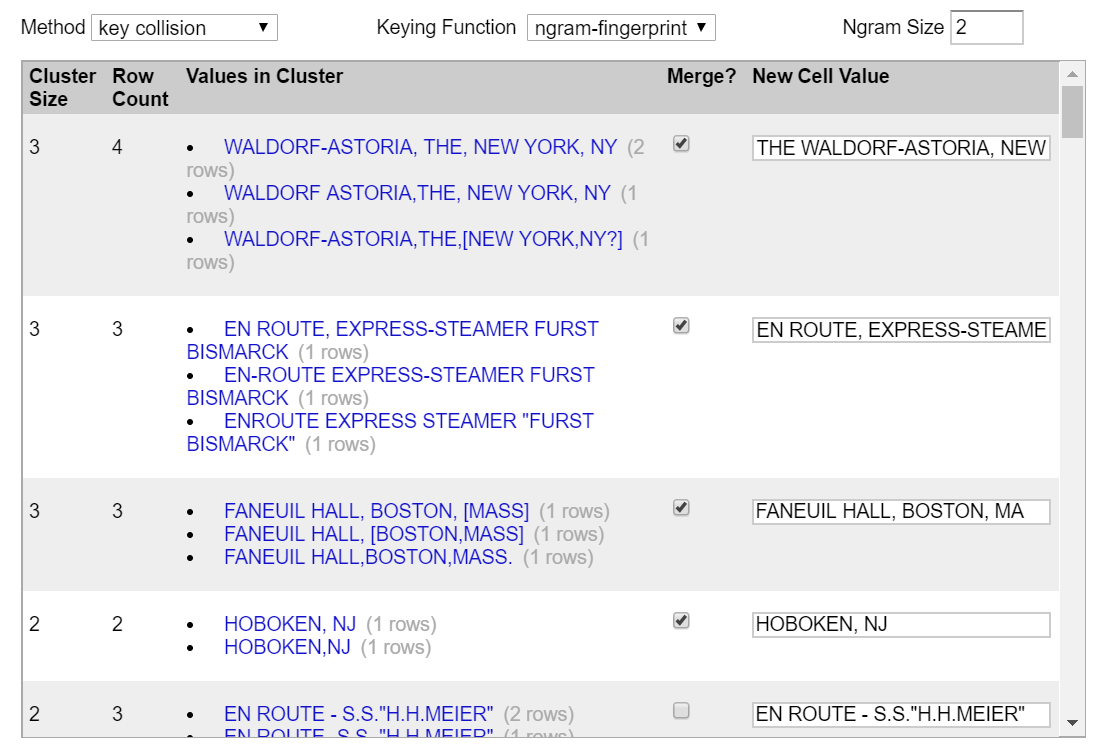
**Column: Place:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Performed clustering for the column using key collision method with fingerprint function.

Also, modified some of the values to standardize the data. For example, added New YORK, NY to the value, add missing 2 characters for state etc. Will split this column into two as well



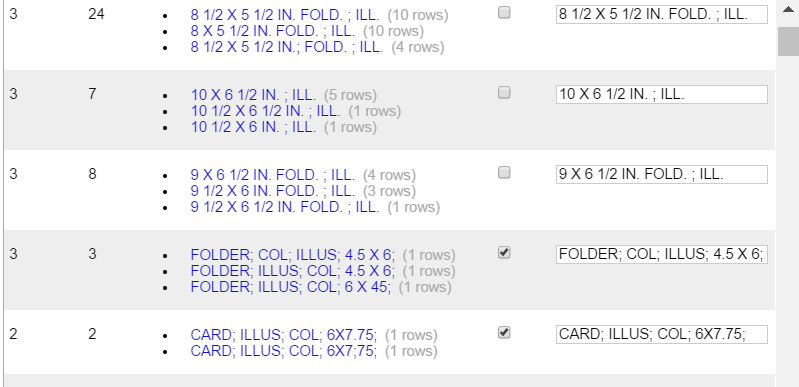
1. Used Clustering with keycollision method and ngram-fingerprint function with Ngram size 2



1. Replaced “(“,”)”, “[“,”]”, “?” using replace function.

**Column name: physical\_description:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Changed the case to Upper case.
4. Performed clustering for the column using key collision method with fingerprint function.
5. Used Clustering with keycollision method and ngram-fingerprint function with Ngram size 2. Didn’t merged the values for numeric string as the size of the menu page might differ in length or breadth.



**Column name: notes:**

1. Performed trim operation.
2. Collapse consecutive whitespaces.
3. Changed the case to Upper case.
4. Performed clustering for the column using key collision method with fingerprint function.
5. Used Clustering with keycollision method and ngram-fingerprint function with Ngram size 2.

