**Data Mining in Google Scholar**

Amey Parab, Baljeet Singh

**Abstract:**

Google Scholar is a platform to search for scholarly articles published in various journals. From one place, search can be made across many disciplines and sources, like research articles, theses, books etc. Search on Google Scholar can be made with title of any research article or name of the Author. The problem with Google Scholar site is that it contains irrelevant citations which are wrongly tagged to the profile owner. For any profile, Google Scholar lists the papers/scholarly articles which are written by them. This list may contain some of the papers which are not written by this author, or some non-scholarly articles are tagged to the profile. There is also a possibility that some of his popular research publications are missing from their profile. This wrong citations usually happen with the authors who has same last name and same initial of first name. Also, the author whose name includes diacritics, ligatures, and apostrophes may miss some citations. This paper intends to refine the results given by the Google Scholar by considering factors like Research interest of the author, keywords from the paper, publisher and co-authors.

**Introduction:**

Google Scholar is an online, freely accessible web search engine that indexes the full text or metadata of scholarly literature across an array of publishing formats and disciplines. Google Scholar lets users to look for the peer-reviewed articles, theses, books, technical reports, abstracts, patent etc. Google Scholar uses Googlebots which are Google’s web-crawling robots to collect documents from the web, filter out the results and make them available via Google Scholar interface. In course of collecting Scholarly documents from the web, some non-scholarly articles are also collected. When a search is made on Google Scholar with any author’s name, it lists all the research articles tagged under him. This may contain some papers which are not written by this author, i.e. it is wrongly tagged under this author. Also, there may be some non-scholarly articles tagged under his name. The wrong citations usually happen when Author’s name (Last name and initial of the first name) is same, or when the author’s name includes diacritics, ligatures, and apostrophes.

**Motivation**: Because of the above discussed problems in Google Scholar, some authors get some papers tagged on their profiles which are not written by then. This happen because of similarity in name with other popular authors. Consequently, this results in the higher h-index of some authors while their publications are very few. Our goal is to make Google Scholar more consistent by ensuring that only Author’s research papers are tagged to their profile.

Our goal is to identify which papers are wrongly tagged under any author based on some features. For author profiles, features are his name, co-author names and his research Interest. For Research papers, features are considered as all author names, Publication date, Research paper’s title and abstract. Based on these features, we will be performing text mining to decide whether these papers belong to the authors.

When a search is made on Google Scholar by the author name, it lists the matching relevant authors with the link to their profiles, their designation, Organization name, number of citations and email id. The one who the user is looking for can be clicked and it will redirect to the Author’s profile. This profile gives the information about the author, his ranking (citation indices), Research Interests, his designation, email, current Organization, Research papers co-authors and optional link to his homepage. The list of title of all the papers tagged to this author is displayed with co-authors, publisher, published year, number of citations and link to the more information for every paper. This information for every paper contains: Authors, Publication date, Conference, Publisher, Description, total citation and link to the paper where it is originally published.

**Feature Selection:**

For every author profiles, the features which are of utmost importance in mining are Author name, co-author names and research interest. For every paper, the features which are of utmost importance are Author names, paper’s title and abstract. Based on above features, clustering and classification will be performed to identify the articles which are wrongly tagged to author.

Data Collection:

As Google Scholar data is not available online, we worked on a program that takes Author name as input and gives all the paper’s information in structured form. For this, we studied the Google Scholar’s HTML structure in and out. We worked on scraping the Google Scholar site to get our data and then wrote to excel files for later.

Following is the information which we extracted:

**For Author**:

* Author’s Name
* Author’s designation
* Author’s Organization
* Author’s Research Interest
* Author’s Homepage (Optional)

**For Research Articles (tagged under Author’s profile):**

* Title
* All Author’s names
* Publication date
* Publisher
* Description

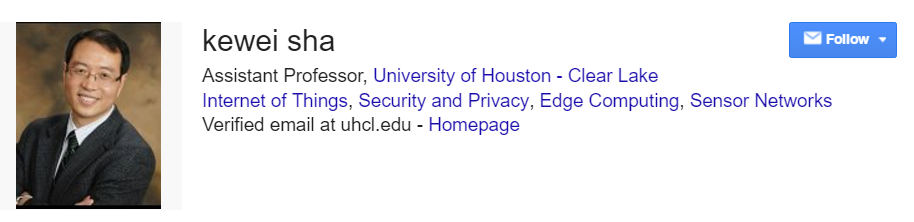
We get all the above data in ready to use form, except the Research Article’s title, abstract and description.

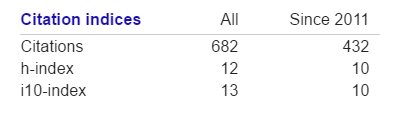
We have used following python’s libraries:

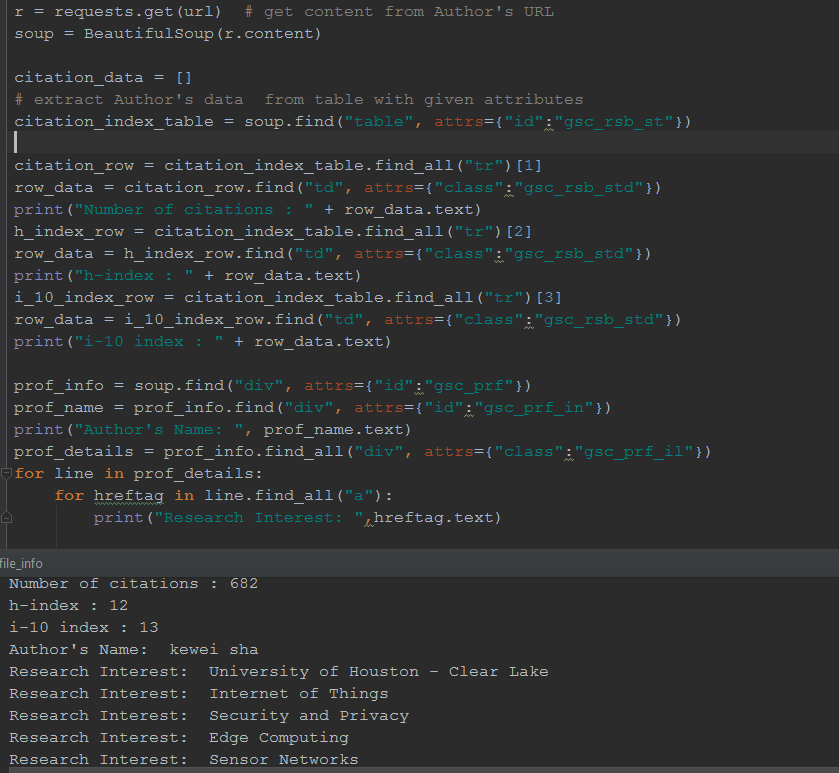
* Requests: Requests is an Apache2 Licensed HTTP library, written in python. The get function in requests takes URL as the input and outputs the whole
* BeautifulSoup: BeautifulSoup is a python library for pulling data out of HTML and XML files.
* CSV: The CSV module is useful for exporting and imported data to and from spreadsheets.
* Jellyfish: Jellyfish is a library of functions for approximate and phonetic matching of strings.
* Difflib: Difflib is a library for comparing sequences.
* NLTK: NLTK is a library for classification, tokenization, stemming, parsing and semantic reasoning.
* Anytree: This is a library which has predefined implantation of tree data structure.
* RE: The re module provides matching operations.

Code snippets:

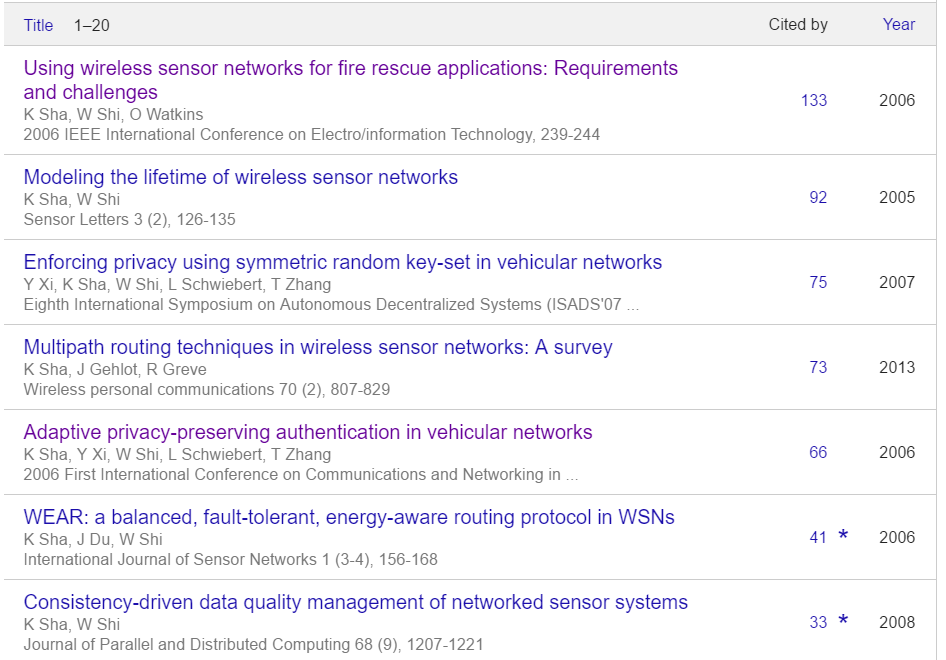
1. Author’s profile: To extract Author’s information (Name, Research Interests, designation, current organization and link to his homepage), below code is designed:

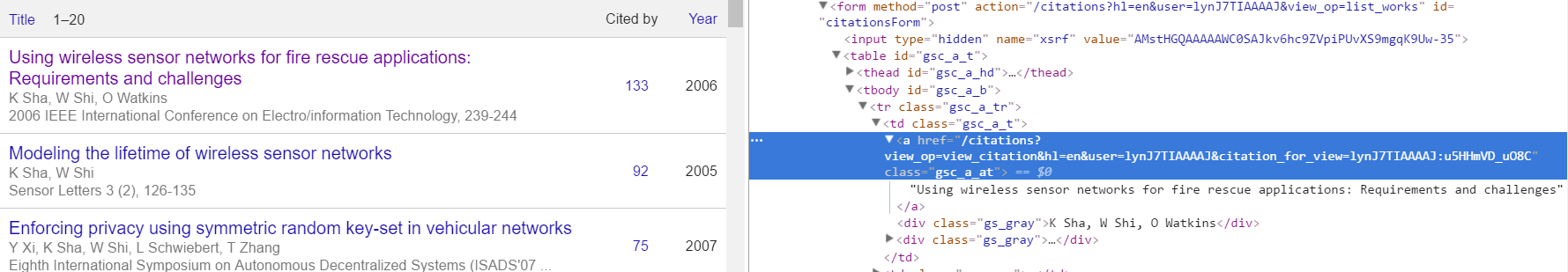




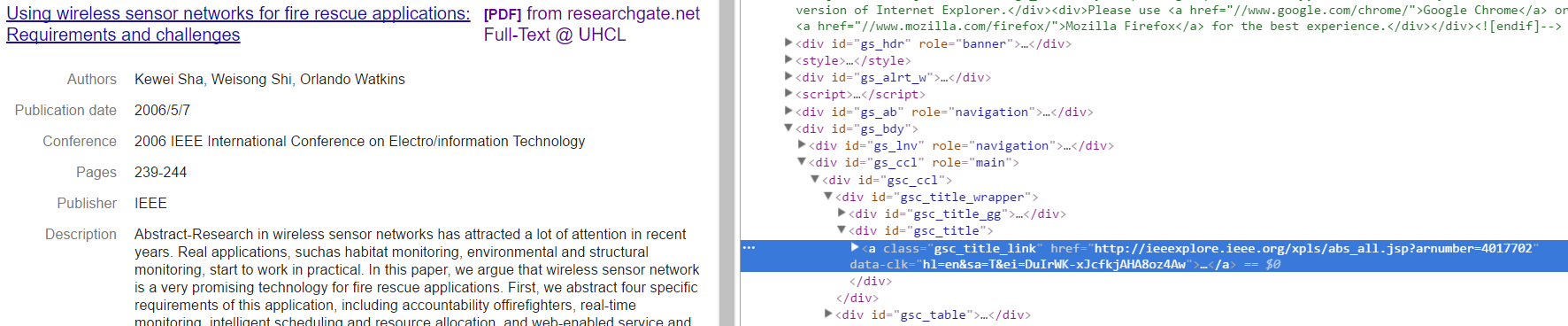


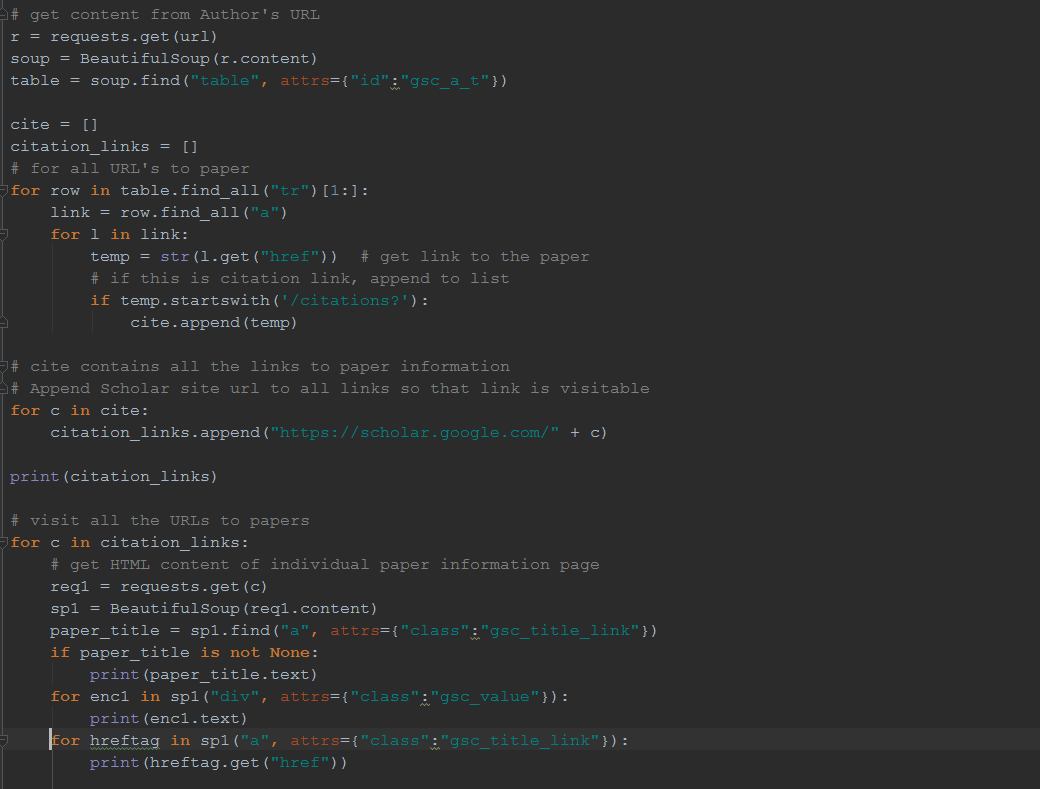
1. Research Paper: To extract each paper’s information (Title, Author names, Publication date, conference, publisher, description), following code is designed

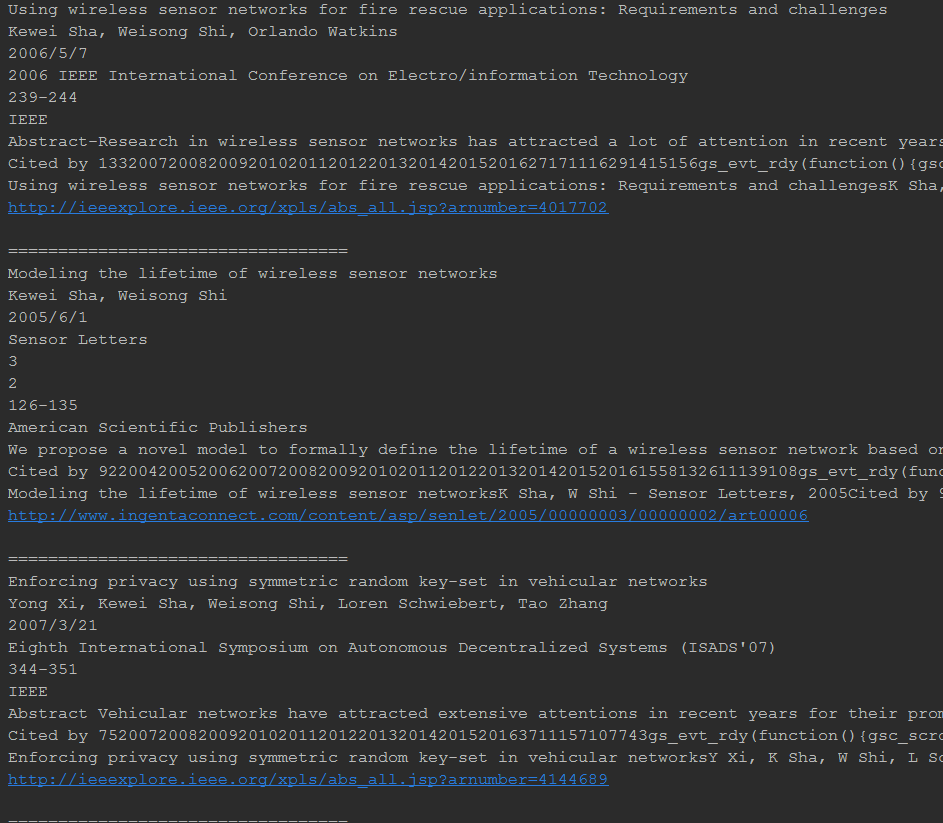






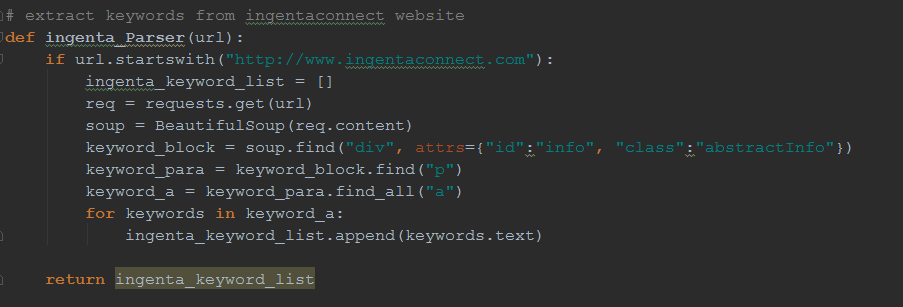




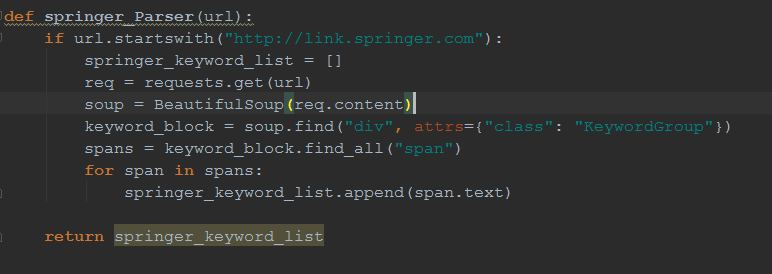


1. Also, we have designed parsers from various websites to extract keywords. We wrote code for websites like ingentaconnect.com, link, springer.com, inderscienceonline.com, content.iospress.com, IEEE, but there are some websites like sciencedirect.com which doesn’t allow bots to parse their websites.

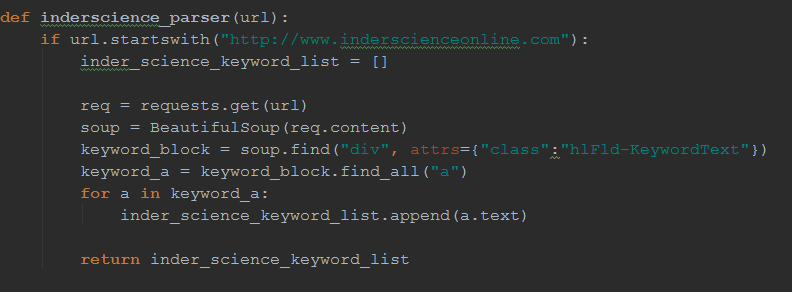
Ingentaconnect parser:



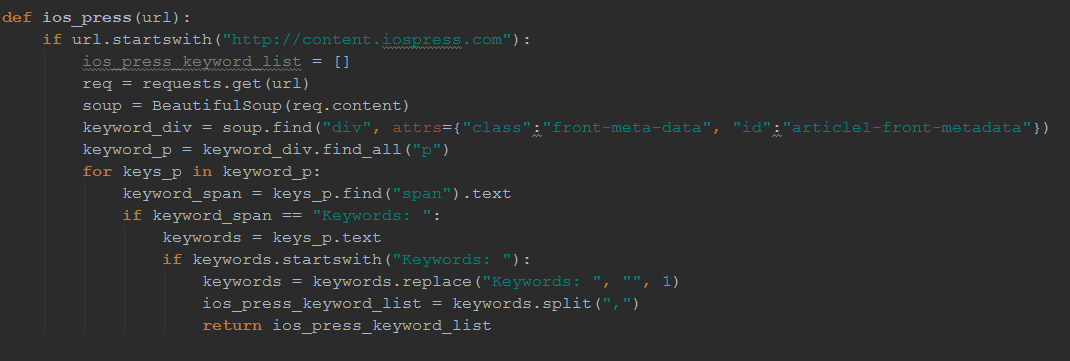
Springer site:



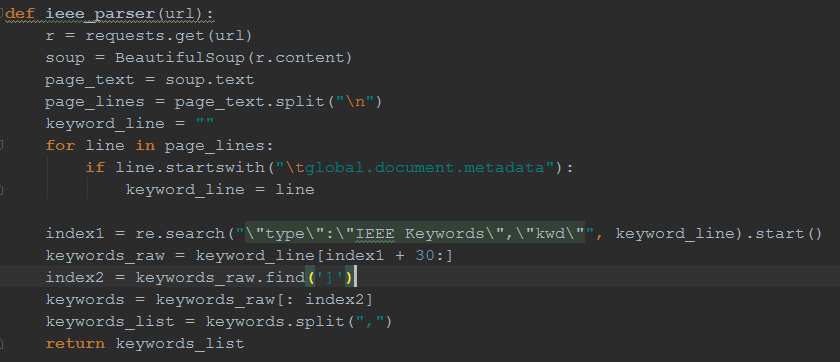
Inderscience parser



Ios press parser code



IEEE parser code



Problem with this approach:

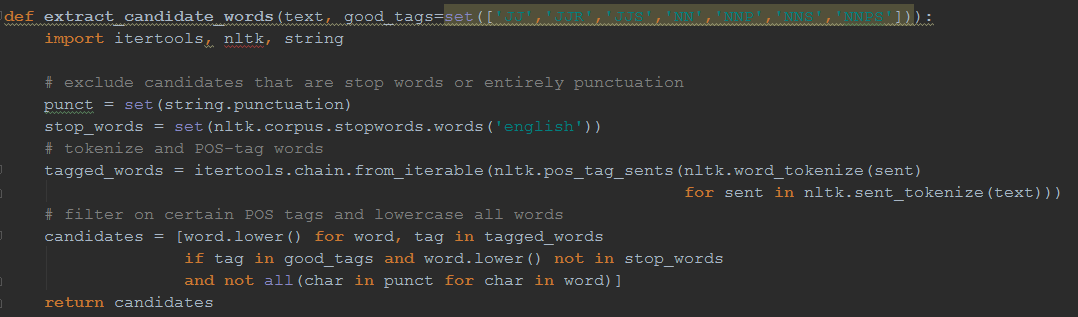
Today, we have thousands of Research publishers and websites where research papers are hosted. It is impossible to study HTML structure of all websites and then write a scraper to extract keywords from them. Therefore, we came with another approach: extraction of keywords from Research paper’s title, abstract and description.

For this approach, we used python’s NLTK (Natural language tool kit) to extract keywords and then further refine them. We take our Research paper’s description from Google Scholar as input. For this, we set our preferred tags as JJ (adjective), JJR (adjective, comparative), JJS (adjective, superlative), NN (Noun, singular or mass), NNP (Proper Noun, Singular), NNS (Noun Plural), NNPS (Proper Noun, plural).

Algorithm:

* Input: Paper’s title, abstract, description
* Remove all punctuations from text
* Remove all stop words from the text
* Tokenize the whole text
* Return all remaining words after refining

Code for the above Algorithm:



Issues faced with this parser:

The keywords generated by this method are not matching up with any of the keywords from ACM taxonomy file or from the research Interest of the Author.

Example: Below is the research paper of Author – Shih-Fu Chang which redirects to the website for which we haven’t written parser. So, the keywords are extracted from paper’s abstract and description.



Keywords generated:

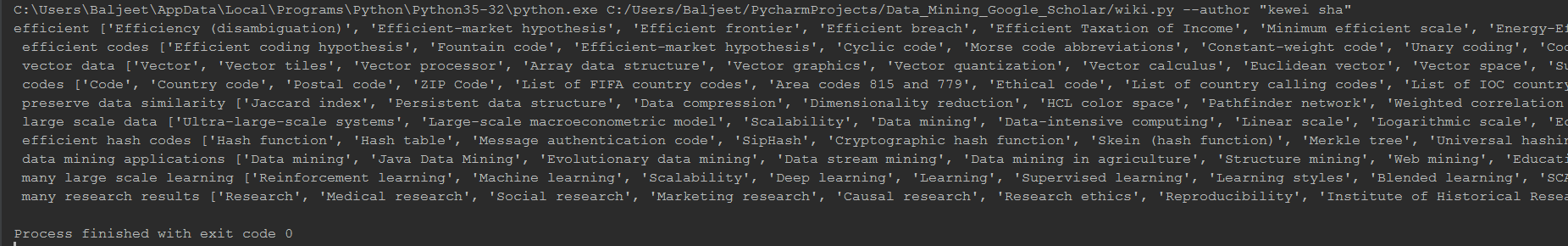
efficient, efficient codes, vector data, codes, preserve data similarity, large scale data, efficient hash codes, data mining applications, many large scale learning, many research results.

In the above example, keywords generated are not very accurate. When we did the matching (string alignment ratios) of these keywords with the ACM taxonomy file, the results were not very good. So we have to come with another approach.

Search made via Wikipedia:

One workaround for the problem we are facing is making keyword search on Wikipedia. We made search of all the keywords generated from the abstract and title. The result is as below:

Screenshot:



Text Output:

Format: keyword[keyword output from wikipedia]

efficient ['Efficiency (disambiguation)', 'Efficient-market hypothesis', 'Efficient frontier', 'Efficient breach', 'Efficient Taxation of Income', 'Minimum efficient scale', 'Energy-Efficient Ethernet', 'Efficient estimator', 'NBL Most Efficient Player Award', 'Ultra-Efficient Products Invader']

efficient codes ['Efficient coding hypothesis', 'Fountain code', 'Efficient-market hypothesis', 'Cyclic code', 'Morse code abbreviations', 'Constant-weight code', 'Unary coding', 'Coding theory', 'Building Codes Assistance Project', 'ZIP Code']

vector data ['Vector tiles', 'Vector', 'Vector processor', 'Array data structure', 'Vector graphics', 'Vector quantization', 'Vector calculus', 'Euclidean vector', 'Vector space', 'Support vector machine']

codes ['Code', 'Country code', 'Postal code', 'ZIP Code', 'List of FIFA country codes', 'Area codes 815 and 779', 'Ethical code', 'List of country calling codes', 'List of IOC country codes', 'Civil code']

preserve data similarity ['Jaccard index', 'Persistent data structure', 'Data compression', 'Dimensionality reduction', 'Pathfinder network', 'HCL color space', 'Weighted correlation network analysis', 'Collaborative filtering', 'Image registration', 'Topological data analysis']

large scale data ['Ultra-large-scale systems', 'Large-scale macroeconometric model', 'Scalability', 'Data mining', 'Data-intensive computing', 'Linear scale', 'Logarithmic scale', 'Economies of scale', 'Data center', 'Quality of Data (QoD)']

efficient hash codes ['Hash function', 'Hash table', 'Message authentication code', 'SipHash', 'Cryptographic hash function', 'Skein (hash function)', 'Merkle tree', 'Universal hashing', 'Distributed hash table', 'Hash calendar']

data mining applications ['Data mining', 'Java Data Mining', 'Evolutionary data mining', 'Data stream mining', 'Data mining in agriculture', 'Structure mining', 'Web mining', 'Educational data mining', 'Dynamic Data Driven Applications Systems', 'Examples of data mining']

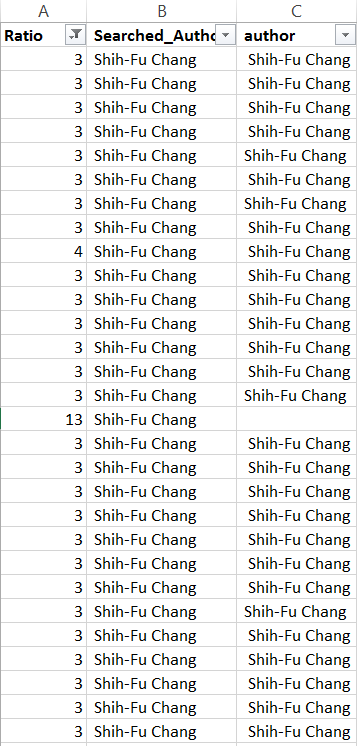
many large scale learning ['Reinforcement learning', 'Machine learning', 'Scalability', 'Deep learning', 'Learning', 'Supervised learning', 'Learning styles', 'Blended learning', 'SCALE-UP', 'Learning disability']

many research results ['Research', 'Medical research', 'Social research', 'Marketing research', 'Causal research', 'Research ethics', 'Reproducibility', 'Institute of Historical Research', 'Central Rice Research Institute', 'Securities research']

As we can see from the output, this output is not very accurate. These keywords generated from Wikipedia are not anywhere close to the keywords present in the ACM taxonomy file or the Author’s Research interest.

As we are not able to take into account those research papers which doesn’t have keywords, we tried simple k-means clustering on the basis of Authors and Co-Author names, which gave very good results. We didn’t use these for further classification because this clustering doesn’t involve any keyword related criteria.

Below is the result set of author Shih-Fu Chang based on Authors and co-author names:



This helped us to convert data into textual form, but not of much use because this doesn’t include keyword feature.

**Current issues and Future work:**

* We need some database or Ontology file which has all the computer science terminology. We need some database which has some tree like structure and all related terms are stored according to their hierarchy. For example, if we make a search across database like – “Artificial Intelligence”, it should yield in all related fields like “Machine Learning”, “Natural Language Processing” etc. We need some vast database which responds to every query of abbreviations as well, like if we make search for “IOT” or “NLP”, it should result in results for “Internet of Things” and “Natural Language Processing”. Currently, we are working with the ACM taxonomy file, which was published in 2012. This file doesn’t have new technologies included, like “IOT”. We have posted queries on different forums asking for library or database of Computer Science terms.
* We have made a general parser for keyword extraction from Research paper’s title, description and abstract. This parser need to be improved as this keywords generate some keywords which are not much accurate.