Towards a Relevant Search of Text, Image and Video Based on Question and Answering

Santosh G1 Prakash S2
1Department of Computer Networking Engineering 2Department of ISE
1PG Student 2Assistant Professor
RNS Institute of Technology, Bangalore

Abstract—Web crawler (also known as a Web spider or Web robot) is a program or automated script which browse
the World Wide Web in a methodical and automated manner. Web crawler is the heart of search engine. Whenever
user query comes, searching is performed within the database of indexed web page. So it is desired that only
the most relevant pages are stored in the database so as to
increase the efficiency of search engine. Our approach can
enable a multimedia question answering (MMQA) approach
as users can find the answers by matching their questions
with the pool.

Keywords: Web crawler, MMQA, Question Answering.

I. INTRODUCTION

Question-answering (QA) is a technique for automatically
answering a question posed in natural language. Compared
to keyword-based search systems, it greatly facilitates the
communication between humans and computer by naturally
stating users’ intention in plain sentences. It also avoids the
painstaking browsing of a vast quantity of information
contents returned by search engines for the correct answers.
However, fully automated QA still faces challenges that are
not easy to tackle, such as the deep understanding of complex
questions and the sophisticated syntactic, semantic
and contextual processing to generate answers. It is found
that, in most cases, automated approach cannot obtain
results that are as good as those generated by human
intelligence.

A Web crawler is one type of bot, or software
agent. In general, it starts with a list of URLs to visit, called
the seeds. As the crawler visits these URLs, it identifies all
the hyperlinks in the page and adds them to the list of URLs
to visit, called the crawl frontier. URLs from the frontier are
recursively visited according to a set of policies.

The behaviour of a Web crawler is the outcome of a combination of
policies:

- A selection policy that states which pages to
download
- A re-visit policy that states when to check for
changes to the pages
- A politeness policy that states how to avoid
overloading Web sites, and
- A parallelization policy that states how to
coordinate

Distributed web crawlers. Many legitimate sites, in
particular search engines, use spidering as a means of
providing up-to-date data. Web crawlers are mainly used to
create a copy of all the visited pages for later processing by
a search engine, which will index the downloaded pages to
provide fast searches. Crawlers can also be used for
automating maintenance tasks on a Web site, such as
checking links or validating HTML codes. Also, crawlers
can be used to gather specific types of information from
Web pages, such as harvesting e-mail addresses (usually for
spam).

II. RELATED WORK

Faced with the vast quantity of information returned by Web
search engines such as Google, Bing, and Yahoo, users can
easily become over-whelmed. Question-answering (QA)
research attempts to tackle this information-overload
problem. Instead of returning a ranked list of documents, as
with current search engines, QA leverages advanced media
content, linguistic analysis, and domain knowledge to return
precise answers to users natural-language questions.

However QA research has largely focused on text.
Given that the vast amount of information on the Web is
now in multimedia form, it is natural to extend text-based
QA research to multimedia QA (MMQA). (Author identifies
all types of answers except pure text as multimedia answers,
including images, video, images and text, and so forth.)

Further MMQA research must bear in mind several key
points. (i) It must manage incomplete metadata and clean up
noisy annotations; (ii) Appropriate multimedia answers are
more intuitive for some questions. (iii) Multimedia answers
are readily available for some types of questions given the
popularity of video- and image sharing sites. Thus, MMQA
can complement text QA in a complete QA paradigm in
which the best answers might be a combination of text and
other mediums.

Current technology is still far from enabling us to
benefit from MMQA. Furthermore, none of these works
fully exploit the rich content on Web 2.0. As it knows, Web
2.0 facilitates interactive information sharing, interoperability, and collaboration on the Internet.

Therefore, an emerging question is how to leverage user-
contributed data such as tagging, comments, and ratings for
MMQA. Such information is rapidly becoming more
abundant with the popularity of social media sites. For
example, YouTube serves 100 million distinct videos and
65,000 uploads daily, and the traffic of this site accounts for
more than 20 percent of all Web traffic and 10 percent of the
whole Internet, comprising 60 percent of the video watched
online. The photo-sharing site Flicker contained more than 6
billion images as of august 2011, and more than 3.5 billion
photos are being uploaded every month on Face book.

III. PROPOSED SYSTEM

At the beginning of each work cycle, a user obtains a URL
from the Frontier data structure, which dispenses URLs
according to their priority and to politeness policies. The
worker thread then invokes the HTTP fetcher. The fetcher
first calls a DNS sub-module to resolve the host component
of the URL into the IP address of the corresponding web
server (using cached results of prior resolutions if possible),
and then connects to the web server, checks for any robots
exclusion rules (which typically are cached as well), and attempts to download the web page. If the download succeeds, the web page may or may not be stored in a repository of harvested web pages (not shown).

Fig. 1: Frame Work of Web crawler

In either case, the page is passed to the Link extractor, which parses the page’s HTML content and extracts hyperlinks contained therein. The corresponding URLs are then passed to a URL distributor, which assigns each URL to a crawling process. This assignment is typically made by hashing the URLs host component, its domain, or its IP address (the latter requires additional DNS resolutions).

Since most hyperlinks refer to pages on the same web site, assignment to the local crawling process is the common case. Next, the URL passes through the Custom URL filter (e.g., to exclude URLs belonging to “black-listed” sites, or URLs with particular file extensions that are not of interest) and into the Duplicate URL eliminator, which maintains the set of all URLs discovered so far and passes on only never-before-seen URLs. Finally, the URL prioritize selects a position for the URL in the Frontier, based on factors such as estimated page importance or rate of change.

Fig. 2: Web Crawler Model

A Key motivation for designing Web crawlers has been to retrieve web pages and add them or their representation to a local repository. In its simplest form a crawler starts from a seed page and then uses the external links within it to attend to other pages. The structure of a basic crawler is shown in figure 2

The basic working of a web crawler can be discussed as follows:

- Maintains a list of unvisited URLs called the frontier, list is initialized with seed URLs which may be provided by a user or another program.
- Each crawling loop involves picking the next URL to crawl from the frontier, fetching the page corresponding to the URL through HTTP.
- Before the URLs are added to the frontier they may be assigned a score that represents the estimated benefit of visiting the page corresponding to the URL.
- The crawling process may be terminated when a certain number of pages have been crawled, if the crawler is ready to crawl another page.

IV. IMPLEMENTATION

User enters the question and selects the answer category and process to web server. These includes title, keywords and description. They will take precedence over the corresponding fields in a HTML-file if they are nonempty.

To get the lucenedata, a special header is sent to the web server when crawling: “request-lucenedata” with empty content. The server also checks that the request is authorized to receive this lucenedata. Some documents may be hidden by the External Visibility Setting. They will take precedence over the corresponding fields in a HTML-file if they are nonempty. This is done in order to be able to index all documents regardless of any access restrictions. Those are instead applied when a user performs a query. Some Site Builder lucenedata is also indexed. These include title, keywords and description.

V. CONCLUSION

It has proposed a web application in which user will get answer in different media format. For a given QA pair, our scheme first predicts which type of medium is appropriate for enriching the original textual answer. It has been developed user interface which is through which user can ask question and see the answer in different formats.

As a future work we will further improve the scheme, such as developing better query generation method and investigating the relevant segments from a video.

ACKNOWLEDGEMENT

I take this Opportunity to express my profound gratitude and deep regards to my guide Prof. Mr. Prakasha S Assistant Professor, RNS Institute of technology, Bangalore, for his exemplary guidance, and constant encouragement throughout.

REFERENCES


