2019

Curtus: An NLP Tool to Map Job Skills to Academic Courses

Daniel Rockwell

CURTUS: AN NLP TOOL TO MAP JOB SKILLS TO ACADEMIC COURSES

Daniel Rockwell
2019

COLUMBUS STATE UNIVERSITY

The Graduate Program in Applied Computer Science

Curtus: An NLP Tool to Map Job Skills to Academic Courses

A THESIS SUBMITTED TO

TSYS SCHOOL OF COMPUTER SCIENCE

IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE

BY

Daniel Rockwell

Columbus, Georgia

2019

Curtus: An NLP Tool to Map Job Skills to Academic Courses

BY

DANIEL ROCKWELL

Committee Chair:

Shamim Khan

Committee Members:

Kyongseon Jeon

Rania Hodhod

Columbus State University

2019

Abstract

Many businesses are burdened with the need to train students for the job instead of finding them prepared for it. Few business leaders feel that colleges prepare students for future jobs from day one. It can be a challenge for colleges to determine if their curricula meet the industry needs. Mapping industry needs to academic courses can be advantageous to both parties as it will allow colleges to be aligned with the industry needs and accordingly satisfy those needs and will allow the industry to hire better prepared graduates. In an attempt to address this, a system prototype that uses a collection of job descriptions from various sites and syllabi of college courses as the input knowledge was developed. The primary goal of the system is to help students to find courses that would be most beneficial in providing them with the skills that match a given job description. The secondary goal is to help faculty to quickly find out information about current skills and tools covered in the existing courses, which accordingly can help them to make decisions about their future courses to satisfy the inductry needs. The system was developed using the Natural Language Toolkit (NLTK) and the Python programming language. Two sets of keywords were used to test the system; the first one is the most common keywords and the second one includes the most and least common keywords. Results from testing the system demonstrate that using the former set of keywords allowed for better results with precision equal to 55% and recall equal to 39.61%.

Keywords: Natural Language Processing; Artificial Intelligence; Natural Language Toolkit; Lemmatization; Course Development; Text Matching

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to provide my thanks to Dr. Shamim Khan for the guidance and help that he provided on this project as my Thesis Advisor. I would also like to thank Dr. Rania Hodhod for providing me assistance in identifying missing sections and making final changes. My gratitude also goes out to my family for providing me with the financial support that has allowed me to focus on my education. Lastly, I would like to give my sincerest thanks to the faculty and staff of the TSYS School of Computer Science for the education that they have provided me in all of my years at Columbus State University.

Chapter 1: Introduction

1.1 Problem Statement

One difficulty that colleges face today is the competition with online platforms that offer free tutorials and courses. A person can go and look on the internet and find a tutorial or a book that will teach them a certain skill without the need for attending a college. That brings up the major issue that is how can the college make a difference and help students better prepare for the workplace. Employers have begun to remove the requirement for college on their job description due to the skills that can be learned or taught outside the college setting. Employers even argue that college education is not preparing students well enough for the workplace no matter how much the students think it is. Jaschik does a good job of further explaining this in his article [1]. Although this issue is widespread and growing with time, this project provides a tool to assist in this problem.

1.2 Goals

The primary goal was to build a system to be used by students. The system, named Curtus, would provide students with a ranked list of classes to take based on a job description. Curtus uses natural language processing to:

1. help students pick the right classes that they need for a particular job;
2. help colleges set up their courses in the best way possible for students to be ready for jobs;

3. help employers know just how well the college is preparing students for their needs.

A student should be able to plug in a job description and find classes that will directly teach them what they need for that job. If the job requires some knowledge in a programming language like C#, it should recommend them classes that teach C# as well as other as other courses that practice the language. Employers can use that same tool to know about available courses at a college and if they are preparing students under their particular job category.

1.3 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field that deals with understanding and interpretation of the human language by computers. NLP falls into a category of artificial intelligence (AI) and has a variety of applications. Today NLP is seen everywhere inside of smartphones, game systems, or even in some household appliances [2]. One of its first big appearances was back in the 1950s when it was used to translate Russian text into English text [3]. At the time however it was not overly successful and ten years of added research on the topic did not yield much progress. Advancements in the field of AI and Machine Learning has led to the reemerged of the field into what we see now [3]. The breadth of application that NLP has now is astounding [4]. Some of the applications of NLP include: Sentiment Analysis [5], Text Summarization [6], Information Extraction [7], Topic Segmentation [6], Question Answering [8], Part of Speech Tagging (POS) [9], Parsing [7],Translation [10], and Argumentation Mining [11].

The most common places that people find NLP useful is in autofill [4] whether it is from a web browser or typing a text or giving voice commands to a computer. Small tasks such as finding misspelled words or incorrect use of words based on the context in a document can also take advantage of NLP.

1.4 NLP and Text Matching

Text Matching which is a term that describes finding how much one text matches another text. It is commonly used in searching for web pages. Matching is done at three different levels being word matching, phrase matching, and sentence matching. Curtus uses word matching to find classes that contain the most skills that a job is looking for.

1.5 Contribution

A final version of Curtus would be able to add a new level of comparison to the area of text matching. Most of the research done in the area of text matching focus on how similar two texts are and the different methods that can be achieved (This is further explained in Chapter 2). For being able to recommend classes that are most relevant to a job, there is one more level of detail that is needed for the matching. Unlike a search engine where words and phrases are typed in manually, Curtus uses job descriptions as its knowledge base. As much of text matching research focus on the searching portion, this project's focus is on knowing the importance of each of the inputs before the text matching. Curtus needs to be able to decide which keywords are the most important and then provide weighted values to those words.

1.6 Challenges

Throughout the project, there were multiple barriers that were faced, and challenges met. One of the challenges was the amount of time spent creating human-traced results to be able to evaluate the system. With over 100 syllabi and six job descriptions, each was hand traced on paper and compared. Another challenge was the fact that the used syllabi do not have the same file type or extension. Extra unforeseen work was done to get all of the information into one shared style while retaining the flexibility to easily add more courses to Curtus.

While implementing the system, it was a challenge to be able to assemble the system in such a way that it could be quickly tested and debugged. A custom shell ended up being made to run commands to import and export files, work with different test sets, view underlying data, and modify the current session.

The final challenge faced for this project was to decide on a good way to properly display the results of the system; finding a good way to display/visualize hundreds of results in one time.

1.7 Thesis organization

This thesis is organized as follows: the next chapter focuses on literature review on related projects and curricula development. Chapter 3 focuses on the design of Curtus and how data is passed through it. Chapter 4 discusses in detail the implementation of the system and programming aspects. Chapter 5 goes over the evaluation methods as well as the test data used by the system. Lastly,

Chapter 6 discusses the results obtained from testing Curtus, conclusion and future work.

Chapter 2: Background

2.1 Other Work in NLP

There is a lot of research being done in NLP out there. Many works focused on the types of data that are extracted through text matching [12], and on using methodologies from other areas for text matching [13]. Other application_based papers showed the application for NLP and text matching in projects like recommending articles in scientific communities [14], and identifying matching citations from papers [15]. Two primary research papers acted as the basis for this research and prior to the implementation of Curtus were information retrieval [12] and text matching as image recognition [16].

The information retrieval paper provided great insight into the different types of information that can be extracted. One item in particular that they discuss was providing weights by how frequently they appear in the text; the system looks at a question or statement and based on the words provided, across a body of knowledge, it would rate the question or statement based on the frequency of how often a word appears. The phrase that has the highest score is given back as the answer. That concept of applying weights and score has been widely used across the field.

The second paper, text matching as image recognition, focused on text matching using concepts from image recognition to find patterns in text. This method can be used to find similar phrases and sentences without the need for both of them to have the same words but instead, share the same meaning and structure.

2.2 Curricula Development

In curricula development, NLP, as well as other forms of AI, have already assisted in many different ways. Natural language has been used in ranges from helping within a classroom to helping across multiple classrooms. One application known as Language Muse helps to assist teachers in building instruction and lesson plans for students that are learning English [10]. It used NLP to provide immediate feedback on their work. NLP was used for summarization and translation of English to Spanish. While this tool helps to restructure a class, Curtus focuses more on a wider range of courses as a whole within a college setting.

Work that has been done in the college setting under the same scope as Curtus was a knowledge map tool built for evaluating medical curricula documents [8]. The tool was used to be able to extract important words or phrases from the documents. This is highly similar to the final goal of Curtus that is to be able to identify the most important keywords from job descriptions. With the other project being similar to Curtus, particularly under the area of measurements, the medical curricula project did not provide any information of an acceptable error range that can gauge Curtus' evaluation.

Curtus aims to provide assistance at a higher level than the other two p. Similar to the tool used for Medical Curricula [8], Curtus extracts keywords from job descriptions that it considers the most important. The differences start to appear when Curtus has to give a value that can define how important a word is. These values are used to find courses whose syllabi are of most relevance to the

keywords. Using the values of the keywords and the frequencies in which they appear in the syllabi, Curtus would provide the courses that apply in a ranked order based on how important the skills are that they offer and how many skills they provide.

Developing these courses can be a major challenge as shown by the amount of work being done to aid in this process [10] [8]. Curtus is a tool that can be used in this area to provide students with classes that can prepare students for jobs. One primary example that it would be able to assist with is the Illinois Institute of Technology's attempt to increase the real-world aspect of their computer science program [17]. Colleges like Illinois Institute would be able to use this system to evaluate the added courses and make sure that they are providing them with the right skills.

Chapter 3: Curtus Architectural Design

3.1 System Design

A collection of course syllabi taught at the TSYS School of Computer Science at CSU and job descriptions from internet job posting sites were used to achieve the thesis goals. Each of the courses syllabi provides details describing the skills learned from the course and topics covered. A database was built using keywords collected from 102 syllabi stored under their original format. Job descriptions were collected from public job posting sites like Glassdoor and LinkedIn, but not necessarily stored in a database, and were used mainly for testing purposes. The full steps for this implementation are displayed in **Error! Reference source not found.**.

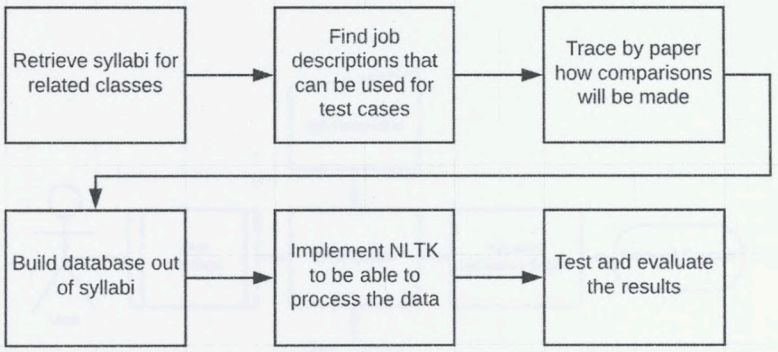


Figure 1: Steps in the project implementation

The implementation goes through the following steps...

1. Retrieve syllabi for related classes
2. Find job descriptions that can be used from test cases
3. Trace by paper how comparisons will be made
4. Build a database out of syllabi
5. Implement NLTK to be able to process the data

6. Test and evaluate the results

The system, however, is not made to evaluate a college program; it can just tell someone what classes they should take. The system is only able to provide recommendations based on the information it has so the classes it provides may be a good fit for a job description compared to the other classes. However, they might not be the best classes to take. It is up to the user to judge how good a course is or how well courses are set up for a certain job.

3.2 System Architecture

The system architecture comprises of four layers: the first layer has the user interface, the second layer has the preprocessor for data input, the third layer has a keywords extraction engine and finally, an output layer, see Figure 2.

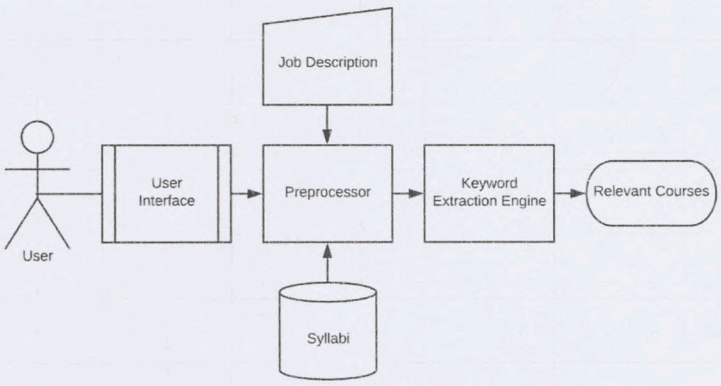


Figure 2: Curtus Architecture

The user interacts with the system through the text-based user interface where he can upload a job description file. That file is then preprocessed along with the syllabi that are already stored in the system. When they are preprocessed, Curtus takes the different file types and pulls out the text into local storage as

regular text. That text is then parsed through to remove any stop words and apply lemmatization to put all of the words in the text into their base form. Removing stop words assures that common words like 'and,' 'the,' and 'to' would not be considered as keywords. Lemmatization removes some strain on the system for making sure all words are in the same tense and sets them to singular. Words like mice and syllabi would be changed to mouse and syllabus. After the preprocessing phase, the keyword extraction engine extracts the important keywords and gives back the results on each syllabus. Those results are used to sort the syllabi in order based on what Curtus considers the most important (syllabi that contain the highest number of matches when compared to the job description are ranked higher). The ranked courses are given back to the user as output.

Chapter 4: System Development

4.1 Libraries Used

Many libraries were used in building this project. **Error! Reference source not found.** provides a list of the libraries used as well as the way in which they were used in order to give credit to them.

Table 1: Libraries Used

NLTK	**Used to be able to extract keywords from the text by frequency. It also provided tools to be able to filter out common words such as and, in, to, the, etc.**
Codecs	Used to be able to read in syllabi that are in HTML format and retrieve only the content from the files.
PyPDF2	Used to be able to read in syllabi that are in PDF format and retrieve only the content from the files.
Codecs	Used to be able to read in syllabi that are in PDF format intended for web pages and retrieves only the content from the file

4.2 Natural Language Toolkit

NLTK, Natural Language Toolkit, is a tool that was used on this project. After looking through multiple different tools available for natural language processing, NLTK came out to be the best fit and most well rounded of the tools that had been found. NLTK has many pros including:

- It is open source, so it is very easy to add onto as well as being well refined
- It has a book that gives detailed instruction on downloading, installing, and importing the library [18]
- It is widely used for research [19]

4.3 System Implementation

The system is implemented using the following steps:

- The algorithms for the keyword matching were written in such a way that keywords are sorted alphabetically using merge sort. Comparing the files is an O(n) time result vs. $O(n^2)$ as a result of the keywords being presorted. This code is shown and discussed in Section 4.4.
- Using a merge sort algorithm as shown under Appendix A, this assures that the program sorts the information with time complexity O(n log n).
- Different file types are covered to make the system as flexible as it can be. This code is displayed and discussed in Section 4.5.
- Nouns are parsed out and sorted in such an order of most frequent occurrence. Also, certain words of interest were noted as well. This code is shown and discussed in Section 4.6.

4.4 Comparing the sorted files

This algorithm works under the assumption that the two lists passed to it are sorted. As a result, it can go through in a clean sweep to find any matching pairs. Figure 3 shows a visual representation of the order in which it is comparing the lists. It is worth noting that if the same list, provided in the example, was compared with them while unsorted, the complexity would go from 18 to 81. That scaled up when it comes to comparing 150 words resulting in the difference between 300 comparisons and 22,500 comparisons. That multiplied across six job description test cases and 102 syllabi to be 13,770,000 comparisons, with

unsorted lists. With sorted lists, the worst number of comparisons is 183,600 only.

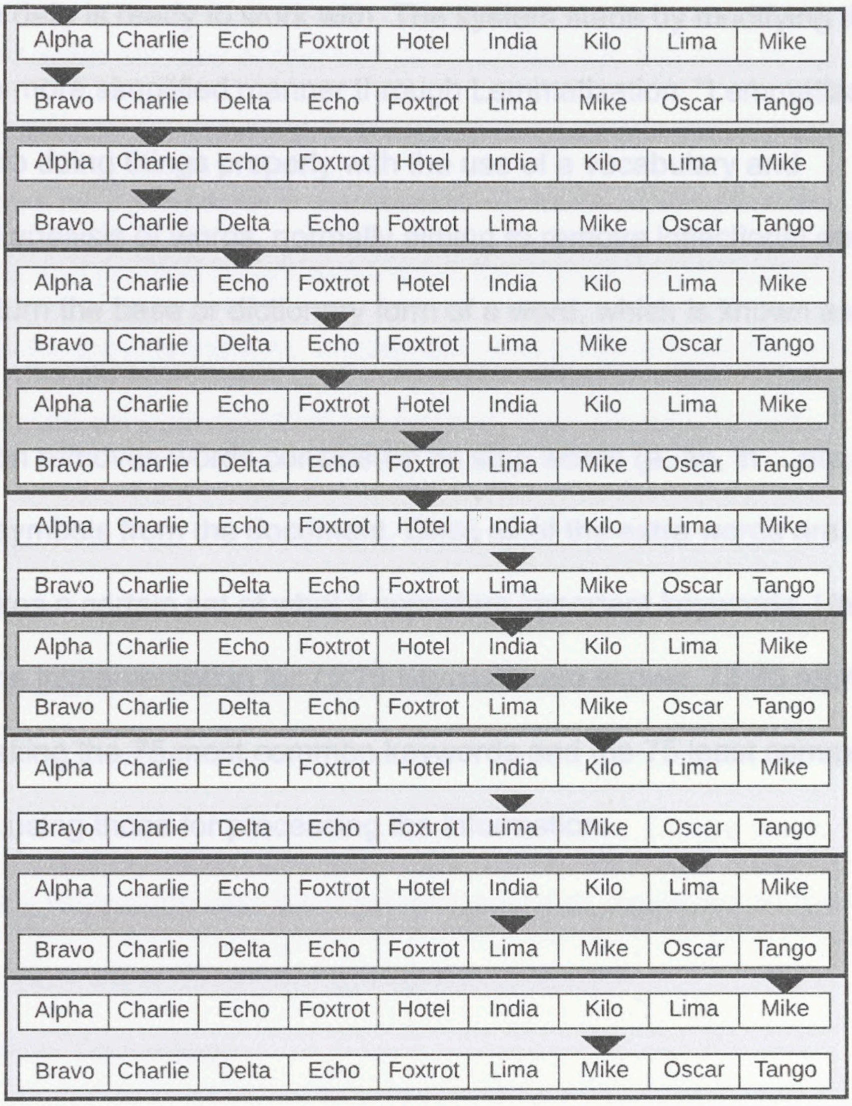


Figure 3: Comparison Of Sorted Lists

4.5 File Types

Thanks to the libraries imported, any syllabus can be added to the system with the file types: Docx, Txt, HTML, and PDF. Usage of each of the libraries as well as the implementation is provided in Appendix A.

4.6 Keyword Extraction

When extracting the keywords, Curtus goes through several steps to make sure the data is ready to work with. The system starts by modifying the data to be in a more simplified manner through Lemmatization. "Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma" [13].

Curtus then removes words considered as stop words (a, an, in …etc.) and removes any symbols from the document. Once all of the extra words are removed, it takes a certain set of what it considers important keywords. Under Appendix A, the implementation for 75:75 keywords are shown. 75:75 represents the program taking the 75 most common keywords and the 75 least common keywords and using those for processing the information.

Chapter 5: System Evaluation

5.1 Overview

Since we were not able to find any other systems in the area of this work, a manual evaluation had to be done to set benchmark results that Curtus' output can be compared to. The system is expected to compare existing courses to 6 job descriptions and provide recommendations for a good set of courses that map to each of the job descriptions. A plausible evaluation is intended where precision and recall are calculated for each of the input job descriptions.

5.2 Test Data

Job descriptions were selected based on the projection that they are good fits in the sense that it is obvious for a human to easily map each of them to a set of courses. The target of the evaluation to see if Curtus can provide the same set of courses, that is Curtus is a plausible system. In addition, Curtus would provide the courses as a ranked set with the most relevant ones displayed on the top. Based on that, jobs within the following areas were chosen:

- Information Technology
- Cyber Security
- Simulations
- Game Programming
- Web Developer
- Java Programmer

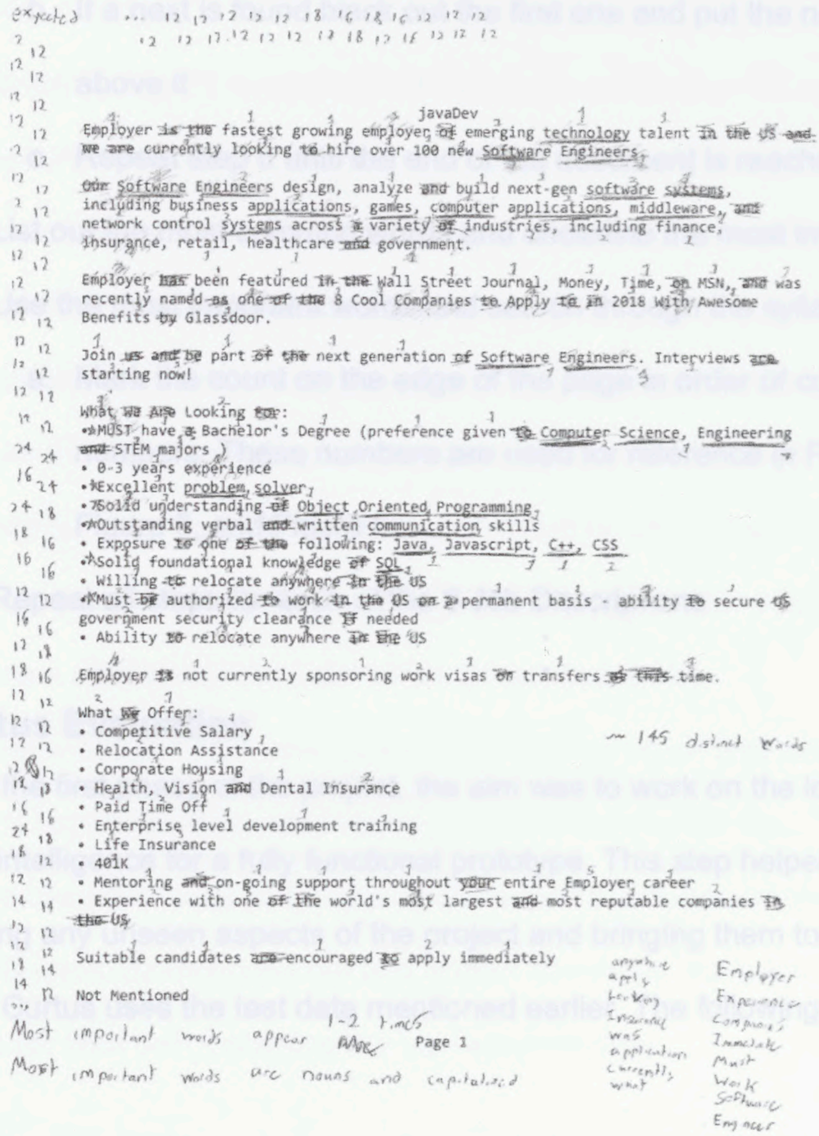
5.3 Human Evaluation

The researcher manualy went over all the testing job descriptions and came up with the best set of courses that map to the job describtions. These sets of courses provide the bench mark data that Curus results will be compared to. Each of the syllabi, course descriptions, and instructor descriptions were looked through and evaluated by hand to see what skills are taught in each of them which is further discussed in Section 5.4.

Curtus was evaluated and measured by the following metrics: 1. how many of the wanted keywords were found, 2. how many total keywords were found, and 3. how good the system is based on how many of the expected courses are provided by Curtus as recommended courses. The comparison of expected courses to recommended courses was measured under precision and recall.

5.4 Hand Traced Validation Set Process

Figure 4 below shows a scanned image of keywords from one of the job descriptions. The hand traced job descriptions was used as a way to evaluate the results obtained from Curtus. Good performance can be indicated by the system findings; if Curus finds the same number of keyword matches as the hand traced ones then the results are considered good.



javaDev

Employer is the fastest growing employer of emerging technology talent in the US and we are currently looking to hire over 100 new Software Engineers.

Our Software Engineers design, analyze and build next-gen software systems, including business applications, games, computer applications, middleware, and network control systems across a variety of industries, including finance, insurance, retail, healthcare and government.

Employer has been featured in the Wall Street Journal, Money, Time, on MSN, and was recently named as one of the 8 Cool Companies to Apply to in 2018 With Awesome Benefits by Glassdoor.

Join us and be part of the next generation of Software Engineers. Interviews are starting now!

What We Are Looking for:
• MUST have a Bachelor's Degree (preference given to Computer Science, Engineering and STEM majors)
• 0-3 years experience
• Excellent problem solver
• Solid understanding of Object Oriented Programming
• Outstanding verbal and written communication skills
• Exposure to one of the following: Java, Javascript, C++, CSS
• Solid foundational knowledge of SQL
• Willing to relocate anywhere in the US
• Must be authorized to work in the US on a permanent basis - ability to secure US government security clearance if needed
• Ability to relocate anywhere in the US

Employer is not currently sponsoring work visas or transfers at this time.

What We Offer:
• Competitive Salary
• Relocation Assistance
• Corporate Housing
• Health, Vision and Dental Insurance
• Paid Time Off
• Enterprise level development training
• Life Insurance
• 401k
• Mentoring and on-going support throughout your entire Employer career
• Experience with one of the world's most largest and most reputable companies in the US

Suitable candidates are encouraged to apply immediately

Not Mentioned

Page 1

Figure 4: Scanned Hand Traced Document (Java Dev)

The following steps were applied manually to each of the job descriptions to extract keywords and use them for matching the syllabi in the test data:

1. Cross out any stop words
2. Note the count of each word in the document
    a. Mark a one above the first

b. If a next is found black out the first one and put the next number above it

c. Repeat step b until the end of the document is reached

3. List out the most common words and underline the most important words
4. Use the most important words and search through the syllabi using them
   a. Mark the count on the edge of the page in order of course numbers. These numbers are used for reference in Figure 5, Figure 6, and Figure 8.
5. Repeat all steps for each of the 6 Job Descriptions

5.5 Curtus Evaluation

In the first phase of the project, the aim was to work on the lowest level of required intelligence for a fully functional prototype. This step helped in discovering any unseen aspects of the project and bringing them to light. For file retrieval, Curtus uses the test data mentioned earlier. The following steps were applied:

1. Extract top 150 most common keywords from each syllabus
2. Extract top 150 most common keywords the job description
3. Evaluate the syllabi to see which one has the most matching keyword
4. Take the top 10 matches and further evaluate them through keyword weights and provide them as results sorted by valued importance. Valued importance was based on how many distinct keywords are found and how many of each of those keywords are found.

Repeat With Changes:

1. Extract the top 75 most common keywords and bottom 75 least common keywords from each syllabus
2. Extract the top 75 most common keywords and bottom 75 least common keywords from the job description
3. Evaluate the syllabi to see which one has the most matching keywords
4. Take the top 10 matches and further evaluate them through keyword weights and provide them as results sorted by valued importance.

Top keywords are considered the most common words within a file and bottom keywords are considered the least common words in a file.

Further evaluated files were ordered by keyword count and were used to view which keywords were most commonly found across the files. Two data sets were used because many of the important keywords are less common in the texts being used. By using the top and bottom keywords, the hope is to reach better results than with just using the top keywords only.

Chapter 6: Results, Conclusion and Future Work

6.1 Results

Job descriptions for a Java Developer, Information Technology, Simulation Engineer, Threat Analyst, Web Developer, and Game Developer were used to test Curtus. With each one of these, Curtus would evaluate each course and provide how many keyword matches are found, how many words there are in the file, weights for the keywords, and a weight for the course. Weights for keywords were assigned by how frequently a word appeared within the file. These weights provide another metric to sort the results. For each job description, Curtus would also provide a word count of each keyword that was considered from the syllabi.

With the word count, the software was able to be refined further by revealing words that were not correctly filtered out by the keyword extraction. Examples would be words like us, we, and them. After the process of filtering out those words, Curtus was tested and results were collected from this set. Those results were then compared with a document containing hand traced forms of the job descriptions that marked which words should be considered the most significant ones in Curtus as well as which courses should be considered good fits for the job. The results of this comparison are shown in Figure 5.

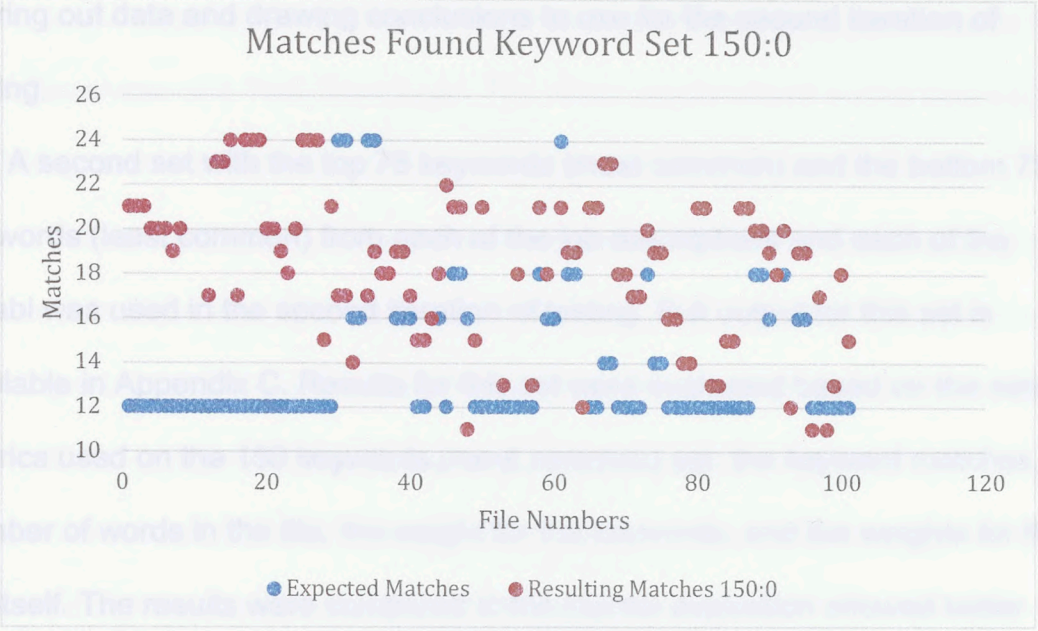


Figure 5: Expected matches versus matches found for top 150 most common keywords

In Figure 5, the blue dots represent the results from human evaluation, and the red dots represent the results obtained from Curtus. The closer the dots are to one another, the better the results. With the first set being used (150 most common words), Curtus ended up missing many of the prime keywords. This can be attributed to the fact that Curtus was simply working off of which syllabi had words that are more common in the English language.

The results were further evaluated using the weights of the files. Smaller files would have a larger weight assigned to each keyword. Based on how many keywords are found in the file, this would then be multiplied by the weight to provide a weight for the file as a whole. Using that finalized value, the top 10 matches were compared to the resulted ones from just counting the keywords. The results were certainly interesting but did not provide any better results that would change the previous conclusion for this set. This left the result from testing Curtus with this set not being a valid solution. This set did, however, help in

filtering out data and drawing conclusions to use for the second iteration of testing.

A second set with the top 75 keywords (most common) and the bottom 75 keywords (least common) from each of the job descriptions and each of the syllabi was used in the second iteration of testing. Full output for this set is available in Appendix C. Results for this set were evaluated based on the same metrics used on the 150 keywords (most common) set: the keyword matches, the number of words in the file, the weight for the keywords, and the weights for the file itself. The results were compared to the human evaluation showed better performance this time as shown in Figure 6.

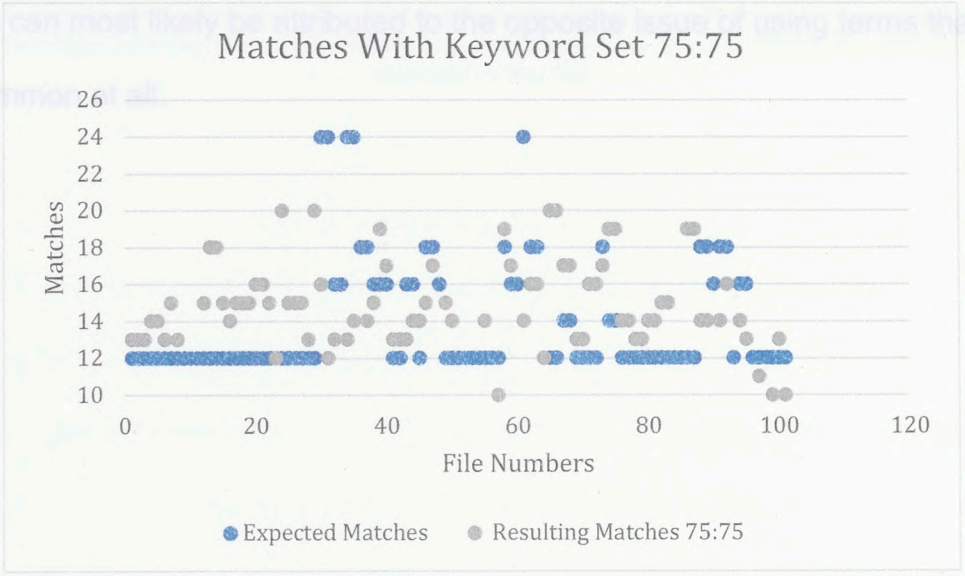


Figure 6: Expected matches versus matches found with 75 most common keywords and 75 least common keywords

In Figure 6, the blue dots represent the results from the human evaluation, and the grey dot represents the results obtained from Curtus. The closer the dots are to one another the better the results. Many of the keywords that were not

considered in the first set appeared in the second set providing good results for Java Developer and Web Developer. The other results based on the keyword matching were not quite as good.

These results were further evaluated similarly to the previous test using the weights of the files. The top 10 matches were compared to the human results showing much better results compared to using the previous set. However they were still not satisfactory results. An example run is shown in Figure 8. If the weights were used the Information Technology and Game Developer jobs had improved decisions while not when considering the other jobs. The results for Information Technology can be attributed to syllabus descriptions using terms that are more common to the field of computer science. The Game Developer results can most likely be attributed to the opposite issue of using terms that are not common at all.

```
Please Input a File:
javaDev.txt
File Found
Evaluating File [100.00%] : DONE: ./JobDescriptions/javaDev.txt
('CLASS_AG', 20)
('CLASS_AR', 20)
('CLASS_BC', 20)
('CLASS_AB', 20)

You should take the following classes
File Name                | Keywords   |  Strength   |
CLASS_AG                 | 20/150     |  <41.867>   |
CLASS_AR                 | 20/150     |  <41.6>     |
CLASS_BC                 | 20/150     |  <33.507>   |
CLASS_AB                 | 20/150     |  <29.107>   |
CLASS_BG                 | 19/150     |  <42.978>   |
CLASS_CA                 | 19/150     |  <33.807>   |
CLASS_BE                 | 19/150     |  <33.757>   |
CLASS_BN                 | 19/150     |  <31.629>   |
CLASS_AU                 | 18/150     |  <27.168>   |
CLASS_CW                 | 18/150     |  <27.168>   |
```

Figure 7: Active Run-End User This figure shows the results of a test run for the prototype. The user provides a txt file with a job description, and it provides the top 10 results sorted by the strength of the file.

6.2 Results Analysis

The obtained results show that the least common words are as important as the most common words. Results from using both sets provide that the second set is far better than the first one. When comparing the two sets the 75:75 was much closer to the expected result as shown in Figure 8.

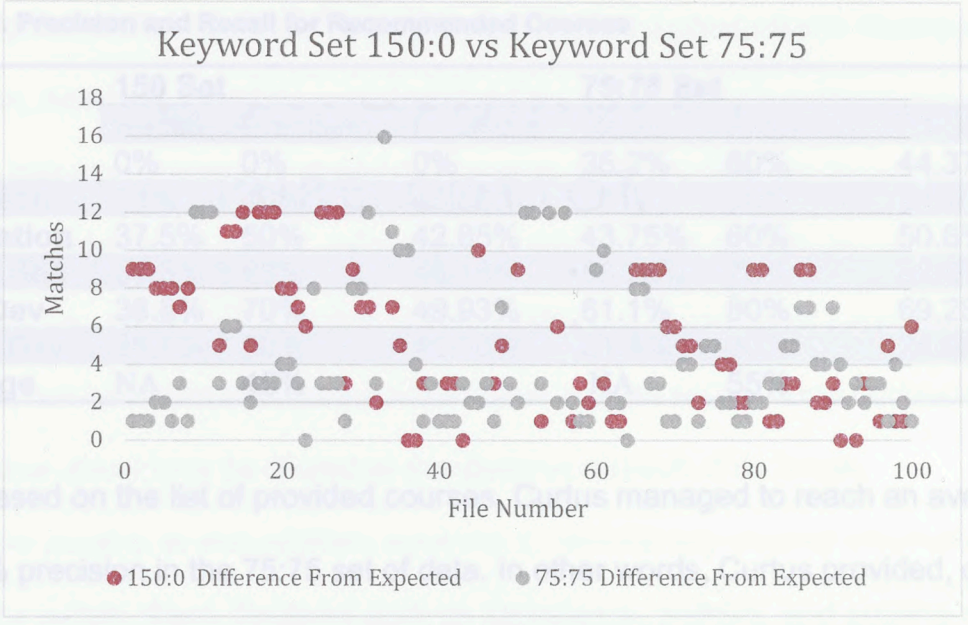


Figure 8: 150:0 and 75:75 Compared.

Figure 8 shows how far the results strayed from the expected results. The 150:0 represents the 150 most common keywords found in the file and the 75:75 represents the set that used the 75 most common keywords and the 75 least common keywords. The lower the dots are on the graph, the better the result is. The first phase however was not intended to be able to have a full solution but instead to reveal what attributes need to be considered and what needed to be added.

Even though the results based on a number of keywords were better under the set of 75:75 that does not answer the main question: does the system

recommend good classes?. Referring back to results from the human evaluation, each of the job descriptions was associated with a list of courses that, from a human perspective, are considered great courses for the job. It is worth noting that the list of provided courses contains 10-15 courses in no particular order. The results for this are displayed in Table 2.

Table 2. Precision and Recall for Recommended Courses

	150 Set			75:75 Set		
	Recall	Precision	F1 Score	Recall	Precision	F1 Score
Java	0%	0%	0%	35.2%	60%	44.37%
Info Tech	53%	40%	45.59%	20%	30%	24%
Simulation	37.5%	50%	42.85%	43.75%	60%	50.6%
Cyber Sec	37.5%	60%	46.15%	56.25%	70%	62.37%
Web Dev	38.8%	70%	49.93%	61.1%	80%	69.28%
Game Dev	35.7%	50%	41.66%	21.4%	30%	24.98%
Average	NA	45%		NA	55%	

Based on the list of provided courses, Curtus managed to reach an average of 55% precision in the 75:75 set of data. In other words, Curtus provided, on average, 5 to 6 classes that can be considered good classes. One important note on the results is that the presence of 4 to 5 classes that are not considered good classes does not mean those classes do not fit at all. These classes most likely can still provide skills applicable to the job. However this simply means that there may still be better courses.

The major question that was brought up from the results was, how the syllabus format affect the results? In this case, it directly affects it. Many of the courses provide more information than just what is taught in the class. For example, many syllabi provide sections with course policies, attendance policies, ...etc. Some of them list extra information about a field that they are tethered to

even though they might not teach those particular skills. That extra information can be very problematic for software like Curtus because in some cases longer syllabi might have only information on what is learned in the class while others might have information that is not part of the content learned in the class.

An example can be if a syllabus for one course gives a great explanation of the targeted skills in a class, it would be considered a good match. That good syllabus, however, could be overshadowed by another syllabus that is longer and simply has more keywords available to match. With the fact that larger weights are assigned to short syllabi, using weights properly can be affected by this problem, i.e., a long syllabus that covers anything a student might learn about in a class, but has extra information would have a small weight. That concludes that a syllabus should not be diluted or devalued as a result of its length.

One solution to this problem would be to remove any unused information from the syllabi (files). Sections such as attendance, policies, and grading would be removed from the file leaving the important information only. This also left a question behind: what makes a good syllabus? For Curtus to provide the best results from a provided syllabus, the syllabus should contain:

- Skills learned within that class;
- A calendar or schedule with detailed information on what topics would be covered each week.

The worst results were obtained from syllabi that:

- contain a broad description of skills that should be gained by the course
- contain either no calendar or undetailed calendar

- provides a background on the field of study presenting skills that are not covered within the course
- have information copy and pasted from other courses and not correctly modified

If certain syllabi were omitted from the system, the results would have been much better. However that would not result in a finalized solution. A finalized solution would be to provide valid results using all the information provided and being able to make its own decisions on which information to consider.

6.3 Further Results

Error! Reference source not found. shows the numeric values resulting from all the test cases. The column Expected shows the number of keywords expected for each of the syllabi for that job description. The total columns display how many keywords Curtus was able to find. The range columns show how different the results were from each of the expected values (human evaluation). The lower the range is the closer the results were to the expected values.

Table 3: Total Results

Job	Expected	150 total	75:75 total	150 range	75:75 range
Java Dev	1404	1702	1317	592	423
Info Tech	1237	1422	1186	385	277
Simulation	1298	1620	1161	490	297
Cyber Sec	1348	1723	1245	517	279
Web Dev	948	1010	793	250	257
Game Dev	1362	1965	1824	731	626

Appendix B has all the graphs providing visual results for all of the individual values that lead to the results of the columns in **Error! Reference source not found.**. Table 4 gives information on each of the details in Appendix B.

Table 4: Short Summary

Appendix	Display	Blue Dots – Expected Matches Red Dots – Matches under 150:0 Keywords Green Dots – Matches under 75:75 Keywords
B1		
	Description	All six tables under this section are pretty simple in how to interpret. The goal of the program was to get the results as close to the blue dots as possible
Appendix	Display	Blue Dots – Expected Matches Red Dots – Matches under 150:0 Keywords
B2		Green Dots – Matches under 75:75 Keywords
	Description (Graph 1)	The First Graph shows the average difference of each of the graphs from the expected results. The closer the graph is to 0 the better the results. The green dots are the minimum of the two graphs making them the better results of the two
	Description (Graph 2)	This graph shows the plotted results of Table 2 above. Same as the graphs from Appendix B1, results are considered better the closer they are to the blue dots.

6.4 Conclusion

Curtus is a tool aimed to map job descriptions to course syllabi and worked to provide students with classes that they can take for a specific job with high confidence. With the constantly changing work environment, this tool provides a method of alleviating the difficulty of keeping up. Curtus provides a solution to this problem and even helped in other ways through the research process leading up to the working prototype. It revealed things such as what makes a good syllabus, as well as providing other means of research in the field of Natural Language Processing. Curtus leaves openings for more research on it. One location being that it does not currently have the right metric to most effectively

order the courses from best to just good. Areas like this help to provide a means of improving a system has already been considered a success.

6.5 Future Work

Improvement of keyword searching can be investigated. This step would aim to give a much more thorough evaluation of how much a syllabus maps to a job description. One way to attempt this can be by applying weights to words using bigrams. Bigrams can be used as a tool to find how important a word might be. In a job description, the first iteration might find the words Python and Java and consider them of equal weight when it comes to just those keywords alone. Bigrams, however, can make a difference by possibly introducing more information about it as it considers associated modifiers like “Requires Java” and “Python System.” This puts the words in a different category where Python would be considered as just being a background while Java is a mandatory requirement for the job. Accordingly, syllabi that have the word Java would have more weight.

References

[1] S. Jaschik, "Well-Prepared in Their Own Eyes.," 20 January 2015. [Online]. Available: https://www.insidehighered.com/news/2015/01/20/study-finds-big-gaps-between-student-and-employer-perceptions.

[2] P. M. Nadkarni, L. Ohno-Machado and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association,* vol. 18, no. 5, p. 544–551, September 2011.

[3] K. S. Jones, "Natural Language Processing: A Historical Review," in *Current Issues in Computational Linguistics: In Honour of Don Walker*, vol. 9, Springer, Dordrecht.

[4] S. Shankar, "Quora: What are the current hot topics in natural language processing," 2016. [Online]. Available: https://www.quora.com/What-are-the-current-hot-topics-in-natural-language-processing.

[5] A. Montoyo, P. MartíNez-Barco and A. Balahur, "Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments," *Decision Support Systems,* vol. 53, no. 4, pp. 675-679, 2012.

[6] G. Dias, E. Alves and J. G. P. Lopez, "Topic Segmentation Algoritms for Text Summarization and Passage Retrieval: An Exaustive Retrieval," *AAAI,* vol. 7, pp. 1334-1340, 2007.

[7] C. A. Thompson, R. J. Mooney and M. E. Califf, "Active Learning for Natural Language Parsing and Information Extraction," *ICML,* pp. 406-414, 1999.

[8] J. C. Denny, J. D. Smithers, R. A. Miller and A. Spickard III, "Understanding Medical School Curriculum Content Using Knowledge Maps," *Journal of the American Medical Informatics Association,* vol. 10, no. 4, pp. 351-362, 2003.

[9] E. Brill, "Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging," *Computational Linguistics,* vol. 21, no. 4, pp. 543-565, 1995.

[10] J. Burstein, J. Shore, J. Sabatini, B. Moulder, J. Lentini, K. Biggers and S. Holtzman, "From Teacher Professional Development to the Classroom: How NLP Technology Can Enhance Teachers' Linguistic Awareness to Support Curriculum Development for English Language Learners.," *Journal of Educational Computing Research,* vol. 51, no. 1, pp. 119-144, 2014.

[11] R. M. Palau and M.-F. Moens, "Argumentation mining: the detection, classification and structure of arguments in text," *ICAIL '09 Proceedings of the 12th International Conference on Artificial Intelligence and Law,* pp. 98-107 , 2009.

[12] D. D. Lewis, "Natural Language Processing for Information Retrieval," *Communications of ACM,* vol. 39, no. 1, pp. 92-101, 1996.

[13] C. D. Manning, R. Prabhakar and H. Schütze, Introduction To Information Retrieval, Cambridge University Press, 2008.

[14] J. Sun, J. Ma, X. Liu, Z. Liu, G. Wang, H. Jiang and T. Silva, "A Novel Approach for Personalized Article Recommendation in Online Scientific Communities," in *46th Hawaii International Conference on System Sciences*, Wailea, Maui, HI, 2013.

[15] B. Marthi, H. Pasula, S. Russell, B. Milch and I. Shpitser, "Identity Uncertainty and Citation Matching," in *Advances in neural information processing systems*, 2003.

[16] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan and X. Cheng, "Text matching as image recognition," in *Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*, Phoenix, Arizona, 2016.

[17] S. Argamon, N. Goharian, D. Grossman and O. Frieder, "A Specialization in Information and Knowledge Management Systemsfor the Undergraduate Computer Science Curriculum," in *Information Technology: Coding and Computing*, 2005.

[18] S. Bird, E. Klein and E. Loper, Natural Language Processing With Python, O'Reilly Media, 2009.

[19] S. Bird and E. Loper, "NLTK: The natural language toolkit," in *ACLdemo '04 Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, Barcelona, Spain, 2004.

Appendix A: Scripts

Appendix A1: Sorting Algorithm

```
def mergeSort(lst):
    if len(lst)>1:
        center = len(lst)//2
        left = lst[:center]
        right = lst[center:]
        mergeSort(left)
        mergeSort(right)
        i=0
        j=0
        k=0
        while i < len(left) and j < len(right):
            if left[i] < right[j]:
                lst[k]=left[i]
                i=i+1
            else:
                lst[k]=right[j]
                j=j+1
            k=k+1
        while i < len(left):
            lst[k]=left[i]
```

```
            i=i+1
            k=k+1
        while j < len(right):
            lst[k]=right[j]
            j=j+1
            k=k+1
```

Appendix A2: Keyword Comparison

```
def compareLists(jobDescriptionKeys, syllabusKeys, show ):
    shared = []
    count = 0
    a = 0
    b = 0
    while(a < len(jobDescriptionKeys) and b < len(syllabusKeys)):
        if(jobDescriptionKeys[a].lower() < syllabusKeys[b].lower()):
            a += 1
        elif(jobDescriptionKeys[a].lower() > syllabusKeys[b].lower()):
            b += 1
        elif(jobDescriptionKeys[a].lower() == syllabusKeys[b].lower()):
            count += 1
            shared.append(syllabusKeys[b])
            if(show == True):
                print(syllabusKeys[b])
```

```
        a += 1
        b += 1
    return shared
```

Appendix A3: Supported Files

```
def getTextFromFile(fileName):
    content = ""
    mySwitch = getFileType(fileName)
    #for txt file types
    if(mySwitch == "txt"):
        f = open(fileName, "r")
        content = f.read()
        f.close()
    #for doc file types
    elif(mySwitch == "doc"):
        content = ""
    #for docx file types
    elif(mySwitch == "docx"):
        doc = docx.Document(fileName)
        fullText = []
        for para in doc.paragraphs:
            fullText.append(para.text)
        content = '\n'.join(fullText)
```

```
#for html file types
elif(mySwitch == "html"):
    f=codecs.open(fileName, 'r', 'utf-8')
    document= BeautifulSoup(f.read(), features="lxml").get_text()
    content = document
#for pdf file types
elif(mySwitch == "pdf"):
    pdfFileObject = open(fileName, 'rb')
    pdfReader = PyPDF2.PdfFileReader(pdfFileObject)
    count = pdfReader.numPages
    fullText = []
    for i in range(count):
        page = pdfReader.getPage(i)
        fullText.append(page.extractText())
    content = '\n'.join(fullText)
    test = ''.join(fullText)
    pdfFileObject.close()
    if(len(test)==0):
        from tika import parser
        parsed = parser.from_file(fileName, xmlContent=True)
        tree = ET.fromstring(parsed["content"])
        content = ET.tostring(tree, encoding='utf8', method='text')
else:
```

```
        print("No matching file type found for: "+mySwitch)
    return content
```

Appendix A4: Keyword Extraction

```
def extractTokens(text):
    try:
        tokens = word_tokenize(text)
    except:
        print("Error 101")
        input("Press enter to exit")
        sys.exit()
    lemmatizer = WordNetLemmatizer()
    bonuses = [lemmatizer.lemmatize(token) for token in tokens]
    stopwordList = stopwords.words('english')
    extrabonus = [bonus for bonus in bonuses if bonus not in stopwordList and
bonus.lower() not in otherWords]
    noSym = [token for token in extrabonus if token != token.upper()]
    return noSym


def getKeywords(text):
    lst = extractTokens(text)
    fdist1 = FreqDist(lst)
    keys = KEYWORDS//2
```

```
most = fdist1.most_common(keys)

least = fdist1.most_common()[-keys:]
result = most + list(set(least)-set(most))

return result
```

Appendix B

Appendix B1

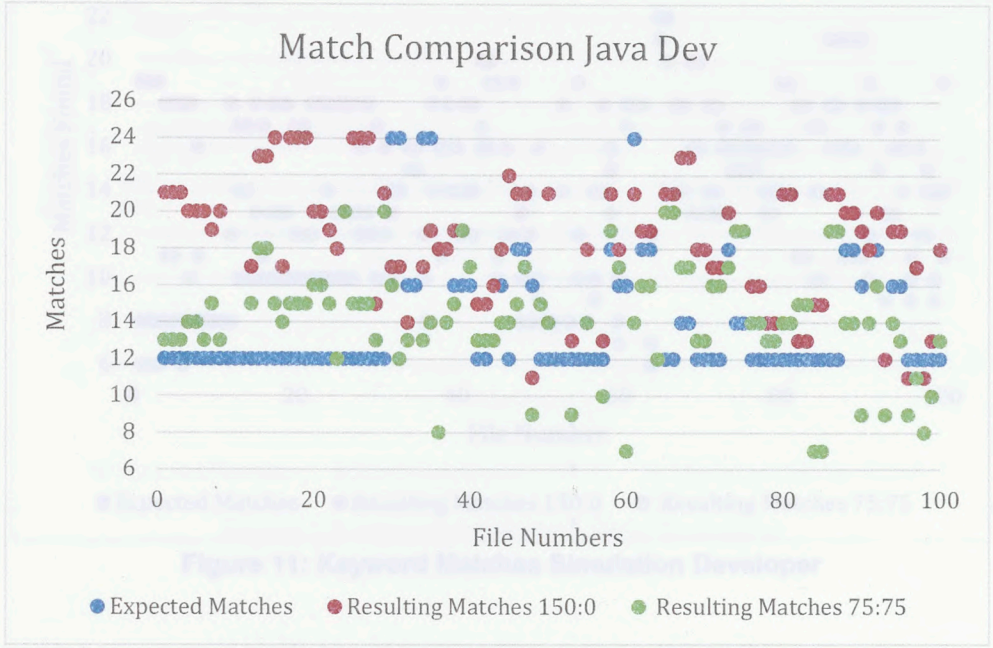


Figure 9: Keyword Matched Java Developer

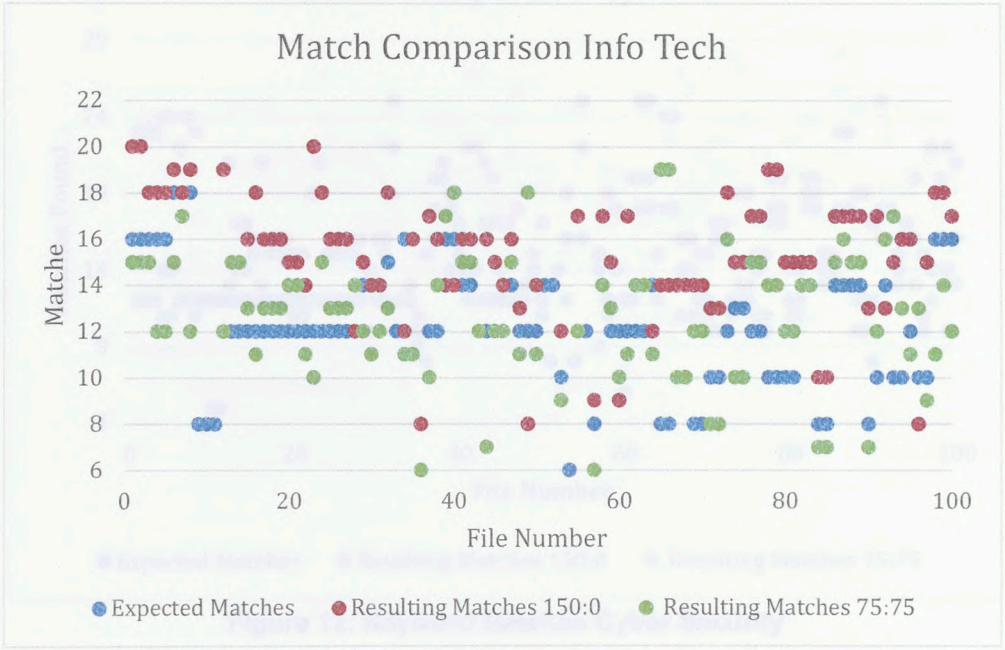


Figure 10: Keyword Matches Infomation Security

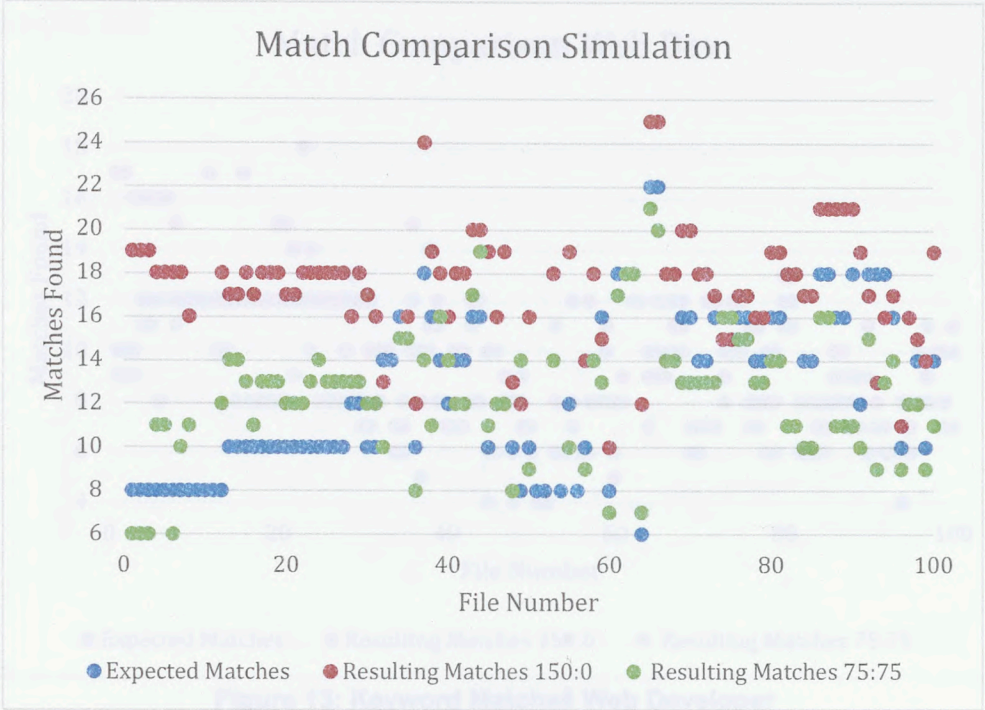


Figure 11: Keyword Matches Simulation Developer

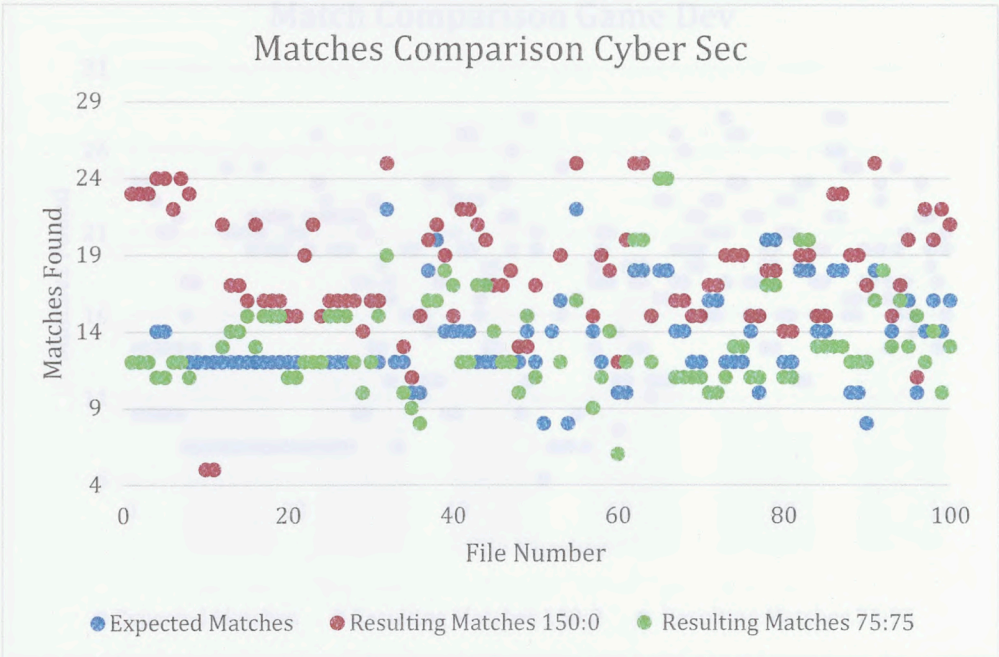


Figure 12: Keyword Matches Cyber Security

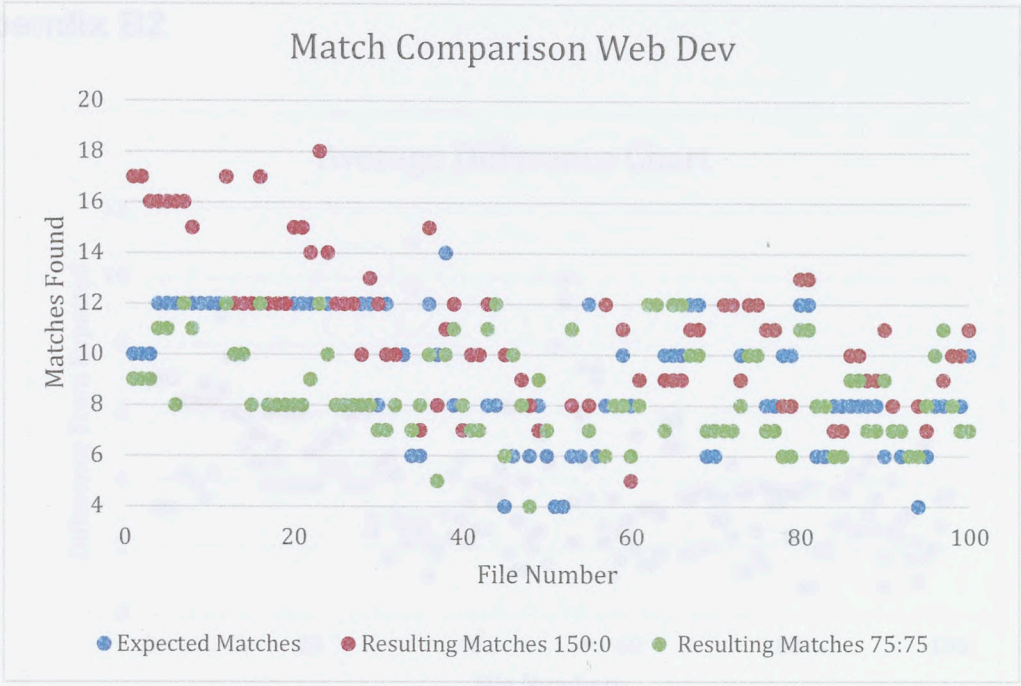


Figure 13: Keyword Matches Web Developer

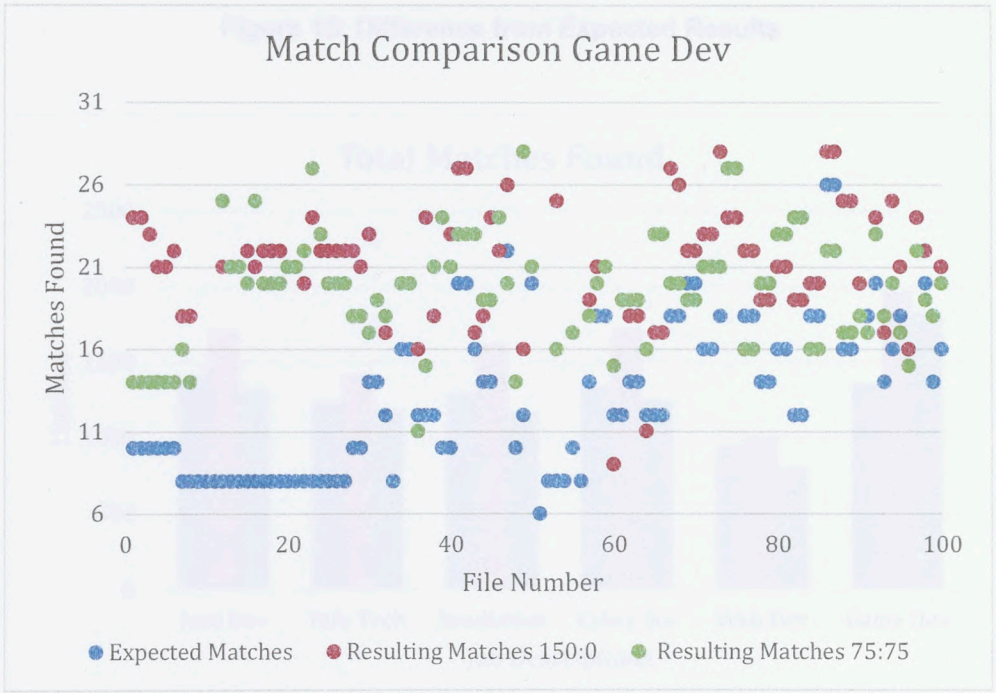


Figure 14: Keyword Matches Game Developer

Appendix B2

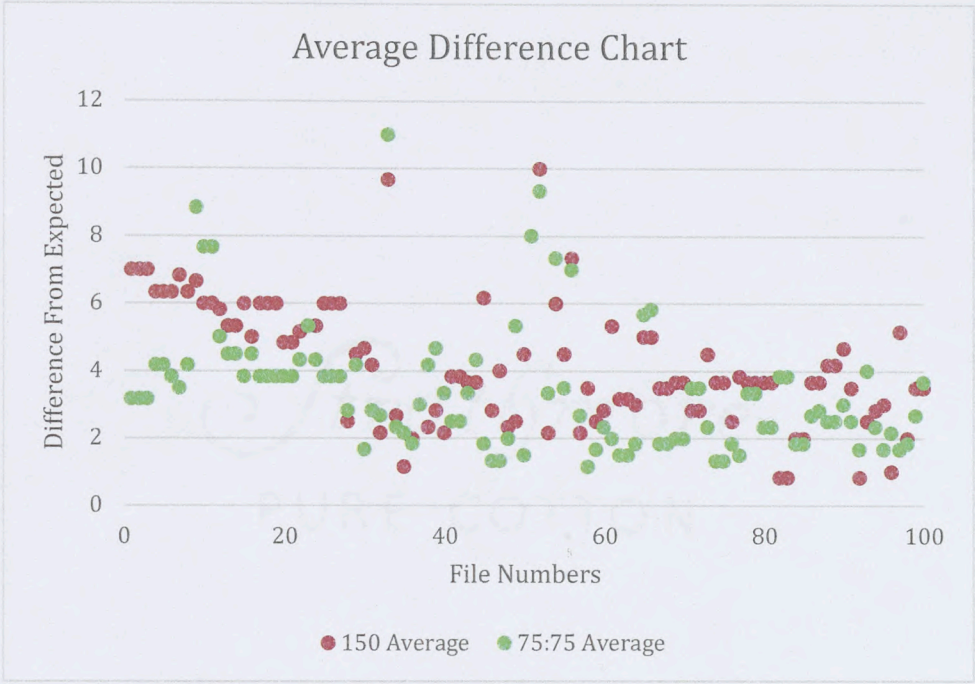


Figure 15: Difference from Expected Results

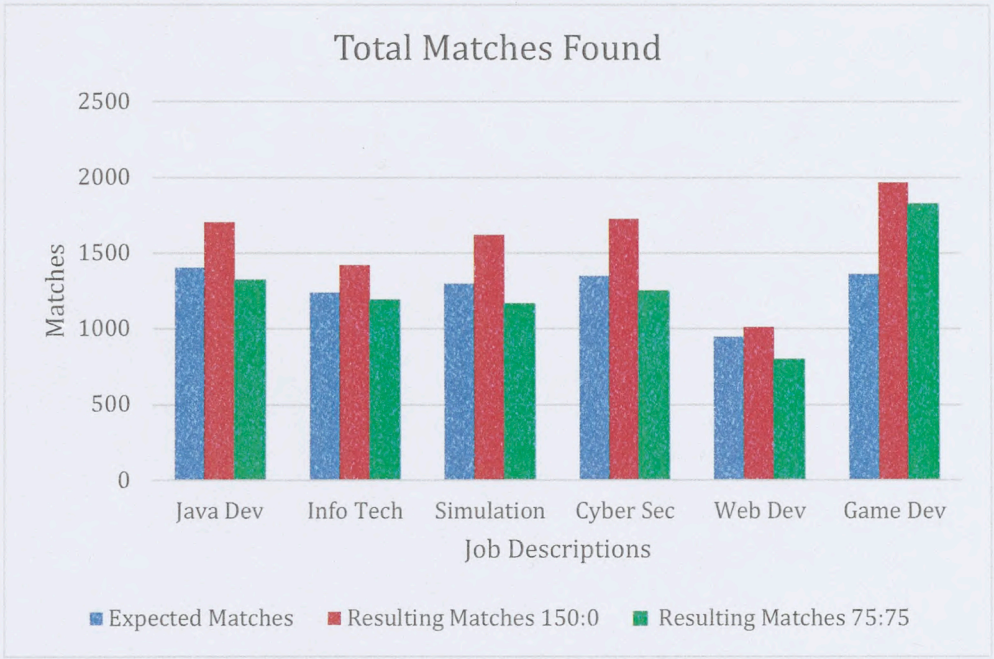


Figure 16: Match Comparison Bar Graph

Appendix C: Execution 75:75 Output

```
Evaluating File [100.00%] : DONE: ./JobDescriptions/javaDev.txt
('CPSC1301_WangS_80814_Fall2018.docx', 20)
('CPSC1302_ShushaneR_80821_Fall2018.pdf', 20)
('CPSC6105_YangJ_80928_Fall2018.docx', 20)
('CPSC6105_YangJ_81041_Fall2018.docx', 20)
```

computer	:134	work	:93	problem	:74	time	:70	security	:60
design	:58	ability	:56	must	:53	knowledge	:53	system	:51
solid	:51	software	:44	following	:44	science	:44	understanding	:42
programming	:41	support	:31	throughout	:30	one	:24	communication	:22
written	:19	mentioned	:19	technology	:18	development	:17	including	:15
apply	:14	network	:13	skill	:12	given	:11	authorized	:10
next	:10	business	:9	major	:8	application	:8	assistance	:8
encouraged	:7	experience	:7	java	:5	analyze	:4	object	:4
part	:4	level	:4	career	:4	journal	:4	new	:4
game	:3	control	:3	javascript	:1	world	:1	secure	:1
health	:1	life	:1	entire	:1	engineering	:1	fastest	:1
starting	:1	basis	:1	named	:1				

```
Evaluating File [100.00%] : DONE: ./JobDescriptions/infoTech.txt
('CPSC6105_YangJ_80928_Fall2018.docx', 19)
('CPSC6105_YangJ_81041_Fall2018.docx', 19)
```

work	:93	need	:90	computer	:88	problem	:74	time	:69
responsibility	:66	ability	:56	team	:53	system	:51	office	:43
understanding	:42	support	:36	issue	:36	required	:31	phone	:27
communication	:22	responsible	:22	attendance	:22	state	:21	written	:19
way	:16	including	:15	maintain	:15	moderately	:14	network	:13
skill	:12	portion	:10	management	:10	activity	:10	quality	:9
follow	:8	comprehensive	:8	services	:8	environment	:7	experience	:7
description	:7	operating	:6	systems	:5	service	:5	hardware	:4
managing	:4	similar	:4	staff	:4	windows	:4	new	:4
task	:3	microsoft	:3	physical	:3	product	:3	set	:2
active	:2	duty	:2	projects	:2	success	:2	member	:2
supervisor	:2	provide	:1	ensuring	:1	perform	:1	laptop	:1
instruction	:1								

```
Evaluating File [100.00%] : DONE: ./JobDescriptions/simulationEngineer.txt
('CPSC6105_YangJ_80928_Fall2018.docx', 21)
```

work	:93	computer	:88	requirement	:84	security	:60	design	:58
project	:54	team	:53	system	:51	concept	:51	modeling	:49
process	:42	impact	:39	software	:38	lab	:36	able	:34
support	:31	throughout	:30	day	:25	one	:24	technology	:18
development	:17	solution	:15	test	:15	algorithm	:12	business	:9
application	:8	related	:8	environment	:7	experience	:7	include	:7
based	:7	model	:5	service	:5	simulation	:5	type	:4
get	:4	best	:4	database	:4	top	:4	analysis	:4
competency	:4	understand	:4	new	:4	threat	:4	task	:3
bring	:3	evaluation	:3	held	:3	control	:3	active	:2
mind	:2	exercise	:2	success	:2	testing	:2	architecture	:2
documentation	:2	obtain	:2	small	:1	closely	:1	growth	:1
world	:1	foster	:1	provides	:1	engineering	:1	begin	:1
defense	:1	cyber	:1	role	:1				

```
Evaluating File [100.00%] : DONE: ./JobDescriptions/threatAnalyst.txt
('CPSC6105_YangJ_80928_Fall2018.docx', 24)
('CPSC6105_YangJ_81041_Fall2018.docx', 24)
```

work	:93	need	:90	also	:72	project	:62	security	:61
ability	:56	knowledge	:53	team	:53	global	:50	information	:44
range	:43	office	:42	basic	:36	portfolio	:35	required	:31
support	:31	demonstrate	:28	one	:24	communication	:22	written	:19
risk	:18	technology	:18	professional	:16	including	:15	solution	:15
within	:15	general	:14	action	:12	skill	:12	technical	:11
given	:11	sense	:10	collaborative	:10	management	:10	effectively	:10
business	:9	presentation	:9	environment	:7	additional	:7	learn	:6
systems	:5	service	:5	meeting	:4	part	:4	best	:4
individual	:4	top	:4	staff	:4	responsibilities	:4	overview	:3
microsoft	:3	area	:3	product	:3	well	:2	high	:2
member	:2	operation	:2	opportunity	:2	delivery	:2	virtual	:1
wide	:1	provide	:1	execute	:1	provides	:1	various	:1
committed	:1	value	:1	domain	:1	defense	:1	develop	:1
cyber	:1	purpose	:1	demonstrated	:1	honesty	:1		

```
Evaluating File [100.00%] : DONE: ./JobDescriptions/webDeveloper.txt
('CPSC1105_NguyenT_80702_Fall2018.pdf', 12)
('CPSC1301L-WangS_80818_Fall2018.docx', 12)
('CPSC1301L_FleenorH_80817_Fall2018.html', 12)
('CPSC1301_FleenorH_80813_Fall2018.html', 12)
('CPSC3131_Lee,Y_80904_Fall2018.docx', 12)
('CPSC5157U_YangJ_81040_Fall2018.docx', 12)
('CPSC5157U_YangJ_82214_Fall2018.docx', 12)
('CPSC6105_YangJ_80928_Fall2018.docx', 12)
('CPSC6105_YangJ_81041_Fall2018.docx', 12)
```

work	:93	problem	:74	also	:72	project	:54	must	:53
range	:43	programming	:41	software	:38	able	:34	fundamental	:33
demonstrate	:28	language	:24	code	:22	technology	:18	development	:17
including	:15	maintain	:15	within	:15	skill	:12	seeking	:10
take	:8	application	:8	environment	:7	experience	:7	solving	:6
outside	:5	least	:5	service	:5	similar	:4	staff	:4
why	:3	microsoft	:3	would	:3	control	:3	product	:3
well	:2	web	:2	small	:1	growth	:1	javascript	:1
wide	:1	version	:1	care	:1	excellence	:1	health	:1
eligible	:1	look	:1	basis	:1	ideal	:1	full	:1

```
Evaluating File [100.00%] : DONE: ./JobDescriptions/gameDev.txt
('CPSC4000_SummersW_80908_Fall2018.docx', 28)
```

work	:93	requirement	:84	material	:82	problem	:74	also	:72
meet	:69	time	:69	responsibility	:66	design	:58	used	:57
ability	:56	organization	:56	project	:54	must	:53	knowledge	:53
team	:53	concept	:51	good	:51	following	:44	office	:42
process	:42	programming	:41	able	:34	required	:31	implement	:27
accommodation	:23	communication	:22	responsible	:22	assigned	:22	using	:21
mentioned	:19	development	:17	solution	:15	your	:13	prior	:12
specifically	:12	skill	:12	deliver	:10	collaborative	:10	management	:10
quality	:9	disability	:9	related	:8	environment	:7	experience	:7
include	:7	based	:7	description	:7	function	:7	solving	:6
base	:6	usually	:5	service	:5	building	:5	multiple	:4
deadline	:4	individual	:4	level	:4	cover	:4	without	:4
described	:4	game	:4	listed	:3	target	:3	made	:3
object-oriented	:3	evaluation	:3	reasonable	:3	physical	:3	would	:3
control	:3	product	:3	guide	:3	duty	:2	except	:2
high	:2	alter	:2	etc.	:2	pattern	:2	interactive	:2
delivery	:2	version	:1	various	:1	player	:1	idea	:1
perform	:1	develop	:1	objective	:1	role	:1	platform	:1
user	:1								

Keywords Being Used: 150

File Name	Java Dev	Info Tech	Simulation	Cyber Sec	Web Dev	Game Dev	
CPSC1105_Berrios-RolonM_80718_Fall2018.docx	13/150	15/150	6/150	12/150	9/150	14/150	
-->Words In File	761	761	761	761	761	761	SD:5.711
-->Weight of Result	1.708%	1.971%	0.788%	1.577%	1.183%	1.84%	
-->Calculated Strength	<14.803>	<19.71>	<3.152>	<12.616>	<7.098>	<17.173>	<12.425>
CPSC1105_BowmanJ_80723_Fall2018.docx	13/150	15/150	6/150	12/150	9/150	14/150	
-->Words In File	762	762	762	762	762	762	SD:5.704
-->Weight of Result	1.706%	1.969%	0.787%	1.575%	1.181%	1.837%	
-->Calculated Strength	<14.785>	<19.69>	<3.148>	<12.6>	<7.086>	<17.145>	<12.409>
CPSC1105_BrumbaughE_80710_Fall2018.docx	13/150	15/150	6/150	12/150	9/150	14/150	
-->Words In File	754	754	754	754	754	754	SD:5.762
-->Weight of Result	1.724%	1.989%	0.796%	1.592%	1.194%	1.857%	
-->Calculated Strength	<14.941>	<19.89>	<3.184>	<12.736>	<7.164>	<17.332>	<12.541>
CPSC1105_BrumbaughE_80715_Fall2018.docx	13/150	15/150	6/150	12/150	9/150	14/150	
-->Words In File	754	754	754	754	754	754	SD:5.762
-->Weight of Result	1.724%	1.989%	0.796%	1.592%	1.194%	1.857%	
-->Calculated Strength	<14.941>	<19.89>	<3.184>	<12.736>	<7.164>	<17.332>	<12.541>
CPSC1105_CanedoJJ_80699_Fall2018.pdf	14/150	12/150	11/150	11/150	11/150	14/150	
-->Words In File	921	921	921	921	921	921	SD:2.436
-->Weight of Result	1.52%	1.303%	1.194%	1.194%	1.194%	1.52%	
-->Calculated Strength	<14.187>	<10.424>	<8.756>	<8.756>	<8.756>	<14.187>	<10.844>
CPSC1105_CanedoJJ_80701_Fall2018.pdf	14/150	12/150	11/150	11/150	11/150	14/150	
-->Words In File	921	921	921	921	921	921	SD:2.436
-->Weight of Result	1.52%	1.303%	1.194%	1.194%	1.194%	1.52%	
-->Calculated Strength	<14.187>	<10.424>	<8.756>	<8.756>	<8.756>	<14.187>	<10.844>
CPSC1105_HuppJ_80705_Fall2018.docx	13/150	15/150	6/150	12/150	8/150	14/150	
-->Words In File	766	766	766	766	766	766	SD:5.925
-->Weight of Result	1.697%	1.958%	0.783%	1.567%	1.044%	1.828%	
-->Calculated Strength	<14.707>	<19.58>	<3.132>	<12.536>	<5.568>	<17.061>	<12.057>
CPSC1105_NguyenT_80702_Fall2018.pdf	15/150	17/150	10/150	12/150	12/150	16/150	
-->Words In File	893	893	893	893	893	893	SD:5.07
-->Weight of Result	1.68%	1.904%	1.12%	1.344%	1.344%	1.792%	
-->Calculated Strength	<16.8>	<21.579>	<7.467>	<10.752>	<10.752>	<19.115>	<14.411>
CPSC1105_SellersC_80704_Fall2018.pdf	13/150	12/150	11/150	11/150	11/150	14/150	
-->Words In File	930	930	930	930	930	930	SD:2.048
-->Weight of Result	1.398%	1.29%	1.183%	1.183%	1.183%	1.505%	
-->Calculated Strength	<12.116>	<10.32>	<8.675>	<8.675>	<8.675>	<14.047>	<10.418>
CPSC1105_SmithA_80703_Fall2018.pdf	0/150	1/150	0/150	1/150	0/150	1/150	
-->Words In File	116	116	116	116	116	116	SD:0.288
-->Weight of Result	0.0%	0.862%	0.0%	0.862%	0.0%	0.862%	
-->Calculated Strength	<0.0>	<0.575>	<0.0>	<0.575>	<0.0>	<0.575>	<0.287>
CPSC1105_WangY_80713_Fall2018.pdf	0/150	3/150	0/150	2/150	1/150	2/150	
-->Words In File	144	144	144	144	144	144	SD:1.464
-->Weight of Result	0.0%	2.083%	0.0%	1.389%	0.694%	1.389%	
-->Calculated Strength	<0.0>	<4.166>	<0.0>	<1.852>	<0.463>	<1.852>	<1.389>
CPSC1105_WangY_81235_Fall2018.pdf	0/150	3/150	0/150	2/150	1/150	2/150	
-->Words In File	144	144	144	144	144	144	SD:1.464
-->Weight of Result	0.0%	2.083%	0.0%	1.389%	0.694%	1.389%	
-->Calculated Strength	<0.0>	<4.166>	<0.0>	<1.852>	<0.463>	<1.852>	<1.389>
CPSC1301L-WangS_80818_Fall2018.docx	15/150	12/150	12/150	13/150	12/150	25/150	
-->Words In File	671	671	671	671	671	671	SD:17.265
-->Weight of Result	2.235%	1.788%	1.788%	1.937%	1.788%	3.726%	
-->Calculated Strength	<22.35>	<14.304>	<14.304>	<16.787>	<14.304>	<62.1>	<24.025>
CPSC1301L_AngelopoulouA_81219_Fall2018.docx	18/150	15/150	14/150	14/150	10/150	21/150	
-->Words In File	795	795	795	795	795	795	SD:9.115
-->Weight of Result	2.264%	1.887%	1.761%	1.761%	1.258%	2.642%	
-->Calculated Strength	<27.168>	<18.87>	<16.436>	<16.436>	<8.387>	<36.988>	<20.714>
CPSC1301L_AngelopoulouA_82575_Fall2018.docx	18/150	15/150	14/150	14/150	10/150	21/150	
-->Words In File	795	795	795	795	795	795	SD:9.115
-->Weight of Result	2.264%	1.887%	1.761%	1.761%	1.258%	2.642%	
-->Calculated Strength	<27.168>	<18.87>	<16.436>	<16.436>	<8.387>	<36.988>	<20.714>
CPSC1301L_CarrollH_80816_Fall2018.docx	15/150	13/150	13/150	15/150	8/150	20/150	
-->Words In File	755	755	755	755	755	755	SD:8.919
-->Weight of Result	1.987%	1.722%	1.722%	1.987%	1.06%	2.649%	
-->Calculated Strength	<19.87>	<14.924>	<14.924>	<19.87>	<5.653>	<35.32>	<18.427>
CPSC1301L_FleenorH_80817_Fall2018.html	14/150	11/150	11/150	13/150	12/150	25/150	
-->Words In File	752	752	752	752	752	752	SD:15.857
-->Weight of Result	1.862%	1.463%	1.463%	1.729%	1.596%	3.324%	
-->Calculated Strength	<17.379>	<10.729>	<10.729>	<14.985>	<12.768>	<55.4>	<20.332>
CPSC1301L_ZhouY_80819_Fall2018.docx	15/150	13/150	13/150	15/150	9/150	20/150	
-->Words In File	757	757	757	757	757	757	SD:8.896
-->Weight of Result	1.982%	1.717%	1.717%	1.982%	1.057%	2.642%	
-->Calculated Strength	<19.82>	<14.881>	<14.881>	<19.82>	<5.637>	<35.227>	<18.378>
CPSC1301L_ZhouY_81215_Fall2018.docx	15/150	13/150	13/150	15/150	8/150	20/150	
-->Words In File	757	757	757	757	757	757	SD:8.896
-->Weight of Result	1.982%	1.717%	1.717%	1.982%	1.057%	2.642%	
-->Calculated Strength	<19.82>	<14.881>	<14.881>	<19.82>	<5.637>	<35.227>	<18.378>

CPSC1301L_ZhouY_82573_Fall2018.docx	15/150	13/150	13/150	15/150	8/150	20/150	
-->Words In File	757	757	757	757	757	757	SD:8.896
-->Weight of Result	1.982%	1.717%	1.717%	1.982%	1.057%	2.642%	
-->Calculated Strength	<19.82>	<14.881>	<14.881>	<19.82>	<5.637>	<35.227>	<18.378>
CPSC1301_AngelopoulouA_81218_Fall2018.pdf	16/150	14/150	12/150	11/150	8/150	21/150	
-->Words In File	995	995	995	995	995	995	SD:8.163
-->Weight of Result	1.608%	1.407%	1.206%	1.106%	0.804%	2.111%	
-->Calculated Strength	<17.152>	<13.132>	<9.648>	<8.111>	<4.288>	<29.554>	<13.648>
CPSC1301_AngelopoulouA_82574_Fall2018.pdf	16/150	14/150	12/150	11/150	8/150	21/150	
-->Words In File	995	995	995	995	995	995	SD:8.163
-->Weight of Result	1.608%	1.407%	1.206%	1.106%	0.804%	2.111%	
-->Calculated Strength	<17.152>	<13.132>	<9.648>	<8.111>	<4.288>	<29.554>	<13.648>
CPSC1301_CarrollH_80809_Fall2018.pdf	15/150	11/150	12/150	12/150	9/150	22/150	
-->Words In File	996	996	996	996	996	996	SD:8.979
-->Weight of Result	1.506%	1.104%	1.205%	1.205%	0.904%	2.209%	
-->Calculated Strength	<15.06>	<8.096>	<9.64>	<9.64>	<5.424>	<32.399>	<13.377>
CPSC1301_FleenorH_80813_Fall2018.html	12/150	10/150	13/150	12/150	12/150	27/150	
-->Words In File	757	757	757	757	757	757	SD:19.41
-->Weight of Result	1.585%	1.321%	1.717%	1.585%	1.585%	3.567%	
-->Calculated Strength	<12.68>	<8.807>	<14.881>	<12.68>	<12.68>	<64.206>	<20.989>
CPSC1301_WangS_80814_Fall2018.docx	20/150	15/150	14/150	12/150	10/150	23/150	
-->Words In File	796	796	796	796	796	796	SD:12.615
-->Weight of Result	2.513%	1.884%	1.759%	1.508%	1.256%	2.889%	
-->Calculated Strength	<33.507>	<18.84>	<16.417>	<12.064>	<8.373>	<44.298>	<22.25>
CPSC1301_ZhouY_80815_Fall2018.docx	15/150	13/150	13/150	15/150	8/150	20/150	
-->Words In File	756	756	756	756	756	756	SD:8.909
-->Weight of Result	1.984%	1.72%	1.72%	1.984%	1.058%	2.646%	
-->Calculated Strength	<19.84>	<14.907>	<14.907>	<19.84>	<5.643>	<35.28>	<18.403>
CPSC1301_ZhouY_81214_Fall2018.docx	15/150	13/150	13/150	15/150	8/150	20/150	
-->Words In File	756	756	756	756	756	756	SD:8.909
-->Weight of Result	1.984%	1.72%	1.72%	1.984%	1.058%	2.646%	
-->Calculated Strength	<19.84>	<14.907>	<14.907>	<19.84>	<5.643>	<35.28>	<18.403>
CPSC1301_ZhouY_82572_Fall2018.docx	15/150	13/150	13/150	15/150	8/150	20/150	
-->Words In File	756	756	756	756	756	756	SD:8.909
-->Weight of Result	1.984%	1.72%	1.72%	1.984%	1.058%	2.646%	
-->Calculated Strength	<19.84>	<14.907>	<14.907>	<19.84>	<5.643>	<35.28>	<18.403>
CPSC1302_RayL_80820_Fall2018.docx	13/150	14/150	13/150	12/150	8/150	18/150	
-->Words In File	744	744	744	744	744	744	SD:6.936
-->Weight of Result	1.747%	1.882%	1.747%	1.613%	1.075%	2.419%	
-->Calculated Strength	<15.141>	<17.565>	<15.141>	<12.904>	<5.733>	<29.028>	<15.919>
CPSC1302_ShushaneR_80821_Fall2018.pdf	20/150	12/150	13/150	10/150	8/150	18/150	
-->Words In File	916	916	916	916	916	916	SD:8.811
-->Weight of Result	2.183%	1.31%	1.419%	1.092%	0.873%	1.965%	
-->Calculated Strength	<29.107>	<10.48>	<12.298>	<7.28>	<4.656>	<23.58>	<14.567>
CPSC2105_LeeS_80822_Fall2018.docx	16/150	11/150	12/150	12/150	7/150	17/150	
-->Words In File	833	833	833	833	833	833	SD:6.527
-->Weight of Result	1.921%	1.321%	1.441%	1.441%	0.84%	2.041%	
-->Calculated Strength	<20.491>	<9.687>	<11.528>	<11.528>	<3.92>	<23.131>	<13.381>
CPSC2105_RogersN_80823_Fall2018.pdf	12/150	12/150	12/150	15/150	7/150	19/150	
-->Words In File	958	958	958	958	958	958	SD:6.71
-->Weight of Result	1.253%	1.253%	1.253%	1.566%	0.731%	1.983%	
-->Calculated Strength	<10.024>	<10.024>	<10.024>	<15.66>	<3.411>	<25.118>	<12.377>
CPSC2106_PekerY_80824_Fall2018.docx	13/150	13/150	10/150	19/150	8/150	18/150	
-->Words In File	697	697	697	697	697	697	SD:10.453
-->Weight of Result	1.865%	1.865%	1.435%	2.726%	1.148%	2.582%	
-->Calculated Strength	<16.163>	<16.163>	<9.567>	<34.529>	<6.123>	<30.984>	<18.921>
CPSC2106_RayL_80863_Fall2018.doc	0/150	0/150	0/150	0/150	0/150	0/150	
-->Words In File	0	0	0	0	0	0	SD:0.0
-->Weight of Result	0%	0%	0%	0%	0%	0%	
-->Calculated Strength	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>
CPSC2108_HodhodR_80887_Fall2018.docx	13/150	11/150	15/150	10/150	7/150	20/150	
-->Words In File	870	870	870	870	870	870	SD:8.712
-->Weight of Result	1.494%	1.264%	1.724%	1.149%	0.805%	2.299%	
-->Calculated Strength	<12.948>	<9.269>	<17.24>	<7.66>	<3.757>	<30.653>	<13.588>
CPSC2108_PerezA_80895_Fall2018.docx	14/150	11/150	15/150	9/150	8/150	20/150	
-->Words In File	930	830	830	830	830	830	SD:9.125
-->Weight of Result	1.687%	1.325%	1.807%	1.084%	0.964%	2.41%	
-->Calculated Strength	<15.745>	<9.717>	<18.07>	<6.504>	<5.141>	<32.133>	<14.552>
CPSC2125_KhanS_80896_Fall2018.pdf	8/150	6/150	8/150	8/150	10/150	11/150	
-->Words In File	719	719	719	719	719	719	SD:2.575
-->Weight of Result	1.113%	0.834%	1.113%	1.113%	1.391%	1.53%	
-->Calculated Strength	<5.936>	<3.336>	<5.936>	<5.936>	<9.273>	<11.22>	<6.939>
CPSC3106_BarkerM_80897_Fall2018.pdf	14/150	10/150	14/150	16/150	5/150	15/150	
-->Words In File	979	979	979	979	979	979	SD:5.391
-->Weight of Result	1.43%	1.021%	1.43%	1.634%	0.511%	1.532%	
-->Calculated Strength	<13.347>	<6.807>	<13.347>	<17.429>	<1.703>	<15.32>	<11.325>
CPSC3108_PekerY_80898_Fall2018.docx	15/150	14/150	11/150	16/150	10/150	21/150	
-->Words In File	656	656	656	656	656	656	SD:11.352
-->Weight of Result	2.287%	2.134%	1.677%	2.439%	1.524%	3.201%	
-->Calculated Strength	<22.87>	<19.917>	<12.298>	<26.016>	<10.16>	<44.814>	<22.679>

CPSC3111_RogersN_80899_Fall2018.pdf	19/150	17/150	16/150	18/150	11/150	24/150	
-->Words In File	560	560	560	560	560	560	SD:16.268
-->Weight of Result	3.393%	3.036%	2.857%	3.214%	1.964%	4.286%	
-->Calculated Strength	<42.978>	<34.408>	<30.475>	<38.568>	<14.403>	<68.576>	<38.235>
CPSC3116_RogersN_80900_Fall2018.pdf	17/150	18/150	14/150	17/150	8/150	21/150	
-->Words In File	510	510	510	510	510	510	SD:15.164
-->Weight of Result	3.333%	3.529%	2.745%	3.333%	1.569%	4.118%	
-->Calculated Strength	<37.774>	<42.348>	<25.62>	<37.774>	<8.368>	<57.652>	<34.923>
CPSC3125_CarrollH_80901_Fall2018.pdf	13/150	15/150	12/150	12/150	7/150	23/150	
-->Words In File	990	990	990	990	990	990	SD:10.223
-->Weight of Result	1.313%	1.515%	1.212%	1.212%	0.707%	2.323%	
-->Calculated Strength	<11.379>	<15.15>	<9.696>	<9.696>	<3.299>	<35.619>	<14.14>
CPSC3125_CarrollH_80902_Fall2018.pdf	13/150	15/150	12/150	12/150	7/150	23/150	
-->Words In File	990	990	990	990	990	990	SD:10.223
-->Weight of Result	1.313%	1.515%	1.212%	1.212%	0.707%	2.323%	
-->Calculated Strength	<11.379>	<15.15>	<9.696>	<9.696>	<3.299>	<35.619>	<14.14>
CPSC3131_Lee,Y_80903_Fall2018.docx	13/150	12/150	17/150	17/150	11/150	23/150	
-->Words In File	605	605	605	605	605	605	SD:15.26
-->Weight of Result	2.149%	1.983%	2.81%	2.81%	1.818%	3.802%	
-->Calculated Strength	<18.625>	<15.864>	<31.847>	<31.847>	<13.332>	<58.297>	<28.302>
CPSC3131_Lee,Y_80904_Fall2018.docx	14/150	7/150	19/150	17/150	12/150	19/150	
-->Words In File	661	661	661	661	661	661	SD:11.675
-->Weight of Result	2.118%	1.059%	2.874%	2.572%	1.815%	2.874%	
-->Calculated Strength	<19.768>	<4.942>	<36.404>	<29.149>	<14.52>	<36.404>	<23.531>
CPSC3165_Hupp_80906_Fall2018.docx	14/150	12/150	11/150	14/150	6/150	19/150	
-->Words In File	782	782	782	782	782	782	SD:8.431
-->Weight of Result	1.79%	1.535%	1.407%	1.79%	0.767%	2.43%	
-->Calculated Strength	<16.707>	<12.28>	<10.318>	<16.707>	<3.068>	<30.78>	<14.977>
CPSC3175_ObandoR_80907_Fall2018.docx	15/150	12/150	12/150	12/150	10/150	24/150	
-->Words In File	751	751	751	751	751	751	SD:14.427
-->Weight of Result	1.997%	1.598%	1.598%	1.598%	1.332%	3.196%	
-->Calculated Strength	<19.97>	<12.784>	<12.784>	<12.784>	<8.88>	<51.136>	<19.723>
CPSC3415_ObandoR_82106_Fall2018.docx	17/150	15/150	12/150	12/150	8/150	20/150	
-->Words In File	701	701	701	701	701	701	SD:10.46
-->Weight of Result	2.425%	2.14%	1.712%	1.712%	1.141%	2.853%	
-->Calculated Strength	<27.483>	<21.4>	<13.696>	<13.696>	<6.085>	<38.04>	<20.067>
CPSC3555_FleenorH_80637_Fall2018.html	9/150	11/150	8/150	10/150	4/150	14/150	
-->Words In File	722	722	722	722	722	722	SD:5.098
-->Weight of Result	1.247%	1.524%	1.108%	1.385%	0.554%	1.939%	
-->Calculated Strength	<7.482>	<11.176>	<5.909>	<9.233>	<1.477>	<18.097>	<8.896>
CPSC4000_SummersW_80908_Fall2018.docx	15/150	18/150	14/150	15/150	9/150	28/150	
-->Words In File	301	301	301	301	301	301	SD:49.906
-->Weight of Result	4.983%	5.98%	4.651%	4.983%	2.99%	9.302%	
-->Calculated Strength	<49.83>	<71.76>	<43.409>	<49.83>	<17.94>	<173.637>	<67.734>
CPSC4111_ObandoR_80909_Fall2018.docx	14/150	11/150	9/150	11/150	7/150	21/150	
-->Words In File	764	764	764	764	764	764	SD:11.349
-->Weight of Result	1.832%	1.44%	1.178%	1.44%	0.916%	2.749%	
-->Calculated Strength	<17.099>	<10.56>	<7.068>	<10.56>	<4.275>	<38.486>	<14.675>
CPSC4121_ZamsteinL_80910_Fall2018.doc	0/150	0/150	0/150	0/150	0/150	0/150	
-->Words In File	0	0	0	0	0	0	SD:0.0
-->Weight of Result	0%	0%	0%	0%	0%	0%	
-->Calculated Strength	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>
CPSC4125_Smith_80911_Fall2018.pdf	0/150	0/150	0/150	0/150	0/150	0/150	
-->Words In File	168	168	168	168	168	168	SD:0.0
-->Weight of Result	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	
-->Calculated Strength	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>
CPSC4175_ZhouY_80912_Fall2018.docx	9/150	9/150	14/150	12/150	11/150	16/150	
-->Words In File	843	843	843	843	843	843	SD:4.967
-->Weight of Result	1.068%	1.068%	1.661%	1.423%	1.305%	1.898%	
-->Calculated Strength	<6.408>	<6.408>	<15.503>	<11.384>	<9.57>	<20.245>	<11.586>
CPSC4205_SummersW_80913_Fall2018.docx	0/150	0/150	0/150	0/150	0/150	0/150	
-->Words In File	0	0	0	0	0	0	SD:0.0
-->Weight of Result	0%	0%	0%	0%	0%	0%	
-->Calculated Strength	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>
CPSC4505_PekerY_80915_Fall2018.docx	14/150	12/150	10/150	16/150	7/150	17/150	
-->Words In File	680	680	680	680	680	680	SD:8.243
-->Weight of Result	2.059%	1.765%	1.471%	2.353%	1.029%	2.5%	
-->Calculated Strength	<19.217>	<14.12>	<9.807>	<25.099>	<4.802>	<28.333>	<16.896>
CPSC4698_FleenorH_82795_Fall 2018.doc	0/150	0/150	0/150	0/150	0/150	0/150	
-->Words In File	0	0	0	0	0	0	SD:0.0
-->Weight of Result	0%	0%	0%	0%	0%	0%	
-->Calculated Strength	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>	<0.0>
CPSC5115U_LeeS_80918_Fall2018.docx	10/150	6/150	9/150	9/150	6/150	18/150	
-->Words In File	823	823	823	823	823	823	SD:8.0
-->Weight of Result	1.215%	0.729%	1.094%	1.094%	0.729%	2.187%	
-->Calculated Strength	<8.1>	<2.916>	<6.564>	<6.564>	<2.916>	<26.244>	<8.884>
CPSC5125U_ObandoR_80919_Fall2018.docx	19/150	14/150	14/150	11/150	8/150	20/150	
-->Words In File	761	761	761	761	761	761	SD:10.581
-->Weight of Result	2.497%	1.84%	1.84%	1.445%	1.051%	2.628%	
-->Calculated Strength	<31.629>	<17.173>	<17.173>	<10.597>	<5.605>	<35.04>	<19.536>

CPSC5127U_WangL_80920_Fall2018.docx	17/150	13/150	13/150	14/150	8/150	21/150	
-->Words In File	588	588	588	588	588	588	SD:13.389
-->Weight of Result	2.891%	2.211%	2.211%	2.381%	1.361%	3.571%	
-->Calculated Strength	<32.765>	<19.162>	<19.162>	<22.223>	<7.259>	<49.994>	<25.094>
CPSC5135U_WoolbrightD_80921_Fall2018.docx	7/150	10/150	7/150	6/150	6/150	15/150	
-->Words In File	457	457	457	457	457	457	SD:9.817
-->Weight of Result	1.532%	2.188%	1.532%	1.313%	1.313%	3.282%	
-->Calculated Strength	<7.149>	<14.587>	<7.149>	<5.252>	<5.252>	<32.82>	<12.035>
CPSC5155U_WangS_80924_Fall2018.docx	14/150	11/150	17/150	12/150	8/150	19/150	
-->Words In File	833	833	833	833	833	833	SD:8.111
-->Weight of Result	1.681%	1.321%	2.041%	1.441%	0.96%	2.281%	
-->Calculated Strength	<15.689>	<9.687>	<23.131>	<11.528>	<5.12>	<28.893>	<15.675>
CPSC5157U_YangJ_81040_Fall2018.docx	16/150	14/150	18/150	20/150	12/150	19/150	
-->Words In File	714	714	714	714	714	714	SD:8.439
-->Weight of Result	2.241%	1.961%	2.521%	2.801%	1.681%	2.661%	
-->Calculated Strength	<23.904>	<18.303>	<30.252>	<37.347>	<13.448>	<33.706>	<26.16>
CPSC5157U_YangJ_82214_Fall2018.docx	16/150	14/150	18/150	20/150	12/150	19/150	
-->Words In File	713	713	713	713	713	713	SD:8.452
-->Weight of Result	2.244%	1.964%	2.525%	2.805%	1.683%	2.665%	
-->Calculated Strength	<23.936>	<18.331>	<30.3>	<37.4>	<13.464>	<33.757>	<26.198>
CPSC6000_KhanS_82251_Fall2018.pdf	12/150	11/150	7/150	12/150	7/150	16/150	
-->Words In File	422	422	422	422	422	422	SD:11.061
-->Weight of Result	2.844%	2.607%	1.659%	2.844%	1.659%	3.791%	
-->Calculated Strength	<22.752>	<19.118>	<7.742>	<22.752>	<7.742>	<40.437>	<20.09>
CPSC6105_YangJ_80928_Fall2018.docx	20/150	19/150	21/150	24/150	12/150	23/150	
-->Words In File	637	637	637	637	637	637	SD:14.548
-->Weight of Result	3.14%	2.983%	3.297%	3.768%	1.884%	3.611%	
-->Calculated Strength	<41.867>	<37.785>	<46.158>	<60.288>	<15.072>	<55.369>	<42.756>
CPSC6105_YangJ_81041_Fall2018.docx	20/150	19/150	20/150	24/150	12/150	23/150	
-->Words In File	641	641	641	641	641	641	SD:14.377
-->Weight of Result	3.12%	2.964%	3.12%	3.744%	1.872%	3.588%	
-->Calculated Strength	<41.6>	<37.544>	<41.6>	<59.904>	<14.976>	<55.016>	<41.773>
CPSC6106_HodhodR_80929_Fall2018.docx	17/150	10/150	14/150	11/150	10/150	20/150	
-->Words In File	842	842	842	842	842	842	SD:8.802
-->Weight of Result	2.019%	1.188%	1.663%	1.306%	1.188%	2.375%	
-->Calculated Strength	<22.882>	<7.92>	<15.521>	<9.577>	<7.92>	<31.667>	<15.914>
CPSC6106_HodhodR_81042_Fall2018.docx	17/150	10/150	14/150	11/150	10/150	20/150	
-->Words In File	838	838	838	838	838	838	SD:8.848
-->Weight of Result	2.029%	1.193%	1.671%	1.313%	1.193%	2.387%	
-->Calculated Strength	<22.995>	<7.953>	<15.596>	<9.629>	<7.953>	<31.827>	<15.992>
CPSC6107_AngelopoulouA_80930_Fall2018.pdf	13/150	12/150	13/150	11/150	7/150	19/150	
-->Words In File	959	959	959	959	959	959	SD:6.606
-->Weight of Result	1.356%	1.251%	1.356%	1.147%	0.73%	1.981%	
-->Calculated Strength	<11.752>	<10.008>	<11.752>	<8.411>	<3.407>	<25.093>	<11.737>
CPSC6107_AngelopoulouA_81043_Fall2018.pdf	13/150	12/150	13/150	11/150	7/150	19/150	
-->Words In File	959	959	959	959	959	959	SD:6.606
-->Weight of Result	1.356%	1.251%	1.356%	1.147%	0.73%	1.981%	
-->Calculated Strength	<11.752>	<10.008>	<11.752>	<8.411>	<3.407>	<25.093>	<11.737>
CPSC6109_LeeS_80931_Fall2018.docx	16/150	8/150	13/150	10/150	7/150	21/150	
-->Words In File	937	937	937	937	937	937	SD:9.678
-->Weight of Result	1.708%	0.854%	1.387%	1.067%	0.747%	2.241%	
-->Calculated Strength	<18.219>	<4.555>	<12.021>	<7.113>	<3.486>	<31.374>	<12.795>
CPSC6109_LeeS_81044_Fall2018.docx	16/150	8/150	13/150	10/150	7/150	21/150	
-->Words In File	937	937	937	937	937	937	SD:9.678
-->Weight of Result	1.708%	0.854%	1.387%	1.067%	0.747%	2.241%	
-->Calculated Strength	<18.219>	<4.555>	<12.021>	<7.113>	<3.486>	<31.374>	<12.795>
CPSC6118_PerezA_80932_Fall2018.docx	17/150	16/150	13/150	11/150	8/150	21/150	
-->Words In File	746	746	746	746	746	746	SD:11.04
-->Weight of Result	2.279%	2.145%	1.743%	1.475%	1.072%	2.815%	
-->Calculated Strength	<25.829>	<22.88>	<15.106>	<10.817>	<5.717>	<39.41>	<19.96>
CPSC6119_WangL_80933_Fall2018.docx	19/150	10/150	16/150	13/150	10/150	27/150	
-->Words In File	712	712	712	712	712	712	SD:20.431
-->Weight of Result	2.669%	1.404%	2.247%	1.826%	1.404%	3.792%	
-->Calculated Strength	<33.807>	<9.36>	<23.968>	<15.825>	<9.36>	<68.256>	<26.763>
CPSC6119_WangL_81046_Fall2018.docx	19/150	10/150	16/150	13/150	10/150	27/150	
-->Words In File	713	713	713	713	713	713	SD:20.402
-->Weight of Result	2.665%	1.403%	2.244%	1.823%	1.403%	3.787%	
-->Calculated Strength	<33.757>	<9.353>	<23.936>	<15.799>	<9.353>	<68.166>	<26.727>
CPSC6125_ChouchaneR_80934_Fall2018.pdf	14/150	15/150	15/150	11/150	7/150	16/150	
-->Words In File	913	913	913	913	913	913	SD:5.223
-->Weight of Result	1.533%	1.643%	1.643%	1.205%	0.767%	1.752%	
-->Calculated Strength	<14.308>	<16.43>	<16.43>	<8.837>	<3.579>	<18.688>	<13.045>
CPSC6125_ChouchaneR_81047_Fall2018.pdf	14/150	15/150	15/150	11/150	7/150	16/150	
-->Words In File	913	913	913	913	913	913	SD:5.223
-->Weight of Result	1.533%	1.643%	1.643%	1.205%	0.767%	1.752%	
-->Calculated Strength	<14.308>	<16.43>	<16.43>	<8.837>	<3.579>	<18.688>	<13.045>
CPSC6126_PekerY_80935_Fall2018.docx	13/150	14/150	13/150	17/150	6/150	20/150	
-->Words In File	684	684	684	684	684	684	SD:10.981
-->Weight of Result	1.901%	2.047%	1.901%	2.485%	0.877%	2.924%	
-->Calculated Strength	<16.475>	<19.105>	<16.475>	<28.163>	<3.508>	<38.987>	<20.452>

CPSC6126_PekerY_81048_Fall2018.docx	13/150	14/150	13/150	17/150	6/150	20/150	
-->Words In File	684	684	684	684	684	684	SD:10.981
-->Weight of Result	1.901%	2.047%	1.901%	2.485%	0.877%	2.924%	
-->Calculated Strength	<16.475>	<19.105>	<16.475>	<28.163>	<3.508>	<38.987>	<20.452>
CPSC6129_ChouchaneR_80936_Fall2018.pdf	14/150	12/150	14/150	11/150	11/150	23/150	
-->Words In File	936	936	936	936	936	936	SD:10.154
-->Weight of Result	1.496%	1.282%	1.496%	1.175%	1.175%	2.457%	
-->Calculated Strength	<13.963>	<10.256>	<13.963>	<8.617>	<8.617>	<37.674>	<15.515>
CPSC6129_ChouchaneR_81049_Fall2018.pdf	14/150	12/150	14/150	11/150	11/150	23/150	
-->Words In File	936	936	936	936	936	936	SD:10.154
-->Weight of Result	1.496%	1.282%	1.496%	1.175%	1.175%	2.457%	
-->Calculated Strength	<13.963>	<10.256>	<13.963>	<8.617>	<8.617>	<37.674>	<15.515>
CPSC6136_RayL_80937_Fall2018.docx	15/150	14/150	11/150	20/150	8/150	24/150	
-->Words In File	441	441	441	441	441	441	SD:26.359
-->Weight of Result	3.401%	3.175%	2.494%	4.535%	1.814%	5.442%	
-->Calculated Strength	<34.01>	<29.633>	<18.289>	<60.467>	<9.675>	<87.072>	<39.858>
CPSC6136_RayL_81050_Fall2018.docx	15/150	14/150	11/150	20/150	8/150	24/150	
-->Words In File	441	441	441	441	441	441	SD:26.359
-->Weight of Result	3.401%	3.175%	2.494%	4.535%	1.814%	5.442%	
-->Calculated Strength	<34.01>	<29.633>	<18.289>	<60.467>	<9.675>	<87.072>	<39.858>
CPSC6148_AngelopoulouA_80938_Fall2018.pdf	7/150	7/150	10/150	13/150	6/150	16/150	
-->Words In File	867	867	867	867	867	867	SD:6.103
-->Weight of Result	0.807%	0.807%	1.153%	1.499%	0.692%	1.845%	
-->Calculated Strength	<3.766>	<3.766>	<7.687>	<12.991>	<2.768>	<19.68>	<8.443>
CPSC6148_AngelopoulouA_82197_Fall2018.pdf	7/150	7/150	10/150	13/150	6/150	16/150	
-->Words In File	867	867	867	867	867	867	SD:6.103
-->Weight of Result	0.807%	0.807%	1.153%	1.499%	0.692%	1.845%	
-->Calculated Strength	<3.766>	<3.766>	<7.687>	<12.991>	<2.768>	<19.68>	<8.443>
CPSC6157_WangL_80940_Fall2018.docx	19/150	15/150	16/150	13/150	9/150	22/150	
-->Words In File	739	739	739	739	739	739	SD:11.756
-->Weight of Result	2.571%	2.03%	2.165%	1.759%	1.218%	2.977%	
-->Calculated Strength	<32.566>	<20.3>	<23.093>	<15.245>	<7.308>	<43.663>	<23.696>
CPSC6157_WangL_82199_Fall2018.docx	19/150	16/150	16/150	13/150	9/150	22/150	
-->Words In File	739	739	739	739	739	739	SD:11.667
-->Weight of Result	2.571%	2.165%	2.165%	1.759%	1.218%	2.977%	
-->Calculated Strength	<32.566>	<23.093>	<23.093>	<15.245>	<7.308>	<43.663>	<24.161>
CPSC6177_PerezA_80941_Fall2018.docx	14/150	15/150	11/150	12/150	7/150	17/150	
-->Words In File	860	860	860	860	860	860	SD:5.97
-->Weight of Result	1.628%	1.744%	1.279%	1.395%	0.814%	1.977%	
-->Calculated Strength	<15.195>	<17.44>	<9.379>	<11.16>	<3.799>	<22.406>	<13.23>
CPSC6177_PerezA_82200_Fall2018.docx	14/150	15/150	11/150	12/150	7/150	17/150	
-->Words In File	860	860	860	860	860	860	SD:5.97
-->Weight of Result	1.628%	1.744%	1.279%	1.395%	0.814%	1.977%	
-->Calculated Strength	<15.195>	<17.44>	<9.379>	<11.16>	<3.799>	<22.406>	<13.23>
CPSC6178_Khan_Fall_2018.pdf	9/150	7/150	11/150	12/150	9/150	18/150	
-->Words In File	934	934	934	934	934	934	SD:6.463
-->Weight of Result	0.964%	0.749%	1.178%	1.285%	0.964%	1.927%	
-->Calculated Strength	<5.784>	<3.495>	<8.639>	<10.28>	<5.784>	<23.124>	<9.518>
CPSC6899_PekerY_82205_Fall2018.docx	14/150	12/150	10/150	16/150	7/150	17/150	
-->Words In File	680	680	680	680	680	680	SD:8.243
-->Weight of Result	2.059%	1.765%	1.471%	2.353%	1.029%	2.5%	
-->Calculated Strength	<19.217>	<14.12>	<9.807>	<25.099>	<4.802>	<28.333>	<16.896>
CPSC6985_HodhodR_82469_Fall2018.docx	16/150	16/150	15/150	18/150	7/150	23/150	
-->Words In File	714	714	714	714	714	714	SD:13.269
-->Weight of Result	2.241%	2.241%	2.101%	2.521%	0.98%	3.221%	
-->Calculated Strength	<23.904>	<23.904>	<21.01>	<30.252>	<4.573>	<49.389>	<25.505>
CPSC6985_KhanS_82582_Fall2018.pdf	9/150	17/150	9/150	13/150	6/150	18/150	
-->Words In File	349	349	349	349	349	349	SD:20.84
-->Weight of Result	2.579%	4.871%	2.579%	3.725%	1.719%	5.158%	
-->Calculated Strength	<15.474>	<55.205>	<15.474>	<32.283>	<6.876>	<61.896>	<31.201>
CPSC6985_WangL_82523_Fall2018.docx	14/150	13/150	13/150	16/150	6/150	20/150	
-->Words In File	537	537	537	537	537	537	SD:13.585
-->Weight of Result	2.607%	2.421%	2.421%	2.98%	1.117%	3.724%	
-->Calculated Strength	<24.332>	<20.982>	<20.982>	<31.787>	<4.468>	<49.653>	<25.367>
CSMT6222_WangS_81052_Fall2018.docx	13/150	11/150	14/150	13/150	8/150	17/150	
-->Words In File	827	827	827	827	827	827	SD:5.553
-->Weight of Result	1.572%	1.33%	1.693%	1.572%	0.967%	2.056%	
-->Calculated Strength	<13.624>	<9.753>	<15.801>	<13.624>	<5.157>	<23.301>	<13.543>
HONS3555_KhanS_81649_Fall2018.pdf	9/150	13/150	9/150	15/150	10/150	15/150	
-->Words In File	682	682	682	682	682	682	SD:6.125
-->Weight of Result	1.32%	1.906%	1.32%	2.199%	1.466%	2.199%	
-->Calculated Strength	<7.92>	<16.519>	<7.92>	<21.99>	<9.773>	<21.99>	<14.352>
MISM4126_HodhodR_81228_Fall2018.docx	11/150	9/150	12/150	12/150	11/150	22/150	
-->Words In File	361	361	361	361	361	361	SD:25.2
-->Weight of Result	3.047%	2.493%	3.324%	3.324%	3.047%	6.094%	
-->Calculated Strength	<22.345>	<14.958>	<26.592>	<26.592>	<22.345>	<89.379>	<33.702>
WBIT2000_GarvinC_81329_Fall2018.docx	8/150	11/150	12/150	14/150	8/150	19/150	
-->Words In File	818	818	818	818	818	818	SD:8.278
-->Weight of Result	0.978%	1.345%	1.467%	1.711%	0.978%	2.323%	
-->Calculated Strength	<5.216>	<9.863>	<11.736>	<15.969>	<5.216>	<29.425>	<12.904>

WBIT3500_Bhagyavait_81330_Fall2018.pdf	10/150	14/150	9/150	10/150	7/150	18/150	
-->Words In File	573	573	573	573	573	573	SD:10.828
-->Weight of Result	1.745%	2.443%	1.571%	1.745%	1.222%	3.141%	
-->Calculated Strength	<11.633>	<22.801>	<9.426>	<11.633>	<5.703>	<37.692>	<16.481>
WBIT4120_KhanS_81331_Fall2018.pdf	13/150	12/150	11/150	13/150	7/150	20/150	
-->Words In File	728	728	728	728	728	728	SD:9.922
-->Weight of Result	1.786%	1.648%	1.511%	1.786%	0.962%	2.747%	
-->Calculated Strength	<15.479>	<13.184>	<11.081>	<15.479>	<4.489>	<36.627>	<16.056>
WBIT4520_Bhagyavati_81332_Fall2018.pdf	10/150	13/150	12/150	9/150	7/150	19/150	
-->Words In File	1041	1041	1041	1041	1041	1041	SD:6.529
-->Weight of Result	0.961%	1.249%	1.153%	0.865%	0.672%	1.825%	
-->Calculated Strength	<6.407>	<10.825>	<9.224>	<5.19>	<3.136>	<23.117>	<9.65>
-->Standard Deviation	10.265	10.642	9.047	12.991	4.094	22.986	
-->Average	<17.729>	<14.944>	<13.812>	<16.742>	<6.503>	<34.322>	

I have submitted this thesis in partial fulfillment of the requirements for the degree of Master of Science

5/14/2019
Date

Daniel Rockwell

We approve the thesis of Daniel P. Rockwell as presented here.

5/20/19
Date

Shamim Khan
Professor of Computer Science, Thesis Advisor

5/15/19
Date

Rania Hodhod
Assistant Professor of Computer Science

5/14/2019
Date

Kyongseon Jeon
Professor of English

5/15/19
Date

Wayne Summers
Chair of TSYS School of Computer Science