# Doing:

1. summarizing the collected datasets in littrature review 2
2. -Adding short names column for datasets
3. -if a dataset do not have a name, then name it after its author or by using some other naming scheme
4. Add in description: download here (\href{url}{Download/Request here}
5. sort the datasets on catagory
6. Add \midrule to seperate catagories
7. Add search period (from - to)
8. Summerize collections of datasets
9. Ranked list of candidate datasets
10. Document how my review differ for the two similar reviews I found
11. Finish contribution column

# Planning to do next

A review that looks at:

1. The capabilities of search engines and forensic tools with respect to search (e.g. search plugins, encoding support, max number of characters/symbols possible in the search phrase, operators, etc)
2. How storage is done (RAM/Cache/Storage) on the search engines and forensic tool, with respect to search (e.g. how indexes are handled, are there preprocessing involved etc)
3. Compiling the Properties of the search algorithms (e.g. single pattern, multi pattern etc, binary, boolean search, case sensitive/insensitive, synonym/similar words, score) from the open source documentation
4. Finding string search implementation in documentation, source code comments, source code (I do not think that looking at scientific articles is a good way to find this information, I have looked for this before without much success... new approach by using the name of the string search algorithms as search terms will be attempted against search databases, github repository and documentation pages)

## Discussed:

- Criteria for applicable dataset candidates for the experiments should be based on:
    - The dataset should not only contain numerical values
    - Text datasets
- Considering using 1 dataset per category (see section 1.1 literature review 2)
    - Search phrases should be based on domain knowledge (have to have some knowledge of the dataset)
    - It may not be feasible to measure recall if the collection is rather large... this may be possible if I also use some smaller datasets
    - testing use cases such as keywords that do not exists (use hash function to generate a string that will not exist) and measure the performance
- Fulltext search is baseline/top priority, could include fuzzy search and faceted search if time is available after performing fulltext search experiments
- The use of API/source code can be in the appendix section of the thesis
- In the analysis section I have to document and justify the results
- Forensic image tools might be a good place too look for forensic tools that can search.
- October should be the month of preparing and performing the experiments.
-

# Tasks (not a complete list):

| Top priority | Medium Priority | Low priority |
| --- | --- | --- |
| • Finishing the literature reviews<br>• Find a list of popular forensic tools that can perform search... and select some of these for inspection.<br>• Experimental design | • Rewrite research questions (combine my theoretical and practical questions that I have so far)<br>• Write a overview of the properties of string search algorithms and their strength/weaknesses (not a in depth analysis)<br>• Use a list of the names of string search matching algorithms and use this as terms to perform internal search on the github repositories for the forensic tools and search engines. Use the same list to search though the open source documentation pages. Advance search in github allows to search under a path e.g. lucene/core/src/test/org/apache/lucene/search/ | • Look at source code for identifying string matching algorithms (not the may focus and might be very time expensive)<br>• Evaluate how good a implemented search algorithm is (source code), this would only be feasible if the code snipped do not have many dependencies and has low complexity |