Effectiveness of Web Page Classification on Finding List Answers

Hui Yang
School of Computing,
National University of Singapore
3 Science Drive 2, Singapore 117543
yangh@comp.nus.edu.sg

Tat-Seng Chua
School of Computing,
National University of Singapore
3 Science Drive 2, Singapore 117543
chuats@comp.nus.edu.sg

Abstract
List question answering (QA) offers a unique challenge in effectively and efficiently locating a complete set of distinct answers from huge corpora or the Web. In TREC-12, the median average F_1 performance of list QA systems was only 6.9%. This paper exploits the wealth of freely available text and link structures on the Web to seek complete answers to list questions. We employ natural language parsing, web page classification and clustering to find reliable list answers. We also study the effectiveness of web page classification on both the recall and uniqueness of answers for web-based list QA.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms
Algorithms, Performance

Keywords
Question Answering, Web page classification

1. Introduction
In the most recent QA main track of the Text REtrieval Conference (TREC), QA systems was developed to retrieve short precise answers or nuggets for factoid, definition, and list questions. The list task requires systems to assemble a set of distinct and complete answers as responses to questions like "*Name all the past and present NFL players.*" "*What are the brand names of Belgian chocolates?*". Comparing to definition and factoid tasks, list questions offer unique challenges in question interpretation and answer search. The TREC-12 QA results [3] reveal the general problem of low recall and non-distinctive answers for answering list questions in many state-of-the-art QA systems. Inspired by the great improvement in answering factoid questions [1] through the use of external resources such as the Web, we extend our original list QA system by exploiting more detailed web knowledge. This paper investigates the effectiveness of applying web page classification to find list answers.

2. System Design
Given millions of web pages returned by search engines, it is non-trivial to identify answers for a list question. Our strategy is to divide-and-conquer. Similar to TREC corpus, where a single document could contain multiple answer instances and the same answer instance might be repeated in multiple documents, web pages could also contain a list of answer instances. For example, for question "*What breeds of dog have won the 'Best in Show' award at the Westminster Dog Show?*", we can find Collection Pages, Topic Pages and Relevant Pages such as those listed in

Sigir'04, July 25–29, 2004, Sheffield, South York Shire, UK.
ACM 1-58113-881-4/04/0007.

Figure 1.

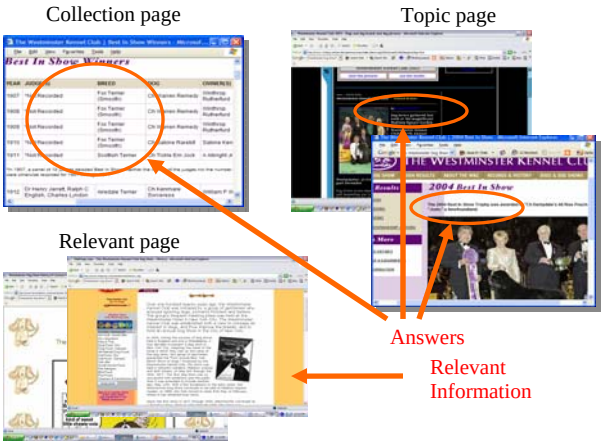

Figure 1: Web page classes: *Collection page* contains a list of items or hyperlinks; *Topic page* represents an answer instance best; *Relevant page* provides supporting information to an answer instance; *Irrelevant page* is not related to any answer.

In order to extract answers from those good answer resources, our system first performs natural language parsing to identify part-of-speech, Named Entities (NE), subject/object information to find the Answer Target Type and various key words. It then formulates a number of web queries based on heuristic patterns and submits these queries to popular web search engines (Google, Altavista and Yahoo) to get the top returned web pages. The retrieved pages are then classified into Collection, Topic, Relevant and Irrelevant sets. It then performs a redistribution of classified pages. It first obtains more Topic pages from the outgoing links from the Collection pages. It then dispatches Relevant pages to different Topic page cluster as supportive materials. Those clusters and collection pages are used as main sources to extract answers. (Figure 2)

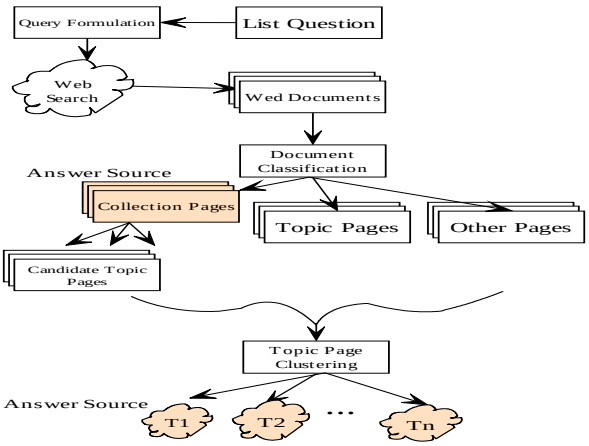

Figure 2: System Architecture

3. Finding Reliable Answer Sources by Web Page Classification

To classify web pages returned by search engines, it is crucial to find a good set of features to represent the web documents. In our approach, we rely on two types of features. First, we obtain the query words based on subject/object detection and named entity recognition of the original questions. In general, we observe that there is a large number of named entities of same type appearing in a Collection Page, typically within a list or table. In a Topic Page, there is also typically a group of named entities, which could correspond to our original query terms or Answer Target type. Second, we found that Topic page is highly likely to repeat the subject terms in its URL, title, or at the beginning of its page. In general, if the subject appears in important locations, such as in HTML tags <title>, <H1> and <H2>, or appears frequently, then the corresponding pages should be Topic pages and their topic is about the answer target.

Based on the above discussion, we designed a set of 29 features based on *Known Named Entity Type (Types of NE appearing in question), Answer Target Type, ordinary Named Entities, list, table, URL, HTML structure, Anchor, Hyperlinks, and document length* to represent the web pages. We also have features like: (a) |*Known_NE*|, which is total number of NEs in the web page with the same NE type as those in the question. In the "*dog breed*" example, it is the number of Location NEs since "*Westminster*" is identified as Location by NER; (b) |*Answer_NE*|, which is number of NEs belonging to expected answer type. In the "*dog breed*" example, it is the number of *Breed* NEs); and (c) |* <a href=*|, which is number of HTML tags to represent a list/table of anchors; number of *in-links and out-links*; etc.

We train two C4.5 Decision Tree classifiers [2] to perform the classification. The first Classifier classifies the web pages into Collection pages and non-collection pages while the second Classifier further classifies the non-collection pages into Topic pages and Others. We use 50 list questions from TREC-10 and TREC-11 for training and all TREC-12 list questions for testing. Some of the decision rules found are as follows:

- OUT_Link >= 25&NE > 78&Answer_NE>= 30->Class CP
- OUT_Link <= 25&Answer_NE<=5& NE>46 ->Class TP
- OUT_Link >= 25 & URL_Depth > 3 -> Others
- NE <= 4 -> Others

After forming the initial sets of Collection page `CPSet`, Topic page `TPSet` and `OtherSet`, we then use the *outgoing links* of Collection pages to find more Topic pages. These outgoing pages are potential Topic pages but not necessarily appearing among the top returned web documents. The new Topic page set becomes `TPSet' = TPSet + {outgoing pages of CPs}`.

Next, we select distinct Topic pages from `TPSet'`. We compare the page similarity between each pair of Topic pages. For those pairs with high similarity above a certain threshold, we keep the page that contains more named entities of answer type in `TPSet'` and move the other into `OtherSet`. The resulting Topic pages in `TPSet'` are distinct and will be used as cluster seeds.

Finally, we identify and dispatch Relevant pages from `OtherSet` into appropriate clusters based on their similarities with the cluster seeds. Each cluster corresponds to a distinct answer instance. Topic page provides the main facts about that answer instance while Relevant pages provide the supporting facts. The rest of web pages are thrown into `IrrelevantSet`.

Through topic page clustering, we eliminate most answer redundancy, and offering a higher chance of finding distinct answers on the noisy Web.

4. Answer Extraction

Collection pages are very good answer resource for list QA. However, to extract the "exact" answers from the resource page is not a trivial task. We need to perform wrapper induction to extract the useful contents. Having the Topic pages clustered for a certain question and analyzing the main Topic pages in each cluster, we can easily extract the possible answers based on the answer target type. The answers obtained from the web pages are then "projected" [1] onto TREC AQUAINT corpus to get the TREC answers. In case when no TREC answer can be found based on the main Topic page of a cluster, we go to the next most relevant page in the same cluster to seek the answer. The process is repeated until either an answer is found in TREC corpus or when all Relevant pages in the cluster have been exhausted. For question *"Which countries did First Lady Hillary Clinton visit?"*, we found 38 answers. The recall is much higher than the best performing system in TREC-12 [3] which only found 26 out of 44 answers.

5. Evaluation

Table 1 shows that we can achieve an overall classification average precision of 0.897 and average recall of 0.851. This performance is adequate to support the subsequent steps of finding complete answers.

Table 1: Performance of Web Page Classification

Web Page Class	Avg Prec.	Avg Rec.
Collection	91.1%	89.5%
Topic	92.0%	88.4%
Relevant	86.5%	83.4%
Overall	89.7%	85.1%

The use of web knowledge and web page classification makes great impact to our list QA system. We tested our system on 37 TREC-12 list questions. The results are encouraging and show that we can improve our TREC-12 system by 47% in F_1 and 59% in recall. The system outperforms the F_1 score of the best TREC-12 QA system by 19.6%. (Table 2)

Table 2: Performance on TREC-12 List Questions

	Recall	*F_1*
Our TREC-12 system w/o using web	0.264	0.317
Best TREC-12 system	unknown	0.392
Presented approach w/ web page classification	0.422	0.469

6. Conclusion

We proposed an innovative way to explore the effect of web information and web page classification on finding answers to list questions. Our system focused on answer completeness and uniqueness. Using the proposed approach, we could achieve a recall of 0.422 and F_1 of 0.469, which is significantly better than the top performing systems in TREC-12 List QA task. The results obtained from the TREC-12 data set demonstrated that our approach is feasible and effective for list QA.

7. References

[1] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng. 2001. "Data-intensive question answering". In Proceedings of the Tenth Text REtrieval Conference (TREC 2001)

[2] J. R. Quinlan, 1993. C4.5: Programs for Machine Learning. Morgan-Kaufmann, San Francisco.

[3] E.M.Voorhees. 2003. “Overview of the TREC 2003 Question Answering Track.” In notebook of the Twelfth Text REtrieval Conference (TREC’2003), 14-27.