Title of Work:
LaSEWeb: Automating Search Strategies over Semi-structured Web Data

Conference:
Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

Rationale to ensure venue quality:
The Association for Computing Machinery (ACM) is one of the most well-known and well recognized venues with regards to computing research. As such, inclusion in an ACM conference can be an indicator of relatively high quality in its own right. Specifically, KDD is referred to as a "premier interdisciplinary conference [that] brings together researchers and practitioners from data science, data mining, knowledge discovery, large-scale data analytics, and big data." This conference, the 20th iteration, featured 5 keynotes, almost 200 papers, 8 talks, 12 tutorials, 25 workshops, a special computing competition, and more. It is likely that a conference of this size that has been running for so long is a reasonably reputable venue. Additionally, the conference was supported by and featured talks from the Allen Institute for Artificial Intelligence, Microsoft, Harvard, & Bloomberg.

Problem Statement:
Frequently, there is a disconnect between the way web data is stored and the way users use natural language to express a question. This can make can complicate information retrieval in two ways – first, a computer may not be able to effectively understand user instructions and second, a user may struggle in extracting their desired answer from the information presented by the machine. Polozov & Gulwani explore a domain-specific language that enables data extraction from a webpage based on structure, layout, and linguistic patterns that better allows the expression of data and information between users and machines, with an algorithm that can rank multiple answers extracted from multiple web pages. This can allow for automation and improved effectiveness of search strategies, and for improved comprehension of results for users.

Paper Synopsis:
Polozov and Gulwani begin by establishing the search tendencies of users interacting with online search engines. Generally, users of search engines tend to structure queries in two formats – a "process batch data" query and a "factoid question" query. Process batch data queries involve making a series of similar queries to a search engine to find a similar piece of information about multiple items in a list. Factoid questions are queries that ask a specific question with one specific answer (such as "Who invented radio?"). Users frequently create search queries with high recall, but not with high precision (meaning that their query will usually return the answer they're looking for, but may not rank the answer or site containing the answer high enough to be easily found). Users have become

accustomed to navigating through a selected set of links (generally the top ten) and exploring the page content to build a list of answer candidates, using a set of patterns that form the basis of a "search strategy."

A basic search strategy is comprised of three types of patterns – linguistic patterns, structural patterns, and visual patterns. Polozov and Gulwani state that these three patterns serve as a way to create structure and add semantics on top of semi-structured web content and HTML. Linguistic patterns are comprised of the textual content of the webpage, along with it's sentence structure and semantic information. Structural patterns are comprised of any relational or otherwise semi-structured information on a page, such as tables and lists. Visual patterns are comprised of the layout and styling of the page, focusing on how the page is presented and what content is emphasized. Polozov and Gulwani, in an attempt to automate repetitive search tasks (i.e. finding information on a web page) involving the above patterns, created LaSEWeb (**La**nguage for **S**tructure **E**xtraction). LaSEWeb consists of domain specific language for programming search strategies, along with an accompanying interpreter, built on the Bing search engine.

LaSEWeb takes in a set of tuples of user query arguments and returns a set of answer strings (the result of Bing searches) with confidence scores. It does so by exploring a list of search results for a "seed query," and executes the LaSEWeb query on a webpage returned as a result to the query. This extracts multiple multiple answer string representations and then clusters the representations together using an application-specific similarity function. Using the semi-structured content of the webpage and CSS attributes of HTML nodes, it extracts textual information and applies NLP and several algorithms to identify and extract the most relevant sections of the page. LaSEWeb achieved 95% precision across seven microsegments of factoid searches and 73% precision in the process batch data searches, with an average recall of 71% over 100,000 user queries obtained from Bing search logs.

The LaSEWeb query itself is executed against a web page, evaluating 2 auxiliary functions for each HTML node. The "BBox" function is the smallest bounding box of a node on the page and the "Text" function is the text inside it stripped of HTML tags. This generates a multi-set of possible answer strings, each labeled with relevant meta-information about the page. The text is tokenized, parsed, and then classified by POS, entity, and synonym status. Structurally, semi-structured HTML nodes are used to "recover" an implicit tabular structure that matches attributes with their values. These pairings are then indexed and parsed for relevancy based on NLP extraction techniques. Visually, the bounding boxes are used to parse and separate content based on their visual location and the way they're displayed on the page, checking bounding boxes for proximity, alignment, and emphasis as compared to other bounding boxes. To evaluate LaSEWeb, Polozov and Gulwani extracted 100,000 user queries across 7 micro-segments from Bing Search logs, using regular expressions to extract user queries that belong to certain micro-segments. The number of queries and recall were calculated, and found to be much more effective than Bing on its own. Finally, they examined five categories of

repeatable search tasks, evaluating how many data entries in the set it could correctly extract and answer.

Future Work:

An interesting future study would involve applying these results to query data obtained from the search logs of competing search engines (mainly Google, but others such as Yahoo and DuckDuckGo could be searched as well) to determine if the search engine has any impact on the query structure and efficiency. Additionally, Google has begun implementing functionality to directly answer a question if possible (search “Who invented radio” on Google) rather than just returning a site. Evaluating the LaSEWeb against these direct answer results rather than against just web sites could offer valuable information within the factoid question search domain.