FADA: *Find All Distinct* Answers

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

Abstract
The wealth of information available on the web makes it an attractive resource for seeking quick answers. While web-based question answering becomes an emerging topic in recent years, the problem of efficiently locating a complete set of distinct answers on the Web is far from being solved. We introduce our system, FADA, which relies on question event analysis, web page clustering, and natural language parsing, to find reliable distinct answers with high recall. The method has been found to be effective in strengthening state-of-the-art Web question answering techniques by emphasizing on answer completeness and uniqueness.

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

General Terms
Algorithms, Performance, Experimentation

Keywords
Question Answering, Web page classification

1. Introduction
While Web question answering systems [2][4] are still in a preliminary stage of development, there is already a large body of research on general Question Answering (QA). Most recent QA research is represented in the QA track of the Text REtrieval Conference (TREC), which involves retrieving short precise answers for factoid, definition, and list questions. Both definition and list tasks require systems to assemble a set of distinct and complete answers as response to questions like "*Who is Donna Elvira?*" "*What are the brand names of Belgian chocolates?*". An analysis of the results of the participated systems in the recent TREC-12 [5] reveals that many systems still suffer from the general problem of low recall and non-distinctive answers for answering definition and list questions. We expect this problem to be amplified when we extend the TREC QA techniques to perform Web QA due to the large amount and great variety of Web documents. This paper investigates the deployment of Event-based QA analysis to tackle this problem and demonstrates that the resulting system called **FADA** (**F**ind **A**ll **D**istinct **A**nswers) could achieve effective question answering on the Web.

2. Distinct Answers in QA Event Space
Everything in the world is related to a certain event. At every moment in the world, there are a lot of events happening simultaneously and each involves a number of related entities. *An Event provides the topic of a question and the related context*. Thus in QA, we consider every question to be related to a certain event. Questions can be asked about the entire events or facets of the events. The question itself typically contains some known facets, which can be used to look for other unknown facets. Hence, *Question* provides information about the topic, context, and known facets while *Answer* lies in unknown facets, which could be discovered after searching a document collection.

From a geometric perspective, a set of *x* attributes defines an *x*-dimensional *QA Event space* in which each event is a point. Definition question could be considered as a group of events sharing *one* attribute/element, which is the question topic. For example, the answers to "*Who is Vlad the Impaler?*" will be a group of events happened on this 16th century warrior prince to show that he "*inspired novel Dracula in 1897*", "*fought Turks in Translavania*" and "*was buried in medieval monastery*". (See Figure 1a)

List question could also be considered as a group of events that share *several* attributes/elements. For example, the answers to "*Which countries were visited by first lady Hillary Clinton?*" will be a group of places aligned on Location axis. We can know from the Event space that Hillary Clinton visited *"Egypt in 1999" and "China in 1998"*. (Figure 1b). The dimension of the solution space depends on the number of unknown elements.

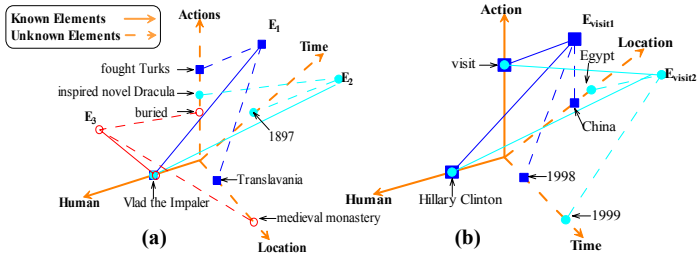

Figure 1: Event Models for "Vlad" and "Hillary visit"

In QA Event space, every event is unique. The problem of finding all distinct answers thus becomes finding all related events for a question. In order to find a complete set of distinct events, we need to perform web document clustering to get all distinct web pages that could represent a QA Event.

3. Find All Distinct QA Event on the Web
FADA performs question parsing to identify the known event elements, and expected answer type. It extracts several sets of words from the original question (known elements) and identifies the detailed question classes (answer element type). It then formulates a number of queries by combining the known elements together with heuristic patterns for list questions. For example, for question "*Name all the past and present NFL players.*", we formed 20+ queries like *"NFL player list", "favorite NFL players", "directory NFL player", etc.* FADA submits these queries to commercial search engines to get the top N Web pages. The retrieved pages are then classified into four classes: *Collection page, Topic page, Relevant page, Irrelevant page* (Table 1) based on their functionality and contribution in finding list answers.

WWW 2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

Table 1: List of Web Page Classes

Web page class	Description
Collection page	Containing a list of items or hyperlinks
Topic page	The best page to represent an event
Relevant page	Relevant to an event by providing either supporting or objection information to the Topic page
Irrelevant page	Not related to any event

3.1 Web Page Classification

In order to classify web pages returned by search engines into 4 categories, we designed a set of 27 features based on *Known elements, Unknown elements, Answer Target elements, Hyperlinks, URL, HTML structure, Anchor, list and Named Entities* to represent web pages. Table 2 gives some of the essential features used in our system. We trained two classifiers: Collection Page Classifier and Topic Page Classifier. Both Classifiers are implemented using Decision Tree C4.5 [3]. We randomly selected 100 *Collection Pages (CP)*, 50 *Topic Pages (TP)* and 50 *Relevant Pages (RP)* to train and test the Collection Page Classifier; and 100 Topic Pages and 100 Relevant Pages to train and test the Topic Page Classifier. Our experiments showed that we could achieve a classification precision of 91.1% and 92% for *CP and TP* respectively.

Table 2: Partial List of Web Page Features

No	Feature	Explanation
10	|Known_NE| / |NE|	Ratio of NEs that belong to Known element type to total # of NEs
12	|Answer_NE| / |NE|	Ratio of NEs that belong to Answer Target type to total # of NEs
14	Content_Length	# of words in a page excluding HTML tags
17	|In_Link|	# of in-links
20	Keyword_in_Title	Boolean indicating presence of keywords in page title
26	| <a href=|	# of HTML tags, including , , , , to represent a list/table of anchors,
27	URL_Depth	The depth of URL

3.2 Web Page Clustering

Based on Web page classification, we formed the initial sets of *CP, TP and Other pages*. First, we used the *outgoing pages* of CPs to find more TPs in order to boost the recall. These outgoing pages are potential TPs but not necessarily appearing among the top N returned web documents. Second, we selected distinct TPs and use them as cluster seeds. Finally, we clustered Other pages into appropriate clusters based on their similarities with the cluster seeds. The page similarities are measured based on a linear combination of overlaps between *Known_NE, Answer_NE, URL similarity and link similarity*. Therefore, each cluster contains relevant information for a unique event. The procedure to form clusters of web pages is summarized as follows:

```
Unsupervised_Web_Page_Clustering (CPSet, TPSet, OtherSet) {
   set IrrelevantSet = null;
   for each cp_i in CPSet
         insert outgoing pages of cp_i into TPSet;
   for each pair {tp_i, tp_j} in TPSet
         if (sim(tp_i,tp_j)> θ)
               move max {ℵ_ANE in tp_i , ℵ_ANE in tp_j }into OtherSet;
   for each rp_i in OtherSet {
         k = argmax {sim(rp_i , tp_j) }
         if (sim(rp_i , tp_k ) > τ)
            insert rp_i into cluster_k;
         else   insert rp_i into IrrelevantSet;
   }
}
```

At the end of this process, Relevant pages are put into clusters, whose center is a Topic page. The average ratio of correct clustering is 54.1% in our experiments. Each cluster corresponds to a distinct event. Topic page provides the main facts about that event while Relevant pages provide supporting materials. Our subsequent tests reveal that the new Topic pages introduced by Collection pages greatly increase overall answer recall by 23%.

4. Answer Generation

Having the Web pages clustered for a certain question, especially when the clusters nicely match distinct events containing the answers, we could easily extract the possible answers based on the answer target type. For the "*Hillary visit*" example, we extracted the *Locations* after performing Named Entity analysis on each cluster and projected the answers onto TREC corpus. We found 38 answers. The recall is much higher than the best performing system [1] in TREC-12 which found 26 out of 44 answers. In the case of multiple answer candidates appearing in the same Topic page, we output the passages that have most variety of NE types since it is likely to be a comprehensive description about all the facets of an event.

5. Evaluation on TREC-12 Question Set

Due to the lack of benchmark for Web question answering, we use the TREC-12 Question set to test the overall performance of our system and compare the answers we found on the Web with the answers provided by NIST. TREC-12 has 37 list questions, in which each question expects an unlimited number of distinct answers of a certain type. NIST assessors constructed these questions. We select 28 questions on Person, Organization, Location, Time and Date as our test set. The results are encouraging and show that we could outperform the average F_1 score of the top 5 TREC-12 QA systems [5] by 120% and the best TREC-12 QA system [1] by 19.6%. (Table 3)

Table 3: Performance on 28 TREC-12 List Questions

	Avg prec.	*Avg recall*	*F_1*
Top 5 TREC12 systems Avg score	-	-	0.213
TREC-12 Best system	-	-	0.392
FADA	0.584	0.422	0.469

6. Conclusion

We have presented the techniques used in FADA system, which aims to find complete, distinct answers on the Web based on QA event analysis, web page clustering and natural language parsing. Using the novel 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.

7. References

[1] S. Harabagiu, D. Moldovan, C. Clark, M. Bowden, J. Williams, J. Bensley, "Answer Mining by Combining Extraction Techniques with Abductive Reasoning," In notebook of the 12th Text REtrieval Conference, 46-53.

[2] Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld, 2001, "Scaling Question Answering to the Web", In Proceedings of the 10th ACM World Wide Web conference.

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

[4] D. R. Radev, Hong Qi, Zhiping Zheng, Sasha Blair-Goldensohn, Zhu Zhang, Waiguo Fan, and John Prager. 2001. "Mining the web for answers to natural language questions". In the 10th International Conference on Information and Knowledge Management.

[5] E.M.Voorhees. 2003. "Overview of the TREC 2003 Question Answering Track." In notebook of the 12th Text REtrieval Conference, 14-27.