1. Introduction

The World Wide Web (WWW) gives users access to billions of interconnected documents on the internet today. Attempts to gain better access to this information was made with the advent of modern day web search engines like Google and Yahoo. The commercial success of these search engines have been a testament to how well they satisfy their use case: providing highly relevant content quickly. Although great for getting website links or carrying out non extensive research on a topic, search engines fail to provide relevant content for extensive & time consuming research. The focus on first page precision rather than recall makes it harder to get extensive data on a specific domain. Thus, carrying out long term research projects turns out to be quite a manual and time consuming task where the user makes multiple queries to the search engine over days of research in the hope of finding new & relevant content. Our project aims to build an extensive data extraction tool that crawls the web to gather and structure data from a specific domain using a knowledge discovery graph created by the user(Fig 1 Example of a knowledge discovery graph).

Although attempts have been made to crawl data from the web, existing approaches are either domain specific or limit the data extracted and linked to only entity based information as seen in the myDIG project under the DARPA Memex Program. The myDIG project and has been used in a few extremely useful use cases including a human trafficking application. However, the focus on entity specific information like an individual's name, address, educational qualification etc. limits the use cases of a web data extraction tool.

As a solution we propose a knowledge discovery approach to web data extraction, a novel tool that gathers the users insights for a specific domain to further understand the domain using public information on the web. For example, an organisation attempting to carry out a SWOT analysis on a company can provide all the information they have about the company and create a SWOT graph (as seen in Fig 2). These attributes (strength, weaknesses, opportunities & threats) are abstract data points and would require structuring vast arrays of data available on the internet. The data obtained in each website is further used to crawl and extract more relevant data. Once the task is complete, the graph is expanded with useful information based on the data collected and structured.

2 Related Work

2.1 First Generation Web Crawlers

Web Crawlers have led to remarkable applications that have arguably revolutionised the way we access documents on the internet. RBSE Spider, WebCrawler and The Wayback Machine are examples of early web crawlers built to index a small number of webpages.

The basic idea of these crawlers was to start with a set of well connected seed urls, visit the pages and add all the urls in the page to a queue. The queue urls are then visited iteratively to obtain more urls and eventually crawl large spaces of the web.

2.2 Second Generation Web Crawlers

The second generation web crawlers aimed to handle the exponential scaling of webpages on the internet e.g. Page and Brin (1998) or to build domain specific crawlers to crawl specific domains on the web (focused web crawlers) e.g. Sphinx and Mercator

2.3 Data Mining using Web Crawlers

A Mining based web crawler is an information retrieval system that aims to gather information from webpages rather than understanding and indexing web structure. Existing crawlers used for data mining either focus on a specific domain or do not carry out any text processing from the data.

2.4 Domain Specific Web Crawlers

The myDIG project under the DARPA Memex Program is the first and currently only tool that carries out generic domain specific data extraction on the web. myDIG focuses on extracting entity based information which is extremely useful for gathering simple entity based information from a vast array of documents. The tool requires labelling of similar data points that need to be extracted for each entity.

3. Project Objectives

We aim to achieve 5 main objectives through our project which are as follows:

1. Building a general purpose web crawler.

2. Topical Classifier using various NLP techniques to gather domain-specific webpages.

3. Building a pseudo-relevance feedback model for improving domain knowledge discovery.

4. Structuring the data obtained from the web crawler into relevant sub-categories which will be used later to build the knowledge graph.

5. Building an easy to read and interactive knowledge discovery graph for the user on the front-end.

4. Methodology

4.1 Architecture Design

4.2 Fetching Seed URLs

The web crawling process starts with selecting a few base URLs that constitute the seed set. These seed URLs act as starting pages which are parsed to extract links that are fed into a URL frontier queue which will be subsequently explored by the web crawler. This area of selection of seed urls has been given less importance in web crawling literature [1] and therefore, needs to be addressed.

In order to initialise the seed set, we plan to use search engine APIs like the Google JSON API, Bing Web Search API etc. The API will take a user query as its input and return a response with a set of search results. The first challenge lies in figuring out the set of terms which can be used to form queries relevant to a particular topic. For solving this, we plan to use a *Pseudo-Relevance Feedback* model [2]. This is an iterative process where new terms are selected for a successive query after performing some analysis on the documents received by the previous query.

The second challenge is the dependency of the URL selection process on the APIs' page ranking algorithm as it gets to decide which pages are more topic-related from its collection of indexed pages of the public web. To overcome this, Cao et al. [3] showed that using one's own *topical classifier* to filter documents and selecting the seeds that the classifier deems apt for the topic instead of simply choosing the top-K documents returned by the API improves the retrieval effectiveness of pseudo-relevance feedback methods.

Hence, we use an iterative classification-based pseudo relevance feedback approach to resolve this area of finding good and on-topic seeds.

The query issuing process will be composed of 2 steps: '*exploitation*' and '*exploration*' [1]. Exploitation is for deciding the best action to take given the information one has whereas exploration is to ensure diversity on the set of seed urls which is done by exploring actions that might seem sub-optimal at the moment, but can improve the results in the future.

4.3 Topical Classifier

Whenever one encounters hyperlinks on a webpage, it is either to help a user navigate or to direct them to another page in order to give them a deeper insight into the topic on the current page. If it is used for the latter reason, the hyperlinks can be easily used for topic-based focused crawling. Davidson [4] gives empirical proof to show that the linked pages have high textual similarity. In order to decide whether the hyperlinked URL goes into the frontier queue, the web crawler needs to predict whether the URL would link to a relevant page or not. THe first step would be to

4.4 Reinforcement Learning

Reinforcement Learning is a model that trains an agent via interaction with an environment. The agent receives rewards based on its interaction with the environment. The decision making process of the agent is modelled as a *Markov Decision Process* (MDD) where the agent has a

set of states S, a set of actions S, a reward function R and a transition function T which specifies the probability to go from a state s to another s'.
The value Q(s,a) of taking an action 'a' at state 's' with a learning rate of y can be modelled in the following way:

$$Q(s,a) = E\ T(s,a,s^*)[R(s,a,s^*) + yV(s^*)]$$

Where T(s,a,s') is the probability of going from state 's' to state 's*' by taking action 'a', R(s,a,s*) is the reward and V(s*) is the current value at state s*.

In the context of web crawling, R(s,a,s*) will be the reward obtained while going from a crawl state s to s*. If the page that leads to state s* is related to the domain that is being crawled, then the reward will be high.

4.4.1 Tunnelling

The goal of the model is to obtain as many relevant pages as possible in the least amount of time and by using the least resources. However, it is sometimes necessary to explore pages that are off topic that might lead to on topic pages. For example, if we want to extract information about Hong Kong, a tourism website might seem like an off topic page but might contain links to web pages related to Hong Kong.

According to Davison, there is substantial evidence that pages related to a particular topic link to other pages in the domain. Thus, pages that are on topic have a higher probability of containing relevant pages even if they my seem off topic. Menczer shows that relevance probability is within a distance of 3 links. Thus we can use parameters like distance from last relevant parent page, how different the new topic is from the domain, how relevant the parent topic is to the domain etc. to figure out whether to proceed on an off topic page.

5. Project Schedule and Milestones

Deadline	Task	Status
October 10	Research and Literature Review	Pending
November 1	Seed URL Fetching	Pending
November 7	General Web Crawler	Pending
December 30	Topical Classifier	Pending
January 10	Knowledge Discovery Graph	Pending
February 29	Reinforcement Learning	Pending
April 1	Testing and Final Report Draft	Pending

Davison [8] shows empirical evidence of topical locality on the Web.
Davison, B.D.: Topical locality in the web. In: SIGIR (2000)
Anchor text and surrounding text of the links are exploited to evaluate links.
Davison [8] shows that titles, descriptions, and anchor text represent the target page and that anchor text is most similar to the page to which it points.
RL in Web Crawling

Queue Priority Policy

Link Analysis Measure techniques - HITS and PageRank

Modeling the crawling environment as a markov decision process

1. Doc
2. Croft, W.B., Metzler, D., Strohman, T.: Search Engines - Information Retrieval in Practice. Pearson
 Education (2009)
3. Cao, G., Nie, J.Y., Gao, J., Robertson, S.: Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research
4. Davison, B.D.: Topical locality in the web. In: SIGIR (2000)

There have been previous research that uses web crawlers to extract domain specific information. Pirkola built a focused web crawler to acquire biological data from the web. Nemeslaki & Pocsarovszky used a web crawler to extract web population data from a social media website & to get time-series data from online business websites. DARPA Memex created a tool that builds a knowledge graph after crawling the web for a specific domain. The project has focused itself on extracting fact based entity data like a person's personal details.

Our Work

In our project, we aim to expand current search capabilities beyond the current commercial search engine solutions by allowing users to crawl the web for extensive information on a particular domain. The user provides a graph to explain the domain fron which data needs to be collected, this data is then used to crawl the web to gather relevant data from the domain. For example, a user trying to carry out an extensive SWOT analysis for a company can create a graph with strength, weaknesses, opportunities & threats connected to the company name. The tool will the use this data to get relevant data related to the company and branch out the graph based on other information found in the crawl. The user can ask for fact based information like 'When was this company founded?, text based information like 'Strengths' of the company or dark data from websites where data is hard to extract e.g. social media websites.

The effectiveness of focused crawling is often evaluated using the measures of harvest rate and coverage. Harvest rate refers to the proportion of documents relevant to the domain to all downloaded documents. Coverage refers to the number of obtained relevant pages at time point T. The domain relevance of the documents is judged by human assessors.

The quality of the downloaded data and the effectiveness of focused crawling vary considerably depending on many factors. One important factor is the method how probably relevant links are identified, e.g. on the basis of the content of documents or on the basis of link anchor texts [2]. The second major factor is how irrelevant documents should be handled [1, 3]. It is quite common that a relevant document points to an irrelevant document which points to another relevant document.

What is available today? I.e. Google

Works well for quick results to get a website or quick facts on a topic but there are alot of use cases that it doesn't fulfill. For example Research for companies, extensive academic researc etc.
The web is not a centrally managed repository of information but a dispersed
A web crawler is used to for downloading or indexing web pages in bulk. Give detailed definition of web crawler and how it works

Challenges:

1. Scale: A lot of information, everything might not be useful
2. Getting well dispersed seed urls
3. Tunnelling
4. Stopping mechanism: when do you have enough information and time problems

5. Mirror Links

Literature Review

Why have you created it?

Project Methodology

Basic Web Crawler

Reinforcement learning web crawler

Graphs