CSE 519 -- Data Science (Fall 2017)
Project Proposal

Rohan Vaish - 111447435
Kiranmayi Kasarapu - 111447596
Shruti Nair - 111481332

Project

We will be working on *Automatically Building of Book Indices* project

"An index is a list of words or phrases and corresponding pointers to locations where useful material related to that heading can be found in a document" - (Source - Wikipedia)
Typical book index contains names of people, places events and relevant concepts arranged alphabetically selected by the indexer as thought to be of any interest to the reader. Usually, the indexer may be the author, the editor or a professional indexer. The pointers may be page numbers, paragraph numbers or sections of the documents (or a combination of these).
Our aim for the semester long project is to build an automatic index builder which will take latex books/documents and index size as input and provide us with updated version of the document with the index section added as the output.

Some Background Research

Identifying and extracting **candidate key phrases** which are index worthy. On a broader level, it consists of two parts,

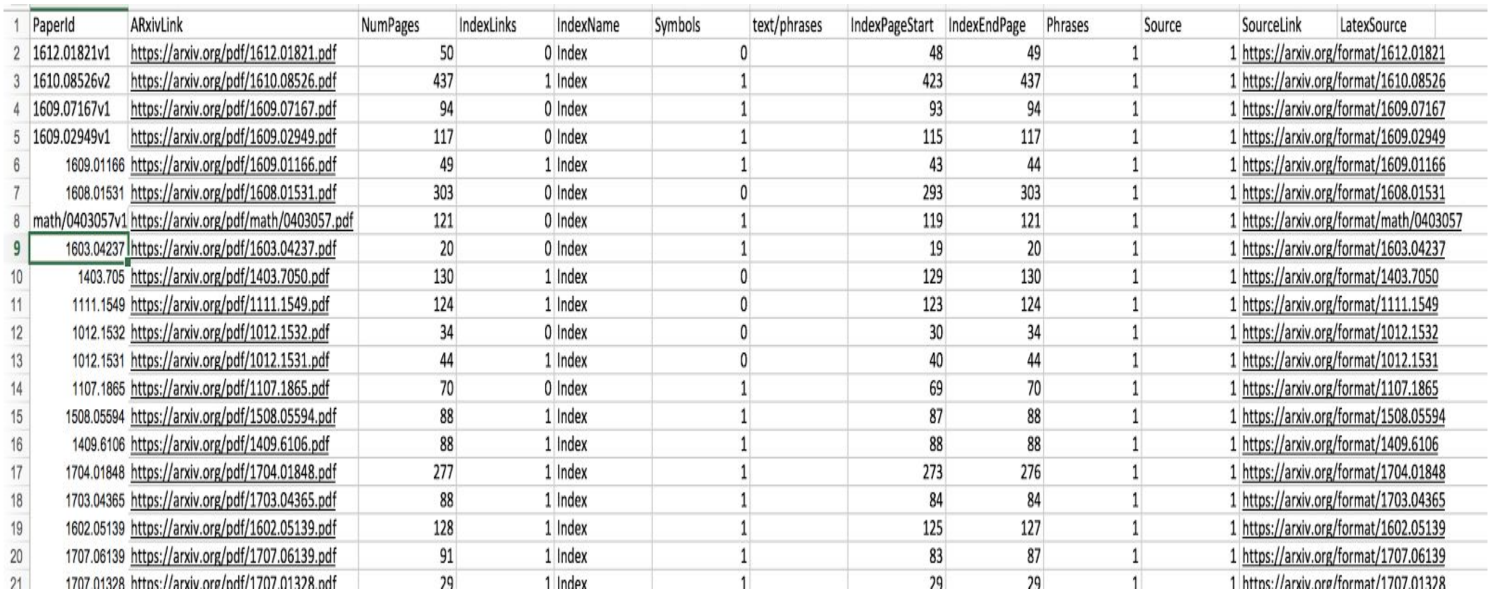
- identifying the largest possible number of phrases/words likely to feature in the index
- minimizing the number of incorrectly proposed candidates.

A simple key phrase extraction can use ***N-grams*** approach (like Google N-grams). This involves extracting all the possible sequences of N words as candidates, typically leaving us with large number of entries. Later, several filters (restrictions) need to be applied to select the relevant entries.
Other approaches include ***noun phrase chunks*** which relies on the theory that major key phrases consist of a noun phrase. There are many other algorithms for this purpose and we need to analyze which one suits indexing a certain kind of document in the best way.
Going forward, we will need to **rank and order** our candidate key phrases. One way might be by distinguishing between the phrase-ness and informative-ness of a phrase. Phrase-ness refers to the degree to which a sequence of words can be considered a phrase, while informative-ness can be calculated using the classical *tf*idf* method or methods like language model informativeness

Data

We have collected data from ARxiv website which have latex sources for most of the journals, articles and papers. Some of these papers have indices. We searched through the site to collect enough papers and articles which are already indexed. We built a dataset using this information. We were able to collect more than 100+ papers which have indices. The latex sources for most of the indexed papers are also available. We went through indices in each paper and analyzed them. Some papers have hyperlinks in the indices which takes back to the phrase/word in the paper. Some papers just mention the page number without hyperlinks. Our goal is to provide hyperlinks, to make it easier for the user to use the index. We have captured these details while building our dataset.

The snapshot of our dataset looks as follows:



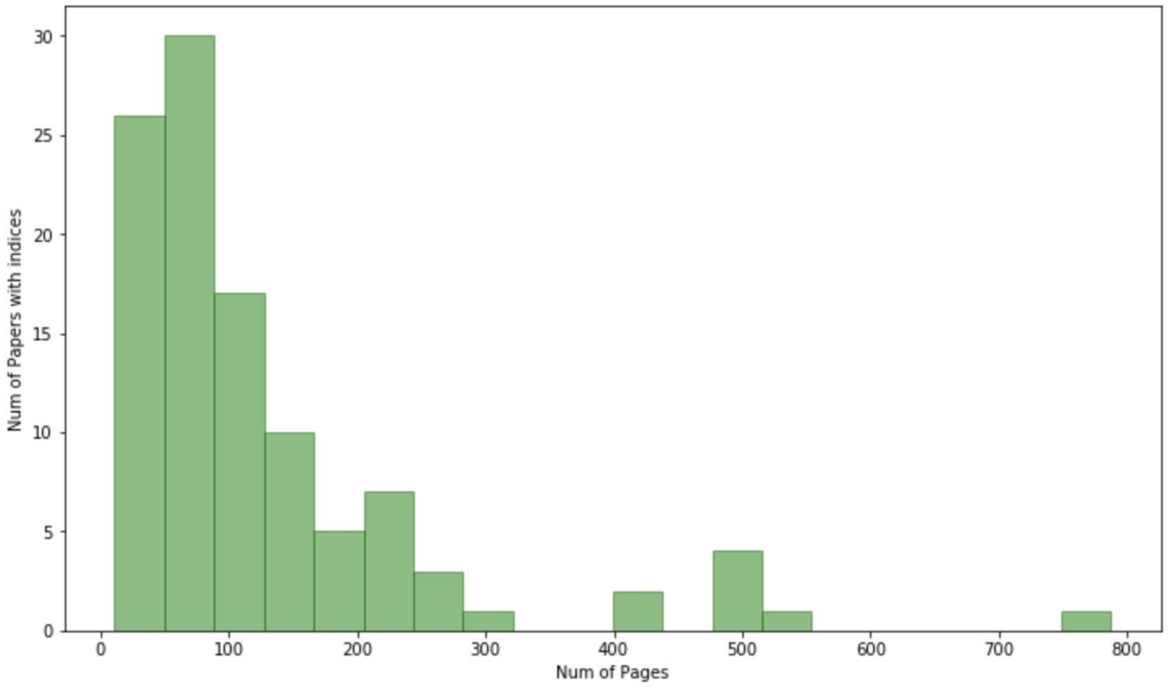
1	PaperId	ARxivLink	NumPages	IndexLinks	IndexName	Symbols	text/phrases	IndexPageStart	IndexEndPage	Phrases	Source	SourceLink	LatexSource
2	1612.01821v1	https://arxiv.org/pdf/1612.01821.pdf	50	0	Index	0		48	49	1	1	https://arxiv.org/format/1612.01821	
3	1610.08526v2	https://arxiv.org/pdf/1610.08526.pdf	437	1	Index	1		423	437	1	1	https://arxiv.org/format/1610.08526	
4	1609.07167v1	https://arxiv.org/pdf/1609.07167.pdf	94	0	Index	1		93	94	1	1	https://arxiv.org/format/1609.07167	
5	1609.02949v1	https://arxiv.org/pdf/1609.02949.pdf	117	0	Index	1		115	117	1	1	https://arxiv.org/format/1609.02949	
6	1609.01166	https://arxiv.org/pdf/1609.01166.pdf	49	1	Index	1		43	44	1	1	https://arxiv.org/format/1609.01166	
7	1608.01531	https://arxiv.org/pdf/1608.01531.pdf	303	0	Index	0		293	303	1	1	https://arxiv.org/format/1608.01531	
8	math/0403057v1	https://arxiv.org/pdf/math/0403057.pdf	121	0	Index	1		119	121	1	1	https://arxiv.org/format/math/0403057	
9	1603.04237	https://arxiv.org/pdf/1603.04237.pdf	20	0	Index	1		19	20	1	1	https://arxiv.org/format/1603.04237	
10	1403.705	https://arxiv.org/pdf/1403.7050.pdf	130	1	Index	0		129	130	1	1	https://arxiv.org/format/1403.7050	
11	1111.1549	https://arxiv.org/pdf/1111.1549.pdf	124	1	Index	0		123	124	1	1	https://arxiv.org/format/1111.1549	
12	1012.1532	https://arxiv.org/pdf/1012.1532.pdf	34	0	Index	0		30	34	1	1	https://arxiv.org/format/1012.1532	
13	1012.1531	https://arxiv.org/pdf/1012.1531.pdf	44	1	Index	0		40	44	1	1	https://arxiv.org/format/1012.1531	
14	1107.1865	https://arxiv.org/pdf/1107.1865.pdf	70	0	Index	1		69	70	1	1	https://arxiv.org/format/1107.1865	
15	1508.05594	https://arxiv.org/pdf/1508.05594.pdf	88	1	Index	1		87	88	1	1	https://arxiv.org/format/1508.05594	
16	1409.6106	https://arxiv.org/pdf/1409.6106.pdf	88	1	Index	1		88	88	1	1	https://arxiv.org/format/1409.6106	
17	1704.01848	https://arxiv.org/pdf/1704.01848.pdf	277	1	Index	1		273	276	1	1	https://arxiv.org/format/1704.01848	
18	1703.04365	https://arxiv.org/pdf/1703.04365.pdf	88	1	Index	1		84	84	1	1	https://arxiv.org/format/1703.04365	
19	1602.05139	https://arxiv.org/pdf/1602.05139.pdf	128	1	Index	1		125	127	1	1	https://arxiv.org/format/1602.05139	
20	1707.06139	https://arxiv.org/pdf/1707.06139.pdf	91	1	Index	1		83	87	1	1	https://arxiv.org/format/1707.06139	
21	1707.01328	https://arxiv.org/pdf/1707.01328.pdf	29	1	Index	1		29	29	1	1	https://arxiv.org/format/1707.01328	

The feature set is explained below:

1. PaperID is the ARxiv ID. We use this to represent each paper/article/journal
2. ARxivLink is the link to the actual paper in pdf format.
3. NumPages represent the number of pages in the paper.
4. IndexLinks, is boolean values which tells us if the index contains hyperlinks or not. A value ‘1’ represents it is hyperlinked, ‘0’ represent plain page numbers.
5. Symbols column represent if there are any symbols that are indexed. This is important because, for some research papers which use certain symbols exhaustively, we will have to decide if we want to index symbols or just the text. This is again a boolean variable which is set to ‘1’ if the certain symbols are indexed.
6. text/phrases - This column tell us if there are is text that is indexed in a particular document, but only symbols are indexed. Its value is 0, if only symbols are indexed.

7. IndexPageStart/IndexPageEnd - They list the starting and ending page numbers of teh index. These fields give a sense of how big the index is.
8. Phrases - This field tell us if the index contains phrases or just single words.
9. Source/SourceLink - These fields tell us if the latex source for the document is available and the link to the latex source file.

Our data set contains around 106 rows.
We did a sample analysis on our dataset to see usually what size of papers are usually indexed.



We see that papers with around 30-120 pages are indexed. We did not find papers of short length around 5-10 pages size to be indexed.

Goals

Our mains goals for this projects are as follows:

1. Keyword extraction - This task is to identify the most important words or phrases that identify a particular document or in a sense define/describe the document in a broader sense. The extracted words and phrases can be used to build the index. For keyword extracting we are going through some of the concepts like Text Mining, Information Retrieval and Natural Language Processing methods.
   We can do keyword extraction on any block of text. For this task, we do not need to supply latex source, plain text will be sufficient
2. Extracting all the text from the Latex Source file: Once we know how to extract key words for a given block of text, now we will have to apply the algorithms on the text from each paper/journal etc. To apply on any paper/journal, we need the text in plaintext format. So the task here is given a latex source file, we will have to extract all the text in plaintext format. We will have to parse the source file according to Latex norms and get

all the content of the paper, so that it can be fed to the keyword extraction algorithm. Once the algorithm returns a set of important keywords, we can proceed further

3. Generating the index in the latex source file: At this point, we have all the necessary and important keywords for a given document. We will have to scan through the latex source file and for every keyword that we want to be used in the index, we have to modify the latex source file and put the index, against each word of interest. This task will generate a modified latex source file. This file upon compiling in Tex, should produce the document with the index built into it.

Future Goals

1. Indexing scanned documents
2. Automating the pdf generation directly, instead of compiling the generated latex source file in the Tex application.

Evaluation

To evaluate how good we are indexing a particular document, we will use our dataset here to verify our results. Our results can depend on multiple factors, like for eg. index size provided by the user, size of the paper, etc.

It may happen that for small index size our program is doing good, and we are able to capture important terms and features related to the document. But for the same document, if we increase the index size, certain important words may not be captured. Since we have various papers and journals of varied length with varied index sizes, we can use these documents to evaluate our results.

Evaluation can be done as a step by step process.

1. We will keep the index size constant and run our algorithm on all the papers in the dataset and verify our results with the actual index in the papers. We will check how many index words are matching the actual index and how many indexes do not match. We can analyze the extra indices and verify if they are actually important or is it just our algorithm is not doing that good.
2. We can take one paper from the dataset and vary the index sizes. We can check for what index size, we are getting maximum index words that are common to the actual index in the paper and what our algorithm provided.
3. We can take papers from different area/topic, like one from maths, one from algorithms and plaintext paper. We can analyze, how our good or bad our algorithm is doing with respect to different subjects areas, and accordingly improve for the others.
4. We can analyze if our algorithm is going good on small papers or long papers, since at least have one long paper of 800 pages, and short paper ranging from 15 pages.