Tom Richmond

Dr. Mike Heroux

10/11/17

CSCI 373

Title and Abstract Exercise

1. The title of this article is catchy but long. Develop one or two titles that retain your interest but are shorter.

What’s new?: Web evolution from a search engine perspective

1. The above abstract has 249 words (can you guess the imposed limit?). Using the Koopman and van Leunen/Lipton articles as guides, do the following:
   1. Identify the missing elements (according to Koopman).

Missing Elements: Problem Statement

* 1. Add these elements.

“With an ever-evolving web, it is becoming increasingly difficult for search engines to provide the most up-to-date result for its users”

* 1. Find the word count and reduce it by 50 words, without losing significant meaning.

Current count: 271 Words

50 word-reduced-abstract (221 Words):

An ever-evolving web makes it increasingly difficult for search engines to provide the most up-to-date result for its users. With our societal dependency on search engines, it is in our interest to find the solution to this problem to improve search engine results. We collected weekly snapshots of 150 Web sites over one year, measuring the evolution of content and link structure. Our measurements focus on aspects which are potentially interesting to search engine designers: link structure evolution over time, the creation rate of new pages and distinct content on the Web, and the rate of content change on existing pages under search-centric measures of degree of change. Our findings indicate a rapid turnover rate of Web pages, including birth and death rates, and hyperlink turnover. For pages that persist over time, we surprisingly found that the degree of content shift as measured using TF.IDF cosine distance does not appear to be consistently correlated with content update frequency. Despite this, the content shift rate of a given page is likely to remain consistent over time. That is, pages that change a great deal in one week will likely change by a similarly large degree in the following week, and vice versa. We conclude with a discussion of the potential implications of our results for the design of effective Web search engines.

* 1. Reduce it by 50 more words, or even more if possible.

100 word-reduced-abstract (171 Words):

An ever-evolving web makes it increasingly difficult for search engines to provide the most up-to-date result. With growing dependency on search engines, finding the solution to this problem and improve search engine results is in our interest. Weekly snapshots were collected from 150 sites over a year, measuring content and link structure evolution. Our measurements focus on potentially interesting aspects to search engine designers: link structure evolution, the creation rate of new pages and distinct content on the Web, and the rate of content change on existing pages. Our findings indicate a rapid turnover rate of Web pages, including birth and death rates, and hyperlink turnover. For pages that persisted, we surprisingly found that the degree of content shift, measured using TF.IDF cosine distance, does not appear to be correlated with content update frequency. Despite this, a given page’s content shift rate is likely to remain consistent on a weekly basis. We conclude with a discussion of the potential implications of our results for the design of effective Web search engines.

1. You have developed a new sorting algorithm for a general list of integers. Your algorithm is *O(n)*. Wow, you are good! The previous best algorithm is a bubble sort, *O(?).* Write a 100-word abstract for a technical article you will submit for publication.

Increasing the efficiency of sorting algorithms has been a topic of discussion for a long time, with the bubble sort previously providing the best solution. The improvement of such sorting algorithms could greatly improve overall performance in many programs that rely upon such algorithms. After analyzing many different sorting algorithms, the Richmond Algorithm has been designed as a way to improve the efficiency of such algorithms to O(n) on a general list of integers. This algorithm’s implications upon the computer science field can drastically change the way programs implement their sorting needs.

1. Write a 200-word-or-fewer abstract for your project. Keep for your final State-of-the-field paper.

The desire to teach a computer how to algorithmically compose music has been a topic in the world of computer science since the 1950’s. One of the biggest limitations of algorithmically composing music to date has been the difficulty of fully automating the process, and removing human intervention at any level. We attempt to remedy this issue by teaching a computer how to learn the rules of composition itself by examining musical scores, rather than explicitly telling the computer the rules. To pursue this automation of the algorithmic composition process, we examined the intersectionality of algorithmic composition with the machine learning concept of classification. Using a Naïve Bayes system, the computer learns to classify pieces of classical music based upon their distinct era, and it then attempts to primitively recreate the musical styles using a technique inspired by cellular-automata. The success of this process is determined by feeding its compositions back into the classifier. It is concluded that the process shows the potential of further hybridizing classification techniques with composition techniques, potentially further removing human intervention from the process, and the results successfully replicate the features that our classifiers looked for within each distinct era.