1 Introduction

Question Answering which seeks to answer logical factoid questions such as "Who shot JFK?". Often this problem is approached by using information retrieval and named entity recognition technology[2]. These systems often used large ontologies that related questions to semantic classes of answers.

The main problem with this approach is the many to many complexity of the question and answer types. When asking a question involving a "what", it can refer to multiple different types of answers such as a solid object (e.g. What is on fire?) or an abstract idea (e.g. What are you thinking?). The same is said for mapping different question types mapping to the same answer. "What is your roommate’s name?" and "Who do you live with?" have the same answer.

Currently there is an emphasis on mining the data from the web. While the internet has seemingly endless amounts of information, not all of this information is useful and much of it is redundant. There are many information retrieval techniques that are able to manage the large amounts of data, but for specific questions, often those systems are too slow and impractical for usage. Users need to peruse multiple documents that are "possibly similar" instead of receiving an answer.

The paper, "Question Answering from the Web Using Knowledge Annotation and Knowledge Mining Techniques", discusses an approach to Question answering using Knowledge Annotation and Knowledge Mining to collaborate on returning specific and helpful answers.

2 Knowledge Annotation

Knowledge Annotation is an approach to answering large classes of frequently occurring questions by utilizing semistructured and structured Web sources.

By creating schemata of question types that are frequently asked, we can retrieve exact or close to exact data from these semistructured websites. These schema have two parts, a question signature and a database query. The question signature maps to a certain type of question such as "Who was born when?" The database query is made after using the question signature to determine the keywords. Because each data source is structured differently a wrapper must be created for each individual source.

3 Knowledge Mining

Knowledge Mining is a statistical approach that leverages massive amounts of Web data to overcome many natural language processing challenges.
One of the main challenges in QA is relating the formulation of questions to different formulations of answers[1]. In this case data redundancy is helpful for overcoming any sophisticated natural language techniques and poor document quality.

3.1 Data Flow of Webmining

Formulate Requests
Takes the question and formulates it into a query. Focuses on quantity to make up for lack of quality. Exact and inexact queries are created.

Execute Requests
Uses Google or any search engine to retrieve text snippets. Exact queries use extra pattern matching on the text snippets.

Generate N-grams
These N-grams are created and serve as the possible answers(candidates).

Vote
The new score of each answer candidate is equal to the sum of the z-scores of all occurrences of the particular n-gram.

Filter Candidates
Uses a course-grained filter and heuristics to reduce the number of answer candidates.

Combine Candidates
Use the short answers to boost scores of longer answers. For example, if ”Angeles” and ”Los Angeles” are both candidates, then the score from the shorter sequence is added to the longer.

Score Candidates
Determines the scores of the possible answers using the following equation. A is the set of keywords in the candidate answer; N is the total number of words in our corpus; Wc is the number of occurrences of word w in the corpus.

\[
\frac{1}{|A|} \sum_{w \in A} \log \left( \frac{N}{W_c} \right)
\]

Get Support
Verifies that the possible answers actually appear in the original text snippets.

References