## How I crawled websites and Index

For this assignment, I am using Nutch 1.15 and Solr 7.50 for crawling and indexing. Here are the two websites I put in seed.txt:

*https://www.igdb.com/discover*

*https://www.igdb.com/advanced\_search*

Also, I am using “crawl” for automatic crawling instead of manually input commands. Here is the command I used:

*bin/crawl -i -D solr.server.url=http://localhost:8983/solr/nutch -s urls/seed.txt GameCrawler 500*

This command does all generate/parse/fetch/update/index/deduplication and loop it for 500 times. However, my disk of VM was full after three days crawling, the total number of documents I crawled is:

*Num Docs: 37781*

Below is a template of the document I crawled.

*"date":["2018-10-10T21:33:42.039Z"],*

*"description":["Enter the Golden Age of Piracy as you sail through rich merchant trade routes. Embody an insatiable pirate captain, dive into the battle for the ruling...",*

*"Enter the Golden Age of Piracy as you sail through rich merchant trade routes. Embody an insatiable pirate captain, dive into the battle for the ruling..."],*

*"title":["Skull & Bones"],*

*"type":["text/html",*

*"text",*

*"html"],*

*"url":["https://m.igdb.com/games/skull-and-bones"],*

*"content":["Skull & Bones\nToggle navigation\nRegister\nSign In\nHome\nCareers\nDiscover chevron\_right\nGames\nReviews\nNews\nComing Soon\nRecently Released\nTop 100\nAbout IGDB\nContribute chevron\_right\nAdd game\nContact\nSettings chevron\_right\nRequest desktop site\nSkull & Bones\n2018\nFollow\nWebsites\nAbout\nAdd To\nEnter the Golden Age of Piracy as you sail through rich merchant trade routes. Embody an insatiable pirate captain, dive into the battle for the ruling of the Indian Ocean and beco... Read More\nOfficial Website Wikipedia page Facebook page Twitter profile Twitch channel\n53\nHypes\nplay\_circle\_filled\nE3 2018 Cinematic Trailer\nplay\_circle\_filled\nE3 2018 Gameplay Trailer\nplay\_circle\_filled\nE3 2017: Cinematic Trailer\nplay\_circle\_filled\nE3 2017 Gameplay Trailer\nplay\_circle\_filled\nE3 2017 - What you need to know\nDetailed Information\nkeyboard\_arrow\_down\nRelease Dates\nDec 31, 2018\nDec 31, 2018 - PlayStation 4\nDec 31, 2018\nDec 31, 2018 - Xbox One\nDec 31, 2018\nDec 31, 2018 - PC (Microsoft Windows)\nDevelopers\nUbisoft Singapore\nPublishers\nUbisoft , Ubisoft Entertainment\nGame Mode\nCo-operative , Multiplayer , Single player\nGenre\nAdventure\nThemes\nAction , Warfare\nRecommendations\nOmensight\nShadow of the Tomb Raider\nTunic\nCode Vein\nDude Simulator\nTanzia\nCitadel: Forged With Fire\nUntitled Danganronpa Game\nAnthem\nThe Elder Scrolls VI\n© 2018 - 8 Dudes in a Garage AB ALL RIGHTS RESERVED\n"],*

*"tstamp":["2018-10-10T21:33:42.039Z"],*

*"digest":["55af158751eb14cadfad1d41dbf7964f"],*

*"host":["m.igdb.com"],*

*"boost":[0.010369517],*

*"id":"https://m.igdb.com/games/skull-and-bones",*

*"lang":["en"],*

*"\_version\_":1613975965235412992},*

The local database is stored in “nutch/data” file, and the index file is stored in “solr/server/nutch/data”. For rest of coding work, I import the whole data file to my project main directory.

## How I use Lucene and Solr to implement the search engine

* For reading the Solr index file, I am using “FSDirectory” and “IndexReader” to read the index files
* Then I used “StandardAnalyzer” to parse the keyword from all the index file. The default field of website parsed is “content”.
* However, the searching result shows it doesn’t return a good website order. So I coded it differently to boost different fields. For example, I doubled the weight of “title” field compare to content field.
* I used Scene Builder and Javafx to implement the GUI. It have user-friendly navigation system and it is very similar to “Google”.
* It has two additional function, returning similar words and return interested field.
* For returning similar word, the algorithm is borrowed from <http://rosettacode.org/wiki/Levenshtein_distance#Java>

Which is based on

<http://www.codeproject.com/Articles/13525/Fast-memory-efficient-Levenshtein-algorithm>

This function can advise user with correct spelling when user has a typo in their input.

* The domain I crawled ([*www.igdb.com*](http://www.igdb.com)*)*  has many fields such as “User”, “Game”, “Platform” and so on. “Returning interested field” function can make user search specific field and avoid search results from other field.