

JAKE MASON
256602.

Final Viva

List of contents

- Project Description
- Aims + Objectives
- Literature Review
 - What currently exists
 - Issues
 - Coding
- User Reception/Feedback
- Results
- Demo
- Unique code snippets
 - Unique search results
 - Result sorting system
- Conclusion
- References

Project Description

- ❖ Address shortcomings of scholarly search engines
- ❖ Relevant content
- ❖ Access to information
- ❖ Web based application
- ❖ Advanced filters
- ❖ Unique repositories
- ❖ More relevant results

Aims



INVESTIGATION



CREATION

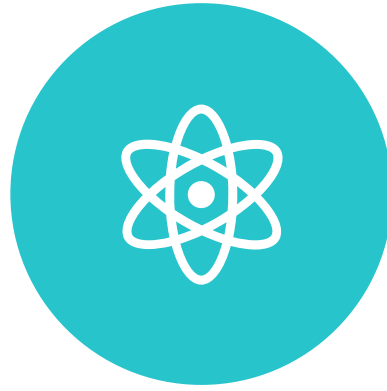


TESTING

Aim 1, Objectives.



CHOOSE SEARCH ENGINES TO
INVESTIGATE



CHOOSE SPECIFIC
TERMS/SUBJECT FOR TESTING



RECORD TIME IT TAKES TO
FIND RELEVANT INFO

Aim 2, Objectives



CHOOSE APPROPRIATE
IDE



GET SYSTEM ONLINE



EASE OF
USE/PRESENTABILITY

Aim 3, Objectives



SUITS ITS INTENDED
PURPOSE.



SAVES TIME.



TESTED.



FEEDBACK USED.



Literature Review: What currently exists

Google Scholar

- ❖ Most popular scholarly search engine.
- ❖ Large archive, lacks quality control [1] [2]
- ❖ Algorithm ranked by total citations, not relevance. [3]
- ❖ Sources used come from German, North and Latin American universities.

Literature Review: What currently exists

Microsoft Academic Search

- ❖ 27.2 million records in archive. [4]
- ❖ Regarded as slow. [5]
- ❖ Not taken seriously by academics [5] [6] [7]
- ❖ Sources – professors at MIT, Latin American web developers

Literature Review: What currently exists

RefSeek [8]

- ❖ Can include non-commercial regular websites.
- ❖ Algorithm allows for more results than competition
- ❖ However, archive is still small.
- ❖ Primary source being a journal entry for Chitkara College

Literature Review: Issues

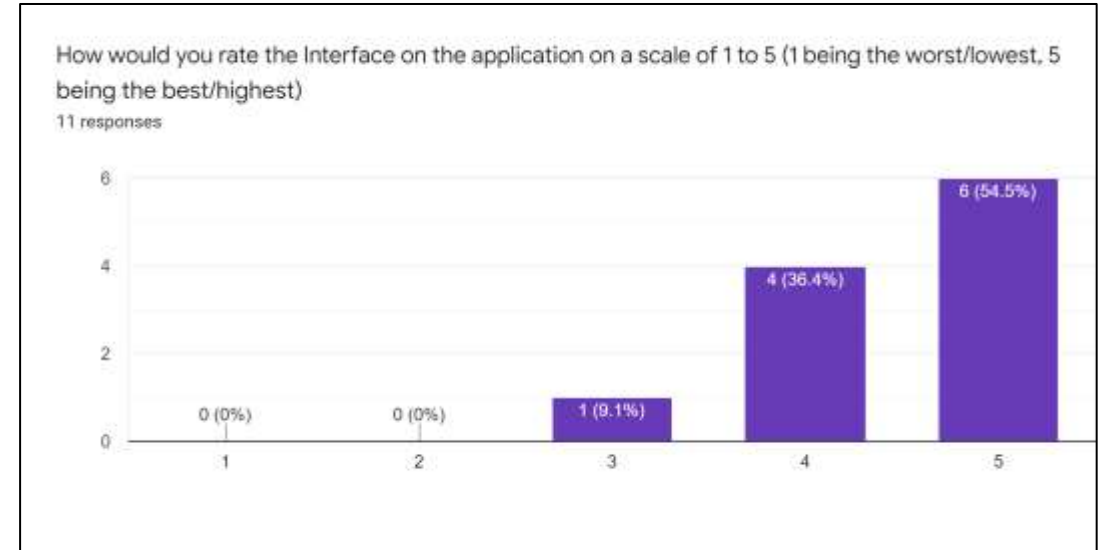
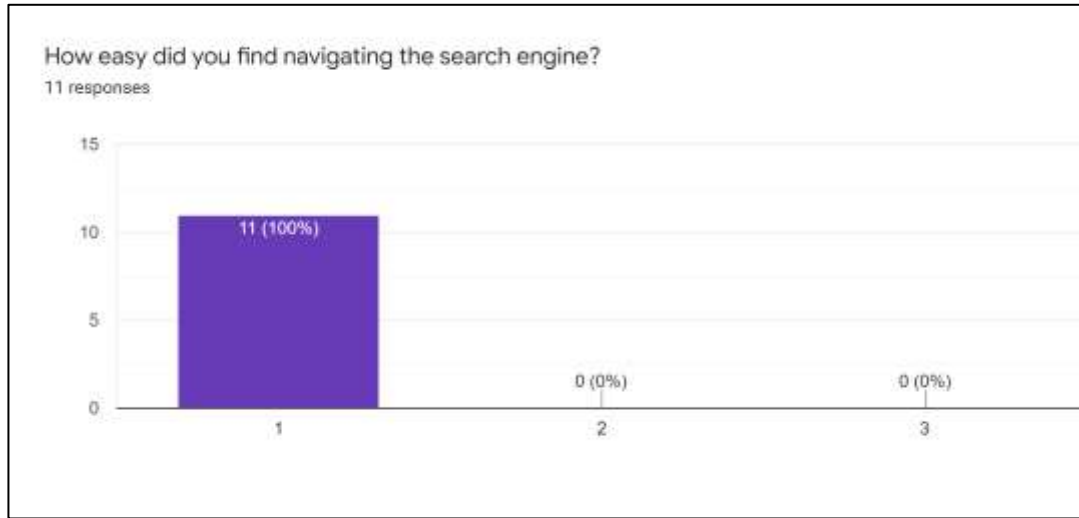
- ❖ Inaccessibility. [9]
- ❖ Paywalls
- ❖ Page unavailable
- ❖ Relevant information [10]
- ❖ Algorithm prioritising citations over relevance
- ❖ Sources: Yale University Library, Yale University, Gary Price (White House Speaker)

Literature Review: Coding - Repositories

- ❖ Unique Search Results per engine [11] (Spanish National Research Council)
- ❖ Tutorial provided by Bielefeld University [12]
- ❖ XML Code [13]
- ❖ Tutorial provided by DSpace
- ❖ DSpace former repository tool of Google

Literature Review: Coding – Inaccessible results

- ❖ “Automated processes for indexing” leads to inaccessibility [14] [15]
- ❖ Index of google based on if site as a home page [16]
- ❖ DSpace indexes just documents [17] [18]
- ❖ Allowing indexing of only available documents



User Reception: Interface

How long did it take for you to find necessary information? (If your answer is an estimation, please state so).

11 responses

2 minutes

4 minutes

9 minutes

less than a minute

1 < minute

7 minutes

11 minutes

14 minutes

around 10 to 20 minutes?

How long did it take for you to find necessary information? (If your answer is an estimation, please state so).

20 responses

Over an hour

23 minutes

38 minutes i think, this is a previous search tho and im just guessing how long it took

Dont know, around 40 minutes maybe. It was a good read.

7-10 minutes i think, estimation /shrug

22 minutes.

an hour

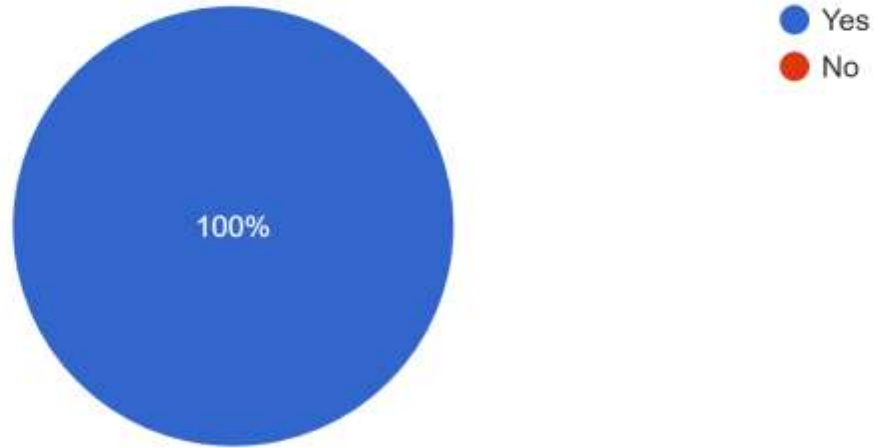
2-3 mins

6m

User Reception: Time Reduction

Do you feel as if you managed to get your tasks done in a timely manner?

11 responses



User Reception: Time Reduction

Results

Reduced research time.

Allowed adding items to favourites.

Search History.

Satisfied test users.

Sorting and repositories work.

Demonstration (History, Favourites, Search, Sorting).



Code Snippets: Unique results

```
*collection
<collection name=csl/collection_name>
<list_repository>
  <repository>
    <name>Zurich Open Platform</name>
    <name_en>Zurich Open Platform</name_en>
    <activation_date>2022-03-10</activation_date>
    <internal_name>ftunizuerichrub</internal_name>
  </repository>
  <repository>
    <name>Közszolgálati Tudásportál Nemzeti Közszerkeleti Egyetem</name>
    <name_en>Knowledge Portal University of Public Service</name_en>
    <activation_date>2022-03-10</activation_date>
    <internal_name>ftunivkdc</internal_name>
  </repository>
  <repository>
    <name>IAUJA: Repositorio de Trabajos Académicos de la Universidad de Jaén</name>
    <name_en>IAUJA: Repositorio de Trabajos Académicos de la Universidad de Jaén</name_en>
    <activation_date>2022-03-10</activation_date>
    <internal_name>ftunivjaentaufa</internal_name>
  </repository>
  <repository>
    <name_en>Scientia. Digital Information Deposit of the Department of Health</name_en>
    <activation_date>2022-03-10</activation_date>
    <name>Scientia. Dipòsit d'Informació Digital del Departament de Salut</name>
    <internal_name>ftscientia</internal_name>
  </repository>
  <repository>
    <name>Linguistic Data and NLP Tools CLARIN</name>
    <activation_date>2022-03-10</activation_date>
    <name_en>Linguistic Data and NLP Tools CLARIN</name_en>
    <internal_name>ftclarinit</internal_name>
  </repository>
  <repository>
    <name>documenta journal</name>
    <name_en>documenta journal</name_en>
    <activation_date>2022-03-10</activation_date>
    <internal_name>ftdocumenta</internal_name>
  </repository>
  <repository>
    <name>Friedrich plus Zeppelin Universität</name>
    <name_en>Friedrich plus Zeppelin Universität</name_en>
    <activation_date>2022-03-09</activation_date>
    <internal_name>ftfriedrichshf</internal_name>
  </repository>
  <repository>
    <activation_date>2022-03-09</activation_date>
    <name_en>Akademia Morska w Szczecinie</name_en>
    <name>Akademia Morska w Szczecinie</name>
    <internal_name>ftamszczecin</internal_name>
  </repository>
  <repository>
    <activation_date>2022-03-09</activation_date>
    <name_en>Marek-Maturo Digital Library</name_en>
    <name>Marek-Maturo Biblioteka Cyfrowa</name>
    <internal_name>ftmabcylartyn</internal_name>
  </repository>
</list_repository>
```

```
*result name="response" maxFound="2" start="0" maxScore="2.4071801"
<doc>
  <date name="dchdate">2022-02-08T22:29:26Z</date>
  <str name="docid">C19668010e21e37ce99aa75d6d0d423b26d535c2b51242c4ac1e95fa729691</str>
  <str name="document">csu</str>
  <str name="dcountry">de</str>
  <str name="dcollection">ftublibpub</str>
  <str name="dperson">PUB - Publikationen an der Universität Bielefeld</str>
  <str name="dtitle">Search engine technology and digital libraries - Moving from theory to practice</str>
  <arr name="dcreator">
    <str>Somann, Friedrich</str>
    <str>Lossau, Norbert</str>
  </arr>
  <arr name="dperson">
    <str>Somann, Friedrich</str>
    <str>Lossau, Norbert</str>
  </arr>
  <arr name="dsubject">
    <str>Elektronische Bibliothek</str>
    <str>Suchmaschinen</str>
    <str>Information Retrieval</str>
    <str>Bielefeld Academic Search Engine</str>
    <str>ddc:820</str>
  </arr>
  <str name="ddescription">Somann F., Lossau N. Search engine technology and digital libraries - Moving from theory to practice. D-lib magazine . 2004;10(9).</str>
  <arr name="dpublisher">
    <str>CMI Acct</str>
  </arr>
  <str name="ddate">2004</str>
  <str name="dyear">2004</str>
  <arr name="dtype">
    <str>http://purl.org/coar/resource/type/t_0501</str>
    <str>infoeu-repo/semantics/article</str>
    <str>doc-type:article</str>
  </arr>
  <arr name="dtypenorm">
    <str>121</str>
  </arr>
  <arr name="didentifiers">
    <str>https://nbn-resolving.org/urn:nbn:de:hbz-pub-16809024</str>
    <str>https://pub.uni-bielefeld.de/record/1680902</str>
    <str>https://pub.uni-bielefeld.de/download/1680902/2315314</str>
  </arr>
  <str name="dlink">https://nbn-resolving.org/urn:nbn:de:hbz-pub-16809024</str>
  <arr name="dlanguage">
    <str>eng</str>
  </arr>
  <arr name="durlpattern">
    <str>infoeu-repo/semantics/altIdentifier/dni/10.1045/september2004-lossau</str>
    <str>infoeu-repo/semantics/altIdentifier/issn/1082-9873</str>
    <str>https://nbn-resolving.org/urn:nbn:de:hbz-pub-16809024</str>
    <str>https://pub.uni-bielefeld.de/record/1680902</str>
    <str>https://pub.uni-bielefeld.de/download/1680902/2315314</str>
  </arr>
  <str name="drights">infoeu-repo/semantics/openaccess ; https://rightsstatements.org/vocab/InC/1.0/</str>
  <arr name="dauthorid">
    <str>Somann, Friedrich | orcid:0000-0002-8297-3548</str>
  </arr>
```

Code Snippets: Sort system

```
<noscript>
  <select name="sort" onchange="this.form.submit();" id="SortBy" class="form-control input-sm" aria-label="Choose a Sort Order">
    <option value="score desc,dctitle_sort asc" selected="selected">Relevance</option>
    <option value="dccreator_sort asc,id asc">Author, A-Z</option>
    <option value="dccreator_sort desc,id asc">Author, Z-A</option>
    <option value="dctitle_sort asc,id asc">Title, A-Z</option>
    <option value="dctitle_sort desc,id asc">Title, Z-A</option>
    <option value="dctype_sort desc,id asc">Date of publication, descending</option>
    <option value="dctype_sort asc,id asc">Date of publication, ascending</option>
  </select>
  <noscript><button type="submit" class="btn btn-default input-sm margin-top0-5 pull-right">Go</button></noscript>
```

Code Snippet: Algorithm

❖ JavaScript

❖ Influenced by DuckDuckBot

❖ Rank based on relevance of keywords

❖ Ranks importance of “nodes”

```
module.exports = function (nodeMatrix, linkProb, tolerance, callback, debug) {  
    if (!nodeMatrix || !linkProb || !tolerance || !callback) {  
        throw new Error("Provide 4 arguments: "+  
            "nodeMatrix, link probability, tolerance, callback")  
    }  
  
    if (!debug) debug=false  
    return new cusAlg(nodeMatrix, linkProb, tolerance, callback, debug)  
}  
  
function cusAlg (nodeMatrix, linkProb, tolerance, callback, debug) {  
    this.outgoingNodes = nodeMatrix  
    this.linkProb = linkProb  
    this.tolerance = tolerance  
    this.callback = callback  
    this.pageCount = this.outgoingNodes.length  
    this.coeff = (1-linkProb)/this.pageCount  
    this.probabilityNodes = []  
    this.incomingNodes = []  
    this.debug=debug  
  
    this.startRanking()  
}  
  
cusAlg.prototype.startRanking = function () {  
    var initialProbability = 1/this.pageCount  
    , outgoingNodes = this.outgoingNodes  
    for (i in outgoingNodes) {  
        this.probabilityNodes.push(initialProbability)  
        for (a in outgoingNodes[i]) {  
            var index = outgoingNodes[i][a]  
            if (!this.incomingNodes[index]) this.incomingNodes[index]=[]  
            this.incomingNodes[index].push(i)  
        }  
    }  
    if (this.debug) this.reportDebug(1)  
  
    this.iterate(1)  
}  
  
cusAlg.prototype.reportDebug = function (count) {  
    console.log("Pages: " + this.outgoingNodes.length)  
    console.log(this.incomingNodes)  
    console.log("___ITERATION "+count+"___")  
    console.log(this.probabilityNodes)  
}  
  
cusAlg.prototype.iterate = function (count) {  
    var result = []  
    for (b in this.probabilityNodes) {  
        var sum = 0  
        for (var a=0; a<this.incomingNodes[b].length; a++) {  
            prob = this.probabilityNodes[this.incomingNodes[b][a]]  
            ct = this.outgoingNodes[this.incomingNodes[b][a]].length  
            sum += (prob/ct)  
        }  
        var res = this.coeff+this.linkProb*sum  
        , max = this.probabilityNodes[b]+this.tolerance  
        , min = this.probabilityNodes[b]-this.tolerance  
        if (min <= res <= max) result.push(res)  
        this.probabilityNodes[b]=res  
    }  
    if (result.length == this.pageCount)  
        return this.callback(null, result)  
    if (this.debug) this.reportDebug(count)  
  
    this.iterate(++count)  
}
```

Code Snippet: Algorithm - Initialize

- ❖ Outgoing nodes - represents each node with outgoing links
- ❖ LinkProb - represents values
- ❖ Tolerance – Lower values are faster, higher values more accurate

```
function cusAlg (nodeMatrix, linkProb, tolerance, callback, debug) {  
    this.outgoingNodes = nodeMatrix  
    this.linkProb = linkProb  
    this.tolerance = tolerance  
    this.callback = callback  
    this.pageCount = this.outgoingNodes.length  
    this.probabilityNodes = []  
    this.incomingNodes = []  
    this.debug=debug  
  
    this.startRanking()  
}
```

Code Snippet: Algorithm - Ranking

- ❖ Probabilities initialised
- ❖ Re-array the graph, generate initial probability
- ❖ Print each iteration

```
cusAlg.prototype.startRanking = function () {  
    var initialProbabilty = 1/this.pageCount  
        , outgoingNodes = this.outgoingNodes  
    for (i in outgoingNodes) {  
        this.probabilityNodes.push(initialProbabilty)  
        for (a in outgoingNodes[i]) {  
            var index = outgoingNodes[i][a]  
            if (!this.incomingNodes[index]) this.incomingNodes[index]=[]  
            this.incomingNodes[index].push(i)  
        }  
    }  
    if (this.debug) this.reportDebug(1)  
  
    this.iterate(1)  
}
```

Code Snippet: Algorithm – Log to console

```
cusAlg.prototype.reportDebug = function (count) {  
    console.log("Pages: " + this.outgoingNodes.length)  
    console.log(this.incomingNodes)  
    console.log("____ITERATION "+count+"____")  
    console.log(this.probabilityNodes)  
}
```

Code Snippet: Algorithm – Calculate

- ❖ Weight divided by outgoing edges
- ❖ Determine new probability within tolerance
- ❖ Push result if it has changed
- ❖ Update probability (node b)
- ❖ Return via callback when done
- ❖ Print iteration

```
cusAlg.prototype.iterate = function (count) {  
    var result = []  
    for (b in this.probabilityNodes) {  
        var sum = 0  
        for (var a=0; a<this.incomingNodes[b].length; a++) {  
            prob = this.probabilityNodes[this.incomingNodes[b][a]]  
            ct = this.outgoingNodes[this.incomingNodes[b][a]].length  
            sum += (prob/ct)  
        }  
        var res = this.coeff+this.linkProb*sum  
            , max = this.probabilityNodes[b]+this.tolerance  
            , min = this.probabilityNodes[b]-this.tolerance  
        if (min <= res <= max) result.push(res)  
        this.probabilityNodes[b]=res  
    }  
    if (result.length == this.pageCount)  
        return this.callback(null, result)  
    if (this.debug) this.reportDebug(count)  
    this.iterate(++count)  
}
```

Conclusions and Recommendations

Most issues found in research were resolved

Interface was satisfactory but could be improved.

Too much focus on certain areas

More planning recommended

Better aims and objectives

Less focus on primary research

Any questions?
