Question Answering using Integrated Information Retrieval and Information Extraction

Barry Schiffman and **Kathleen R. McKeown**
Department of Computer Science
Columbia University
New York, NY 10027
bschiff,kathy@cs.columbia.edu

Ralph Grishman
Department of Computer Science
New York University
New York, NY 10003
grishman@cs.nyu.edu

James Allan
University of Massachusetts
Department of Computer Science
Amherst, MA 01003
allan@cs.umass.edu

Abstract

This paper addresses the task of providing extended responses to questions regarding specialized topics. This task is an amalgam of information retrieval, topical summarization, and Information Extraction (IE). We present an approach which draws on methods from each of these areas, and compare the effectiveness of this approach with a query-focused summarization approach. The two systems are evaluated in the context of the *prosecution* queries like those in the DARPA GALE distillation evaluation.

1 Introduction

As question-answering systems advance from handling factoid questions to more complex requests, they must be able to determine how much information to include while making sure that the information selected is indeed relevant. Unlike factoid questions, there is no clear criterion that defines the kind of phrase that answers the question; instead, there may be many phrases that could make up an answer and it is often unclear in advance, how many. As system developers, our goal is to yield high recall without sacrificing precision.

In response to questions about particular events of interest that can be enumerated in advance, it is possible to perform a deeper semantic analysis focusing on the entities, relations, and sub-events of interest. On the other hand, the deeper analysis may be errorful and will also not always provide complete coverage of the information relevant to the query. The challenge, therefore, is to blend a shallower, robust approach with the deeper approach in an effective way.

In this paper, we show how this can be achieved through a synergistic combination of information retrieval and information extraction. We interleave information retrieval (IR) and response generation, using IR in high precision mode in the first stage to return a small number of documents that are highly likely to be relevant. Information extraction of entities and events within these documents is then used to pinpoint highly relevant sentences and associated words are selected to revise the query for a second pass of retrieval, improving recall. As part of this process, we approximate the relevant context by measuring the proximity of the target name in the query and extracted events.

Our approach has been evaluated in the framework of the DARPA GALE[1] program. One of the GALE evaluations involves responding to questions based on a set of question templates, ranging from broad questions like "Provide information on *X*", where X is an organization, to questions focused on particular classes of events. For the experiments presented here, we used the GALE program's *prosecution* class of questions. These are given in the following form: "Describe the prosecution of *X* for *Y*," where *X* is a person and *Y* is a crime or charge. Our results show that we are able to achieve higher accu-

[1] Global Autonomous Language Exploitation

racy with a system that exploits the justice events identified by IE than with an approach based on query-focused summarization alone.

In the following sections, we first describe the task and then review related work in question-answering. Section 3 details our procedure for finding answers as well as performing the information retrieval and information extraction tasks. Section 4 compares the results of the two approaches. Finally, we present our conclusion and plans for future work.

1.1 The Task

The language of the question immediately raises the question of what is meant by prosecution. Unlike a question such as "When was *X* born?", which is expected to be answered by a clear, concrete phrase, the prosecution question asks for a much greater range of material. The answer is in no way limited to the statements and activities of the prosecuting attorney, although these would certainly be part of a comprehensive answer.

In the GALE relevance guidelines[2], the answer can include many facets of the case:

- Descriptions of the accused's involvement in the crime.
- Descriptions of the activities, motivations, and involvement in the crime.
- Descriptions of the person as long as they are related to the trial.
- Information about the defense of the suspect.
- Information about the sentencing of the person.
- Information about similar cases involving the person.
- Information about the arrest of the person and statements made by him or her.
- Reactions of people involved in the trial, as well as statements by officials or reactions by the general public.

[2]BAE Systems Advanced Information Technologies, "Relevance Guidelines for Distillation Evaluation for GALE: Global Autonomous Language Exploitation", Version 2.2, January 25, 2007

The guidelines also provide a catchall instruction to "include reported information believed to be relevant to the case, but deemed inadmissible in a court of law."

It is easy to see that the use of a few search terms alone will be insufficient to locate a comprehensive answer.

We took a broad view of the question type and consider that any information about the investigation, accusation, pursuit, capture, trial and punishment of the individual, whether a person or organization, would be desireable in the answer.

1.2 Overview

The first step in our procedure sends a query tailored to this question type to the IR system to obtain a small number of high-quality documents with which we can determine what name variations are used in the corpus and estimate how many documents contain references to the individual. In the future we will expand the type of information we want to glean from this small set of documents. A secondary search is issued to find additional documents that refer to the individual, or individuals.

Once we have the complete document retrieval, the foundation for finding these types of events lies in the Proteus information extraction component (Grishman et al., 2005). We employ an IE system trained for the tasks of the 2005 Automatic Content Extraction evaluation, which include entity and event extraction. ACE defines a number of general event types, including *justice* events, which cover indictments, accusations, arrests, trials, and sentencings. The union of all these specific categories gives us many of the salient events in a criminal justice case from beginning to end. The program uses the events, as well as the entities, to help identify the passages that respond to the question.

The selection of sentences is based on the assumption that the co-occurrence of the target individual and a judicial event indicates that the target is indeed involved in the event, but these two do not necesssarily occur in the same sentence.

2 Related Work

A large body of work in question-answering has followed from the opening of the Text Retrieval Con-

ference's Q&A track in 1999. The task started as a group of factoid questions and expanded from there into more sophisticated problems. TREC provides a unique testbed of question-answer pairs for researchers and this data has been influential in furthering progress.

In TREC 2006, there was a new secondary task called "complex, interactive Question Answering," (Dang et al., 2006) which is quite close to the GALE problem, though it incorporated interaction to improve results. Questions are posed in a canonical form plus a narrative elaborating on the kind of information requested. An example question (from the TREC guidelines) asks, "What evidence is there for transport of [drugs] from [Bonaire] to the [United States]?" Our task is most similar to the fully-automatic baseline runs of the track, which typically took the form of passage retrieval with query expansion (Oard et al., 2006) or synonym processing (Katz et al., 2006), and not the deeper processing employed in this work.

Within the broader QA task, the *other* question type is closest to the requirements in GALE, but it is too open ended. In TREC, the input for *other* questions is the name or description of the target, and the response is supposed to be all information that did not fit in the answers to the previous questions. While a few GALE questions have similar input, most, including the prosecution questions, provide more detail about the topic in question.

A number of systems have used techniques inspired by information extraction. One of the top systems in the *other* questions category at the 2004 and 2005 evaluations generated lexical-syntactic patterns and semantic patterns (Schone et al., 2004). But they build these patterns from the question. In our task, we took advantage of the structured question format to make use of extensive work on the semantics of selected domains. In this way we hope to determine whether we can obtain better performance by adding more sophisticated knowledge about these domains. The Language Computer Corporation (LCC) has long experimented with incorporating information extraction techniques. Recently, in its system for the *other* type questions at TREC 2005, LCC developed search patterns for 33 target classes (Harabagiu et al., 2005). These patterns were learned with features from WordNet, stemming and named entity recognition.

More and more systems are exploiting the size and redundancy of the Web to help find answers. Some obtain answers from the Web and then project the answer back to the test corpus to find a supporting document (Voorhees and Dang, 2005). LCC used "web boosting features" to add to key words (Harabagiu et al., 2005). Rather than go to the Web and enhance the question terms, we made a beginning at examining the corpus for specific bits of information, in this prototype, to determine alternative realizations of names.

3 Implementation

As stated above, the system takes a query in the XML format required by the GALE program. The query templates allow users to amplify their requests by specifying a timeframe for the information and/or a locale. In addition, there are provisions for entering synonyms or alternate terms for either of the main arguments, i.e. the accused and the crime, and for related but less important terms.

Since this system is a prototype written especially for the GALE evaluation in July 2006, we paid close attention to the way example questions were given, as well as to the evaluation corpus, which consisted of more than 600,000 short news articles. The goal in GALE was to offer comprehensive results to the user, providing all snippets, or segments of texts, that responded to the information request. This required us to develop a strategy that balanced precision against recall. A system that reported only high-confidence answers was in danger of having no answers or far fewer answers than other systems, while a system that allowed lower confidence answers risked producing answers with a great deal of irrelevant material. Another way to look at this balancing act was that it was necessary for a system to know when to quit. For this reason, we sought to obtain a good estimate of the number of documents we wanted to scan for answers.

Answer selection focused first on the name of the suspect, which was always given in the query template. In many of the training cases, the suspect was in the news only because of a criminal charge against him; and in most, the charge specified was the only accusation reported in the news. Both location and

date constraints seemed to be largely superfluous, and so we ignored these. But we did have a mechanism for obtaining supplementary answers keyed to the brief description of the crime and other related words

The first step in the process is to request a seed collection of 10 documents from the IR system. This number was established experimentally. The IR query combines terms tailored to the prosecution template and the specific template parameters for a particular question. The 10 documents returned are then examined to produce a list of name variations that substantially match the name as rendered in the query template. The IR system is then asked for the number of times that the name appears in the corpus. This figure is adjusted by the frequency per document in the seed collection and a new query is submitted, set to obtain the *N* documents in which we expect to find the target's name.

3.1 Information Retrieval

The goal of the information retrieval component of the system was to locate relevant documents that the summarization system could then use to construct an answer. All search, whether high-precision or high-recall, was performed using the Indri retrieval system [3] (Strohman et al., 2005).

Indri provides a powerful query language that is used here to combine numerous aspects of the query. The Indri query regarding Saddam Hussein's prosecution for crimes against humanity includes the following components: source restrictions, prosecution-related words, mentions of Saddam Hussein, justice events, dependence model phrases (Metzler and Croft, 2005) regarding the crime, and a location constraint.

The first part of the query located references to prosecutions by looking for the keywords *prosecution*, *defense*, *trial*, *sentence*, *crime*, *guilty*, or *accuse*, all of which were determined on training data to occur in descriptions of prosecutions. These words were important to have in documents for them to be considered relevant, but the individual's name and the description of the crime were far more important (by a factor of almost 19 to 1).

The more heavily weighted part of the query, then, was a "justice event" marker found using information extraction (Section 3.2) and the more detailed description of that event based on phrases extracted from the crime (here *crimes against humanity*). Those phrases give more probability of relevance to documents that use more terms from the crime. It also included a location constraint (here, *Iraq*) that boosted documents referring to that location. And it captured user-provided equivalent words such as *Saddam Hussein* being a synonym for *former President of Iraq*.

The most complex part of the query handled references to the individual. The extraction system had annotated all person names throughout the corpus. We used the IR system to index all names across all documents and used Indri to retrieve any name forms that matched the individual. As a result, we were able to find references to *Saddam*, *Hussein*, and so on. This task could have also been accomplished with cross-document coreference technology but our approach appeared to compensate for incorrectly translated names slightly better than the coreference system we had available at the time. For example, *Present rust Hussein* was one odd form that was matched by our simple approach.

The final query looked like the following:

```
#filreq( #syn( #1(AFA).source ... #1(XIE).source )
   #weight(
     0.05 #combine( prosecution defense trial sentence
                    crime guilty accuse )
     0.95 #combine(
        #any:justice
        #weight(1.0 #combine(humanity against crimes)
               1.0 #combine(
                  #1(against humanity)
                  #1(crimes against)
                  #1(crimes against humanity))
                1.0 #combine
                   #uw8(against humanity)
                   #uw8(crimes humanity)
                   #uw8(crimes against)
                   #uw12(crimes against humanity)))
        Iraq

        #syn( #1(saddam hussein)
              #1(former president iraq))

        #syn( #equals( entity 126180 ) ...))))
```

The actual query is much longer because it contains 100 possible entities and numerous sources. The processing is described in more detail elsewhere (Kumaran and Allan, 2007).

3.2 Information Extraction

The Proteus system produces the full range of annotations as specified for the ACE 2005 evaluation, including entities, values, time expressions, relations,

[3] http://lemurproject.org/indri

and events. We focus here on the two annotations, entities and events, most relevant to our question-answering task. The general performance on entity and event detection in news articles is within a few percentage points of the top-ranking systems from the evaluation.

The extraction engine identifies seven semantic classes of entities mentioned in a document, of which the most frequent are persons, organizations, and GPE's (geo-political entities – roughly, regions with a government). Each entity will have one or more *mentions* in the document; these mentions include names, nouns and noun phrases, and pronouns. Text processing begins with an HMM-based named entity tagger, which identifies and classifies the names in the document. Nominal and pronominal mentions are identified either with a chunker or a full Penn-Treebank parser. A rule-based coreference component identifies coreference relations, forming entities from the mentions. Finally, a semantic classifier assigns a class to each entity based on the type of the first named mention (if the entity includes a named mention) or the head of the first nominal mention (using statistics gathered from the ACE training corpus).

The ACE annotation guidelines specify 33 different event subtypes, organized into 8 major types. One of the major types is justice events, which include arrest, charge, trial, appeal, acquit, convict, sentence, fine, execute, release, pardon, sue, and extradite subtypes. In parallel to entities, the event tagger first identifies individual event mentions and then uses event coreference to form events. For the ACE evaluation, an annotated corpus of approximately 300,000 words is used to train the event tagger.

For each event mention in the corpus, we collect the trigger word (the main word indicating the event) and a pattern recording the path from the trigger to each event argument. These paths are recorded in two forms: as the sequence of heads of maximal constituents between the trigger and the argument, and as the sequence of predicate-argument relations connecting the trigger to the argument[4]. In addition, a set of maximum-entropy classifiers are trained: to distinguish events from non-events, to classify events by type and subtype, to distinguish arguments from non-arguments, and to classify arguments by argument role. In tagging new data, we first match the context of each instance of a trigger word against the collected patterns, thus identifying some arguments. The argument classifier is then used to collect additional arguments within the sentence. Finally, the event classifier (which uses the proposed arguments as features) is used to reject unlikely events. The patterns provide somewhat more precise matching, while the argument classifiers improve recall, yielding a tagger with better performance than either strategy separately.

[4]These predicate argument relations are based on a representation called GLARF (Grammatical-Logical Argument Representation Framework), which incorporates deep syntactic relations and the argument roles from PropBank and NomBank.

3.3 Answer Generation

Once the final batch of documents is received, the answer generator module selects candidate passages. The names, with alternate renderings, are located through the entity mentions by the IE system. All sentences that contain a *justice* event and that fall within a mention of a target by no more than *n* sentences, where *n* is a settable parameter, which was put at 5 for this evaluation, form the core of the system's answer.

The tactic takes the place of topic segmentation, which we used for other question types in GALE that did not have the benefit of the sophisticated event recognition offered by the IE system. Segmentation is used to give users sufficient context in the answer without needing a means of identifying difficult definite nominal resolution cases that are not handled by extraction.

In order to increase recall, in keeping with the need for a comprehensive answer in the GALE evaluation, we added sentences that contain the name of the target in documents that have *justice* events and sentences that contain words describing the crime. However, we imposed a limitation on the growth of the answer size. When the target individual is well-known, he or she will be mentioned in numerous contexts, reducing the likelihood that this additional mention will be relevant. Thus, when the size of the answer grew too rapidly, we stopped including these additional sentences, and produced sentences only from the *justice* events. The threshold for triggering this shift was 200 sentences.

3.4 Summarization

As a state-of-the-art baseline, we used a generic multidocument summarization system that has been tested in numerous contexts. It is, indeed, the backup answer generator for several question types, including the prosecution questions, in our GALE system, and has been been tested in the topic-based tasks of the 2005 and 2006 Document Understanding Conferences.

A topic statement is formed by collapsing the template arguments into one list, e.g., "saddam hussein crimes against humanity prosecution", and the answer generation module proceeds by using a hybrid approach that combines top-down strategies based on syntactic patterns, alongside a suite of summarization methods which guide content in a bottom-up manner that clusters and combines the candidate sentences (Blair-Goldensohn and McKeown, 2006).

4 Evaluation

The results of our evaluation are shown in Table 1. We increased the number of test questions over the number used in the official GALE evaluation and we used only previously unseen questions. Documents for the baseline system were selected without use of the event annotations from Proteus.

We paired the 25 questions for judges, so that both the system's answer and the baseline answer were assigned to the same person. We provided explicit instructions on the handling on implicit references, allowing the judges to use the context of the question and other answer sentences to determine if a sentence was relevant – following the practice of the GALE evaluation.

Our judges were randomly assigned questions and asked whether the snippets, which in our case were individual sentences, were relevant or not; they could respond *Relevant*, *Not Relevant* or *Don't Know*. In cases where references were unclear, the judges were asked to choose *Don't Know* and these were removed from the scoring.[5]

[5]In the GALE evaluation, the snippets are broken down by hand into *nuggets* – discrete pieces of information – and the answers are scored on that basis. However, we scored our responses on the basis of snippets (sentences) only, as it is much more efficient, and therefore more feasible to repeat in the future.

Our system using IE event detection and entity tracking outperformed the summarization-based baseline, with average precision of 68% compared with 57%. Moreover, the specialized system sustained that level of precision although it returned a much larger number of snippets, totaling 2,086 over the 25 questions, compared with 363 for the baseline system. We computed a relative recall score, using the union of the sentences found by the systems and judged relevant as the ground truth. For recall, the specialized system scored an average 89% versus 17% for the baseline system. Computing an F-measure weighting precision and recall equally, the specialized system outperformed the baseline system 75% to 23%. The difference in relative recall and F-measure are both statisticaly significant under a two-tailed, paired t-test, with $p < 0.001$.

5 Conclusion and Future Work

Our results show that the specialized system statistically outperforms the baseline, a well-tested query focused summarization approach, on precision. The specialized system produced a much larger answer on average (Table 1). Moreover, our answer generator seemed to adapt well to information in the corpus. Of the six cases where it returned fewer than 10 sentences, the baseline found no additional sentences four times (Questions B006, B011, B015 and B022). We regard this as an important property in the question-answering task.

A major challenge is to ascertain whether the mention of the target is indeed involved in the recognized *justice* event. Our event recognition system was developed within the ACE program and only seeks to assigns roles within the local context of a single sentence. We currently use a threshold to consider whether an entity mention is reliable, but we will experiment with ways to measure the likelihood that a particular sentence is about the prosecution or some other issue. We are planning to obtain various pieces of information from additional secondary queries to the search engine. Within the GALE program, we are limited to the defined corpus, but in the general case, we could add more varied resources.

In addition, we are working to produce answers using text generation, to bring more sophisticated summarization techniques to make a better presen-

QID	System with IE				Baseline System			
	Precision	Recall	F-meas	Count	Precision	Recall	F-meas	Count
B001	0.728	0.905	0.807	92	0.818	0.122	0.212	11
B002	0.713	0.906	0.798	108	0.889	0.188	0.311	18
B003	0.770	0.942	0.848	148	0.875	0.058	0.109	8
B004	0.930	0.879	0.904	86	1.000	0.154	0.267	14
B005	0.706	0.923	0.800	34	0.400	0.231	0.293	15
B006	1.000	1.000	1.000	3	0.000	0.000	0.000	17
B007	0.507	1.000	0.673	73	0.421	0.216	0.286	19
B008	0.791	0.909	0.846	201	0.889	0.091	0.166	18
B009	0.759	0.960	0.848	158	0.941	0.128	0.225	17
B010	1.000	0.828	0.906	24	0.500	0.276	0.356	16
B011	0.500	1.000	0.667	6	0.000	0.000	0.000	18
B012	0.338	0.714	0.459	74	0.765	0.371	0.500	17
B013	0.375	0.900	0.529	120	0.700	0.280	0.400	20
B014	0.571	0.800	0.667	7	0.062	0.200	0.095	16
B015	0.500	1.000	0.667	2	0.000	0.000	0.000	10
B016	1.000	0.500	0.667	5	0.375	0.600	0.462	16
B017	1.000	1.000	1.000	13	0.125	0.077	0.095	7
B018	0.724	0.993	0.837	199	0.875	0.048	0.092	8
B019	0.617	0.954	0.749	201	0.684	0.100	0.174	19
B020	0.923	0.727	0.814	26	0.800	0.364	0.500	15
B021	0.562	0.968	0.711	162	0.818	0.096	0.171	11
B022	0.667	1.000	0.800	6	0.000	0.000	0.000	18
B023	0.684	0.950	0.795	196	0.778	0.050	0.093	9
B024	0.117	0.636	0.197	60	0.714	0.455	0.556	7
B025	0.610	0.943	0.741	82	0.722	0.245	0.366	18
Aver	**0.684**	**0.893**	**0.749**	**83**	**0.566**	**0.174**	**0.229**	**14**

Table 1: The table compares results of our answer generator combining the Indri and the Proteus ACE system, against the focused-summarization baseline. This experiment is over 25 previously unseen questions. The differences between the two systems are statistically significant ($p < 0.001$) for recall and f-measure by a two-tailed, paired t-test. A big difference between the two systems is that the answer generator produces a total of 2,086 answer sentences while sustaining an average precision of 0.684. In only three cases, does the precision fall below 0.5. In contrast, the baseline system produced only 362, one-sixth the number of answer sentences. While its average precision was not significantly worse than the answer-generator's, its precision varied widely, failing to find any correct sentences four times.

tation than an unordered list of sentences.

Finally, we will look into applying the techniques used here on other topics. The first test would reasonably be *Conflict* events, for which the ACE program has training data. But ultimately, we would like to adapt our system to arbitrary topic areas.

Acknowledgements

This material is based in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

Sasha Blair-Goldensohn and Kathleen McKeown. 2006. Integrating rhetorical-semantic relation models for query-focused summarization. In *Proceedings of 6th Document Understanding Conference (DUC2006)*.

Hoa Trang Dang, Jimmy Lin, and Diane Kelly. 2006. Overview of the TREC 2006 question answering track. In *Proceedings TREC*. Forthcoming.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. NYU's english ACE 2005 system description. In *ACE 05 Evaluation Workshop*. On-line at http://nlp.cs.nyu.edu/publication.

Sanda Harabagiu, Dan Moldovan, Christine Clark, Mitchell Bowden, Andrew Hickl, and Patrick Wang. 2005. Employing two question answering systems in TREC 2005. In *Proceedings of the Fourteenth Text Retrieval Conference*.

B. Katz, G. Marton, G. Borchardt, A. Brownell, S. Felshin, D. Loreto, J. Louis-Rosenberg, B. Lu, F. Mora, S. Stiller, O. Uzuner, and A. Wilcox. 2006. External knowledge sources for question answering. In *Proceedings of TREC*. On-line at http://www.trec.nist.gov.

Giridhar Kumaran and James Allan. 2007. Information retrieval techniques for templated queries. In *Proceedings of RIAO*. Forthcoming.

D. Metzler and W.B. Croft. 2005. A Markov random field model for term dependencies. In *Proceedings of ACM SIGIR*, pages 472–479.

D. Oard, T. Elsayed, J. Wang, Y. Wu, P. Zhang, E. Abels, J. Lin, and D. Soergel. 2006. Trec 2006 at maryland: Blog, enterprise, legal and QA tracks. In *Proceedings of TREC*. On-line at http://www.trec.nist.gov.

Patrick Schone, Gary Ciany, Paul McNamee, James Mayeld, Tina Bassi, and Anita Kulman. 2004. Question answering with QACTIS at TREC-2004. In *Proceedings of the Thirteenth Text Retrieval Conference*.

T. Strohman, D. Metzler, H. Turtle, and W.B. Croft. 2005. Indri: A language-model based search engine for complex queries (extended version). Technical Report IR-407, CIIR, UMass Amherst.

Ellen M. Voorhees and Hoa Trang Dang. 2005. Overview of the TREC 2005 question answering track. In *Proceedings of the Fourteenth Text Retrieval Conference*.