MITRE's Qanda at TREC-11[*]

John D. Burger, Lisa Ferro, Warren Greiff,
John Henderson, Marc Light[†], Scott Mardis, Alex Morgan

The MITRE Corporation
[†]The University of Iowa

{john, lferro, greiff, jhndrsn, mardis, amorgan}@mitre.org
marc-light@uiowa.edu

Introduction

Qanda is MITRE's TREC-style question answering system. Since last year's evaluation, principal improvements to the system have been aimed at making it faster and more robust. We discuss the current architecture of the system in Section 1. Some work has gone into better answer formation and ranking, which we discuss in Section 2. After this year's evaluation, we have done a number of ROVER-style system combination experiments using the judged answer strings made available by NIST. We report on some success with this in Section 3. We have also performed a detailed categorization of previous TREC results according to answer type and grammatical category, as well as an analysis of Qanda's own question analysis component—see Section 4 for these analyses.

1. TREC-11 System Description

Catalyst

Last year, Qanda was re-engineered to use a new architecture for human language technology called *Catalyst*, (Burger & Mardis 2002). Developed at MITRE for the DARPA TIDES program, the Catalyst architecture is specifically designed for fast processing and for combining the strengths of Information Retrieval (IR) and Natural Language Processing (NLP) into a single framework. Catalyst uses a dataflow architecture in which standoff annotations are passed from one component to another, the components being connected in arbitrary topologies (currently restricted to acyclic ones). The use of standoff annotations permits components to be optimized for just those pieces of information they require for their processing.

Major system components

Qanda has a by now familiar architecture—questions are analyzed for expected answer types, documents are retrieved using an IR system and are then processed by various taggers to find entities of the expected type in contexts that match the question. Below we describe each of the major components in turn.

- Question analysis: This component is run after the question has been subjected to POS and named entity tagging. It uses a simple grammar, currently hand-written, to identify important components of the question—see Section 4 below for more detail.
- IR wrappers: Catalyst components have been written for several IR engines, taking the results of the question analysis and formulating an IR query. For TREC-11, we used the Java-based Lucene engine (Apache 2002). Lucene's query language has a phrase operator, and also allows query components to be given explicit weights. Qanda uses both of these capabilities in constructing queries from the information extracted from the question. For TREC-11, the top 25 documents were retrieved.
- Passage processing: Retrieved documents are tokenized, and sentence boundaries are detected. Because some downstream components run more slowly than the rest of the system, Qanda scores each sentence by summing the log-IDF (inverse document frequency) of each word that overlaps with the question. Only those sentences with a sufficient score are passed on to the rest of the system.
- Named entity tagging: Qanda uses Phrag (Burger et al. 2002), an HMM-based tagger, to identify named persons, locations and organizations, as well as temporal expressions. Phrag is also used as a POS tagger for question analysis.
- Numeric tagging: A simple pattern-based tagger uses an extensive list of unit phrases to identify measures, as well as currency, percentages and other numeric phrases.

[*]This work was supported in part by the Advanced Research and Development Activity (ARDA) under contract number contract F19628-99-C-0001.

- Other taggers: We have a simple facility for constructing taggers from fixed word- and phrase-lists. These were used to re-tag many named locations more specifically as cities, states/ provinces, and countries. Qanda also identifies various other (nearly) closed classes such as precious metals, birthstones, various animal categories, etc.
- Answer formation and ranking: Candidates are identified and merged, a number of features are collected, and a score is computed—see Section 2.

Qanda's answer formation component attempts to find the best answer phrase as well as the best supporting context for that answer—the former may not be a substring of the latter due to candidate merging. For our TREC-11 submission, the answer phrase was used as the actual (scored) answer string, while the context was included as the secondary (unscored) justification.

2. Answer Ranking

Qanda only examines sentences that match the question sufficiently using the IDF-weighted overlap described above. It collects candidate answers by gathering phrasal annotations from all of the semantic taggers, and identifies the following features:

- *Context IDF Overlap*: Described above.
- *Context Bigram Overlap*: Raw count of word bigrams in common with the question.
- *IR Ranking* of the source document by the IR system.
- *Type Same*: Boolean, true if the candidate and expected answer types are identical.
- *Type Similar*: Partial credit if the candidate's type is "related" to the expected answer type, e.g., *COUNTRY* and generic *LOCATION*.[1]
- *Candidate Overlap*: Raw count of non-stop words in common between the candidate itself and the question, to bias against entities from the question being chosen as answers.
- *Minimal Overlap Distance*: Number of characters between the candidate and the closest non-stop question word in the context.[2]
- *Numeric Date*: 1 if the expected answer type is temporal and the candidate contains a numeric token, 0 otherwise, to bias against unresolved relative dates such as *yesterday*.

Candidates with similar textual realizations are merged, with the combined candidate retaining the highest value for each feature. Accordingly, an additional feature is maintained:

- *Merge Count*

After all of the (merged) candidates have been acquired, the raw feature values described above are normalized with respect to the maximum across all candidates, resulting in values between 0 and 1.[3] Features normalized in this way are more commensurate across questions, especially word overlap and related features (Light et al. 2001).

Each feature has a fixed weight, and a simple additive model is used to give each candidate an overall score. Our official TREC submission used (minimally) hand-tuned weights.

Log-linear models for answer scoring and confidence estimation

We are currently experimenting with acquiring the weights for the answer scoring component using logistic regression on past TREC datasets, resulting in a log-linear model, as has been used by Ittycheriah et al. (2001) and others. One issue is that because the model estimates a conditional probability (namely correctness given features of the question and candidate), the resulting scores are not necessarily commensurate across questions, and so the answers cannot be easily ranked for confidence, as required in TREC this year. Our current approach is to re-score the top candidate for each question using a second log-linear model. This uses features of the question, such as expected answer type, that do not affect the first model's candidate ranking, as well as features derived from applying the first model, such as the top candidate's original score, the total score mass given to the top *N* candidates, etc. These last features are similar to those used by Czuba et al. (2002).

[1] Currently this is done using a simple hand-built table, but with sufficient training data, we expect to use the log-linear model described below to acquire weights for most sensible pairs of types.

[2] Words would arguably be a more intuitive unit for this feature.

[3] The normalized values are computed so that the intuitively "best" feature value is 1, the worst 0—this is primarily for the developers' convenience, but also so weights are all positive, and more easily reasoned about.

3. System-Combination Experiments—Exploiting Diversity

Progress in question answering technology can be measured as individual systems improve in accuracy, but it is not the only way to witness technological progress. A question one can ask is how well we can perform automatic question answering *as a community*. If we were asked to enter an Earth English system in an intergalactic TREC, how well would we do? One easy answer is that we would perform as well as the best QA system. A second answer is that perhaps we could do even better by combining systems—this might be expected to work if different systems were independent in their errors. The follow-up question is how would we build such a system?

Lower bounds on the highest possible performance current technology can achieve on a given dataset have practical value, as well. They allow us to better estimate how well systems are doing with respect to the underlying difficulty of the dataset, and continually provide performance targets that are known to be achievable. Without such lower bounds on optimal performance, one cannot determine if technological progress in a domain has simply stalled.

NIST's ROVER system for combining speech recognizer output gives ASR researchers an updated goal to shoot for after every evaluation, as well as an implicit measure of the extent to which systems are making the same errors (Fiscus 1997). The work herein initiates a similar set of experiments for question answering technology.

Methods

The task we are faced with is straightforward. Given a collection of answers to a question, choose the one most likely to be correct. For our purposes, each answer consists of the answer string and an identifier for an associated document. Our data was limited in that it did not indicate which answers were provided by which system—see the discussion below. Note that we use no knowledge of the question or of the document collection. Our assumption is that the authors of the individual systems have milked the information in their inputs to the best of their capabilities. Our goal is to combine their outputs, not to re-investigate the original problem.

In this year's main QA evaluation there were 67 different systems or variants thereof involved. Thus, our corpus consists of 67x500 answers. To guard against any implicit bias due to repeated experimentation on the small dataset available, we randomly selected a 100-question subset for development of our techniques—the remaining 400 questions were kept as a test set, evaluated only once, when development was complete. While we may have wished to pursue parametric techniques, we felt that this training set was too small to explore any but the simplest (non-parametric) techniques. An exception is the experiments described below involving priors over the document sources, which still only involved four parameters.

Voting is an easily understood technique for selecting an answer from among the 67 suggestions. Unfortunately, voting techniques do not provide a mechanism for utilizing full knowledge of partial matches between proposed answers. While his original goal was the selection of representative DNA sequences, Gusfield (1993) introduced a general method for selecting a candidate sequence that is close to an ideal centroid of a set of sequences. His technique works for all distance measures that support a triangle inequality, and offers a bound that the sum of pairwise distances (SOP) from proposed answers to the chosen answer will be no more than twice the SOP to the actual centroid (even though the centroid may not be in the set). This basic technique has been used successfully for combining parsers (Henderson 1999). Appealingly, the centroid method reduces to simple voting when an "exact match" distance is used (the complement of the Kronecker delta).

One advantage of both simple voting and the centroid method is that they give values (distances) that are comparable between questions. An answer that receives 20 votes is more reliable than an answer that receives 10 votes, and likewise for generalized SOP values. This gives a principled method for ranking results by confidence and measuring average precision, as required for this year's TREC evaluations.

In selecting appropriate distance measures between answers, both words and characters were explored as atomic units of similarity. Two well-known non-parametric distances are available in the literature: Levenshtein edit distance on strings and Tanimoto distance on sets (Duda et al. 2001). We experimented with each of these, and also generalized the Tanimoto distance to handle multisets by defining a function to map multisets to simple sets: Given a multiset containing instances of a repeated element x we can create a simple set by subscripting, e.g., $<x,x,y,z>$ □ $\{x_1,x_2,y,z\}$. We can then use the standard Tanimoto

	Dev Set (100 Qs)				**Test Set (400 Qs)**	
	Strict		**Loose**		**Strict**	
	P	**avgP**	**P**	**avgP**	**P**	**avgP**
exact string match	50	70	54	74	42	65
word set	54	75	58	78	46	68
word bag	54	75	58	78	46	68
character set	51	65	57	67	46	62
character bag	60	81	64	85	50	**74**
word bag w/ doc priors	**66**	83	74	88	51	72
character bag w/ doc priors	64	81	69	86	50	72
5-character bag w/ doc priors, weighted numeric strings	**66**	**85**	**76**	**90**	**53**	73

Figure 1: Answer selection results (percentages, best results in bold)

distance on the resulting simple sets.[4]

Overall, systems seemed to be conservative and answered with the NIL document (no answer) at a rather high rate (17% of all answer strings this year). To compensate for this, a "source prior" was collected from the 100-question training set. These four numbers recorded the accuracy expected when systems generated answers from the four document sources (AP, NYT, XIE, and NIL). Those numbers were then used to scale the distance measures for the corresponding answer strings. Other than these priors, no other features of the document ID string were used.

Results

Several measurements were made to ascertain the quality of the various selection techniques, as seen in Figure 1. Precision, **P**, indicates the accuracy of the technique, the percentage of the answers that were judged to be correct. **avgP** is the main measure used by NIST this year—the average precision of all prefixes of the sequence of answers placed in order of high to low confidence. **Strict** corresponds to the correctness criterion used by NIST—the answer must be exact and justified by the referenced document (assessor judgment 1). The **Loose** figures discard these two criteria (assessor judgment ≥ 1). The **Loose P** measure was the one that was optimized during development.

In Figure 1 we see both development and test set results for answer selection experiments involving a sample of the distance measures with which we experimented. All of the design and selection of the distance measures was done using hill-climbing on the development set, and only after this exploration was complete was the performance on the test set measured. Two general observations can be made about these results (and others not shown): taking into account a prior based on the document source (including NIL) is useful, as is working with feature bags from the answers rather than sets. The best-performing selection system used all character strings of length 5 and less as features, combined with the multiset Tanimoto distance measure described above, and scaled with document source priors. Furthermore, a numeric string mismatch was weighted to be twice as costly as mismatching a non-numeric string. Question 1674 provides an example that contrasts this best selector with a simple voting scheme (exact string match):

> *What day did Neil Armstrong land on the moon?*
> *1969* (simple voting—incorrect)
> *July 20, 1969* (best measure above—correct)

While a plurality of systems answered with *1969*, many others answered with variants of the correct answer that differed in punctuation, as well as *on July 20, 1969*; *July 18, 1969*; *July 14, 1999*; even simply *20*. All of these, including the incorrect *1969*s, contributed to the correct answer being selected.

The disparity between the dynamic range of these systems on the development dataset and the test dataset suggests that the dev set sample size of 100 (6700 proposed answers and NILs) is too small to draw conclusions on the relative quality of selection techniques. Still, consistencies in rank orderings of selection techniques between the two datasets strongly suggest that these methods of system combination are effective.

It is important to note that in these experiments we did not have access to several useful evidence sources. First, this year's submissions included system

[4] Recall $D_T(S_1, S_2) = 1 - |S_1 \square S_2| / |S_1 \square S_2|$.

estimates on answer confidence, if only implicitly. The selection mechanism could take advantage of this by weighting each submitted answer string appropriately. Second, past TRECs show that some systems are reliably more accurate than others, and if each answer string were labeled with a system ID, even if anonymized, we could use system-level features in the selector, such as a simple prior. Given sufficient training, we might even take question features into account, learning that certain systems are better at certain types of questions. We would like to pursue the use of these and other evidence sources in the future.

4. Analysis of Questions and Answers

In this section, we report on a number of analyses we have performed, both on Qanda and on all-system results from previous TRECs. We describe the features Qanda extracts from questions, and evaluate its performance on one of these. We also describe a detailed categorization of the TREC-9 answer corpus with respect to semantic and syntactic types.

Question analysis in Qanda

Phrag, our HMM-based tagger, first annotates questions using separate models for part-of-speech and named entities. Qanda also runs a simple lookup-based tagger that maps head words to answer types in Qanda's ontology using a set of approximately 6000 words and phrases, some extracted semi-automatically from WordNet, some identified by hand. Based on these annotations, Qanda's main question analysis component uses a parser with a simple hand-optimized grammar to identify the following aspects of each question:

- Answer type: a type in Qanda's (rather simple) ontology, e.g., *PERSON* or *COUNTRY*.
- Answer restriction: an open-domain phrase from the question that describes the entity being sought, e.g., *first woman in space*.
- Salient entity: What the question is "about". Typically a named entity, this corresponds roughly to the classical notion of *topic* discussed below, e.g., *Matterhorn* in *What is the height of the Matterhorn?*
- Geographical restriction: Any phrase that seems to restrict the question's geophysical domain, e.g., *in America*.
- Temporal restriction: Any phrase that similarly restricts the relevant time period, e.g., *in the nineteenth century*.

As part of our post-TREC analysis, we have begun to examine how well Qanda performs on these various aspects. One way of evaluating this is to create an annotated test set of questions, tagged with the "correct" result, and score Qanda against this. For example, we might annotate *When did the art of quilting begin?*, with the answer type *LOCATION*—if Qanda's prediction matches this, its question analysis was correct in this instance. However, there is another approach to this evaluation. As described in the next section, we have annotated the TREC-9 answer key with semantic types, and so one might ask how often the system predicts one of the actual answer types, according to the answer key. For our example question, *medieval Europe*—a *LOCATION* answer—was judged to be correct by the TREC assessors. Had this been the only correct answer found, Qanda's prediction would be counted wrong, under the analysis we describe here.

In Figure 2 we present this analysis in terms of the question phrase used, and as a percentage of all questions in the set. This helps us to see which question types might have the biggest impact on our performance. For example, Qanda does rather well at predicting an answer type for *how many* questions, but these only constitute 5.44% of the questions in the set. On the other hand, of the 29.71% of the set that were unadorned *what* questions, Qanda's question

	Correct	**Incorr.**	**Total**
at what X	0.23	0.00	0.23
for what X	0.00	0.23	0.23
in what X	0.00	0.23	0.23
what in-situ	0.00	0.45	0.45
what kind	0.00	0.68	0.68
what type	0.00	0.68	0.68
what X	5.90	5.90	11.79
what	3.17	26.30	29.71
how hot	0.00	0.45	0.45
how large	0.00	0.23	0.23
how long	0.00	0.68	0.68
how many	5.22	0.23	5.44
how much	1.13	0.00	1.13
how tall	0.00	0.45	0.45
how wide	0.23	0.00	0.23
name	0.23	0.00	0.23
tell	0.00	0.23	0.23
when	9.07	0.00	9.07
where	12.70	0.91	13.61
who	20.41	1.59	22.00
why	0.00	0.23	0.23
Grand Total	**58.50**	**41.27**	**100.00**

Figure 2: Answer type correctness (percentage of all questions)

component performed very poorly. We hope to perform similar evaluations for the other question aspects listed above.

Manual answer analysis of the TREC-9 question corpus

Here we report on an analysis of the answers returned by all systems participating in TREC-9. Our study was done as part of a larger investigation, consisting of two levels: First, to identify Topic and Focus constituents for each question, and second, to characterize the Topic and Focus constituents by referent, and in the case of certain expressions, by grammatical type.

Before we explain what each of these levels of analysis entailed, we will first establish what we mean by Topic and Focus, as the terms and concepts are often used interchangeably in the Q&A literature. We use the terms Topic and Focus as they are defined in classic formal linguistics, dating back to the mid 19th century (see Hajicova 1984, for an early historical overview) and continuing on to recent times in linguistic schools such as Functional Grammar (Dik et al. 1981) and generative grammar (Rochemont 1986). Variably termed theme/rheme, topic/comment, presupposition/focus, they are defined in discourse theory roughly as follows:

Topic: The constituent(s) of a sentence identifying given, presupposed, or "old" information at the time of the utterance.

Focus: The constituent(s) of a sentence identifying what is new to the discourse at the time of the utterance.[5]

In questions, the *wh*-word is the Focus, and the rest of the question is typically the Topic. The answer to a question is also Focused. Question/Answer pairs have long been used in traditional Topic/Focus research papers to unambiguously illustrate and identify Topic and Focus constituents. E.g., from Dik et al. (1981):

(1) question: (a) ***What** did John buy?*
answers: (b) *John bought **an umbrella**.*
(c) ***an umbrella***

Bold is used in (1) to identify the Focus constituents; normal weight text indicates Topic constituents. Ordinarily, utterances such as (1a) would occur in a context in which John's buying activity were already presupposed. Earlier models of discourse did not anticipate the context in which humans would be entering factual questions into computers "out of the blue." However, since TREC has yet to intentionally introduce questions with false presuppositions, in our analysis we assumed the presuppositions were true and considered them Topic constituents.

Returning to the discussion of the analysis of the TREC question set, we identified the Topic and Focus constituents of each question, for example:

(2) <FOCUS>*Who*</FOCUS> <TOPIC>*invented the paper clip*</TOPIC>*?*

In addition, we used a REF attribute to classify the entity or activity REFerenced by the constituent, where the value for REF comes from an entity/activity ontology, shown in Figure 3 below. For certain expressions, we also used an EXP attribute to identify whether the EXPression is a name, descriptor, or directional phrase. Except for cases requiring a "direction" value (see example 5), EXP is typically only used for classic "Named Entities" such as persons, locations and organizations. Artifacts will also sometimes have an EXP attribute. Here is the previous example with these attributes marked:

(3) <FOCUS REF="person" EXP="name">
Who</FOCUS> <TOPIC REF="levin_26_4">
invented the paper clip</TOPIC>*?*

The markup in example 3 identifies the answer to this question as a named person and identifies the Topic of the question as a creation activity (levin_26_4 is the class of c*reate* verbs.) The annotation of the Topic constituents in the TREC-9 questions has not been finalized at this time, so in the remainder of this section we will discuss only the results of the Focus tagging.

In determining the value for REF and EXP in Focus constituents, we looked at the actual answers as recorded in an answer key we developed previously. This answer key[6] was compiled by manually examining all the answers returned by all of the TREC-9 systems. From those judged correct by the TREC assessors, we extracted short answer phrases. To perform the Focus analysis, we annotated the answer key itself, rather than the *wh*-word as shown in example 3, because there are often multiple correct answers to a given question.[7] We tagged each possible

[5] There are actually two types of Focus: Completive and Contrastive. Here we refer only to Completive Focus.

[6] See http://trec.nist.gov/data/qa/add_qaresources.html

[7] Multiple answers are due to two factors: different phrasings of the same correct answer and completely different correct answers. We did not distinguish between these two factors in our analysis of the answers.

- **Entity**
  - organism
    - person (includes deities)
    - animal (non-human)
    - plant
  - body_part
  - plant_part
  - organization
  - other_agent
  - celestial (e.g., Earth, Sun, Horsehead Nebula)
  - geological (e.g., mountain, river, continent, oceans)
  - gsp (Geo-Social political entity)
    - country
    - city (villages, towns)
    - province (counties, states)
  - recreational (e.g., parks, preserves, monuments)
  - other_location
  - facility
  - artifact
    - titled_work
      - book
      - movie
      - music
    - vehicle
    - award
    - instrument (musical)
  - substance
    - liquid
    - solid
    - gas
  - temporal
    - date
    - time

- **Abstract**
  - language
  - thought
  - utterance
    - translation
    - statement
    - description
    - question
  - technique
  - quantity
  - age
  - measure
    - mass
    - volume
    - area
    - length (height, etc.)
    - frequency (any type of rate)
    - temperature
    - weight
    - energy
    - illumination
    - duration
    - monetary
  - signal
    - appearance
      - color
      - shape
    - sound
    - sensation
    - flavor
    - scent

- **Disease**
- **Phenomenon** (e.g., physical phenomenon)
- **Manner** (e.g., slowly, well)
- **Mode** (by plane, by camel)
- **Event**
- **Activity**
  - Levin (1993) verb classes where possible, else FrameNet classes
- **Emotion** (feelings)
- **Stative** (being, having, spatial relations)
  - physiological (e.g., bodily symptoms such as fever and depression)
- **Nationality**
- **Weather** (e.g., rain, cloud, fog)

Figure 3: Entity and activity ontology for question analysis

answer as a Focus constituent, and applied the correct REF and EXP attributes. For example:

(4) *What is Francis Scott Key best known for?*
<FOCUS REF="levin_26_7">*penned the national anthem*</FOCUS>;
<FOCUS REF="music" EXP="descriptor">*the national anthem*</FOCUS>;
<FOCUS REF="music" EXP="name">*Star-Spangled Banner*</FOCUS>

(5) *Where did Woodstock take place?*
<FOCUS REF="city" EXP="name">*Bethel*
</FOCUS>;
<FOCUS REF="city" EXP="direction">
50 miles from Woodstock</FOCUS>

Metonyms, dangling modifiers, and similar expressions can occur as answer phrases, creating the difficulty that the literal interpretation out of context, versus the intended referent within the given context, may be distinct. Thus, a third attribute, LITREF, identifies the entity or activity referred to by the phrase in isolation. REF is used for the intended referent in the context of the question. For example:

(6) *What is the most common cancer?*
<FOCUS REF="disease">*skin cancer*</FOCUS>;
<FOCUS REF="disease" LITREF="body_part">
skin</FOCUS>

(7) *Name an American made motorcycle.*
<FOCUS REF="vehicle" LITREF="organization">
Harley-Davidson</FOCUS>

Question corpus analyses

We took the annotation of the answer key and collapsed all identically tagged answers in order to identify the set of unique answer types associated with each question. We consider an answer type "unique" if it differs by all three attributes (REF, LITREF, and

	Question Phrase					
	who	**what**	**when**	**where**	**how**	**name**
Number of questions	102	231	40	60	48	15
Number of answer types	8	63	2	13	12	13
Average number of answer types per question	1.19	1.19	1.03	2.57	1.02	1.60
Percentage of questions with more than one answer type	16.67	14.72	2.50	68.33	2.08	33.33

Figure 4: Range of answer types by question type

EXP). Thus an answer of type *PERSON NAME* is considered distinct from answer of type *PERSON DESCRIPTOR*. We also categorized each question by its *wh*-phrase (question phrase) to provide a rudimentary form of question typing. Some of the patterns that emerged are presented and discussed below.

Figure 4 shows the range of answer entities/activities associated with the major question types in TREC-9. The *what* questions exhibit the highest number of different answer types (63), but only 14.72% of the individual *what* questions have more than one answer type. This is because, although *what* questions have as their foci a broad range of entities/activities, each individual question is typically concerned with only a particular entity or activity. For example *What is platinum?* has four different answer phrasings, but they all refer to an entity of type *SOLID*.

In contrast, the *where* questions utilize only 13 answer types, but 68.33% of the *where* questions have more than one answer type. This is largely explained by the range of granularity that is acceptable as an answer, where a geological area, country, state, or city can suffice, as well as what we called *direction* expressions like *110 miles northwest of New York City*. Thus the granularity of the entity ontology has an effect here; had we grouped all of these under a single *LOCATION* category, the number of answer types for *where* questions would be greatly reduced.

As stated above, we consider answer types unique if the form of the answer (EXP= name, descriptor, or direction) differs. However, for individual questions, it is not very common to have answer types that differ only by the expression form. *Where* questions, which can have three values for EXP, exhibit the most cases of this: of the 60 where questions, nine (15%) have duplicate REF values but unique EXP values. For example, *Where are diamonds mined?* is answered variously by country name, country descriptor, geological name, and geological direction. *Who* questions come in second, but fairly low; of the 102 *who* questions, eight (8%) have answer types that differ only by EXP (person name and person descriptor). Of the 231 *what* questions, only two have both organization name and organization descriptor, and only one has both person name and person descriptor.

Figure 5 shows the top ten answer types for *what* questions, and Figure 6 does the same for *where* questions. The **(no answer)** label in Figure 5 reflects

Answer Type	**Percentage of *what* questions**
organization	11.64
person	8.73
animal	6.18
artifact	5.45
date	4.36
disease	4.36
(no answer)	3.64
geological	3.64
quantity	3.64
city	3.27

Figure 5: Top ten *what*-question answer types

Answer Type	**Percentage of *where* questions**
city	19.48
country	18.83
geological	18.18
province	15.58
gsp	6.49
other_location	6.49
facility	5.19
recreational	4.55
organization	2.60
body_part	0.65

Figure 6: Top ten *where*-question answer types

	EXPression Type			
	name	**descriptor**	**direction**	**(no value)**
who	68.60	21.49	0.00	9.92
what	36.73	9.09	0.00	54.18
where	70.13	4.55	11.04	14.29
name	41.67	4.17	0.00	54.17

Figure 7: Expression types for selected question types (percentages)

questions for which there were no answers in the key, because no systems answered them correctly.

For *who* questions, 80.17% of the answers were of type *PERSON*, 9.09% were *ORGANIZATION*, and 4.96% had no answer. All but one of the *when* questions had a *DATE* answer type—*When did the art of quilting begin?* had *medieval Europe* (a *GSP*) as one possible answer. *Name* imperatives (see example 7 above) display a range of foci, but 42% fall into one of three categories: *VEHICLE* (16.67%), *ORGANIZATION* (12.5%), and *OTHER_LOCATION* (12.5%).

Finally, Figure 7 shows the common EXPression types for those questions that can be answered with names. Many answers lack an EXP value because they refer to entities that do not typically bear names. However, the high number of answers with no EXP values also reflects the preliminary nature of this annotation scheme, particularly for the *what* and *name* questions. While unambiguous names were marked consistently as such, we were conservative in the use of the DESCRIPTOR value until we could see what entities emerged from the data. In the future, we will be refining the guidelines to make better use of the DESCRIPTOR value, and perhaps expanding EXP to include other values like *ADJECTIVE* and *ADVERB*.

Other analyses of question corpora

There have been many previous efforts at classifying questions. We mention a few here for comparison purposes. Weischedel et al. (2002) reported on an analysis of the combined questions of TREC-8, 9 and 10. They found a prevalence of people, locations, countries/cities/states, and definitions. Their cumulative results for all three TRECs are not directly comparable to what we've reported here, due to differences in the ontologies used, and also because our analysis is based on an examination of the answers rather than the questions. Hovy et al. (2000) use an ontology similar to the one in Figure 3. But where our ontology is used to characterize the Topic and Focus constituents, theirs represents the user's intention in asking the question, so that the ontology includes categories like *Why-Famous*. Thus, similar-looking tactics can have very different underlying approaches; One future goal is to apply multiple approaches to the same corpus, for a richer understanding of questioning and answering phenomena.

5. Conclusion

As well as the requisite description of this year's system architecture, we have discussed some preliminary work on log-linear models for answer selection and confidence estimation. We would like to pursue this further, using more features and more sophisticated models. We also presented promising initial results on question answering system combination—we will be exploring this further, hopefully making use of system-specific priors as well as confidence information in the answer selection.

We analyzed the TREC-9 answer corpus and examined the output of Qanda's question processing component with respect to those questions. This indicated some mismatches between the system's expectations about answer types and the actual answers found in TREC-9. We hope to remedy these problems, as well as subject other system components to such scrutiny. We would also like to analyze the TREC-11 answers in a like manner.

References

Apache Software Foundation, 2002. “Jakarta Lucene—Overview”. http://jakarta.apache.org/lucene/.

John D. Burger, John C. Henderson, William T. Morgan, 2002. “Statistical named entity recognizer adaptation”, *Proceedings of the Conference on Natural Language Learning.* Taipei.

John D. Burger, Scott Mardis, 2002. “Qanda and the Catalyst architecture”, *AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*.

Krzysztof Czuba, John Prager, Jennifer Chu-Carroll, 2002. “A machine-learning approach to introspection in a question answering system”, *Conference on Empirical Methods in Natural Language Processing*.

Simon Dik, Maria E. Hoffmann, Jan R. de Jong, Sei Ing Djiang, Harry Stroomer, Lourens de Vries, 1981. “On the typology of focus phenomena”, in Teun Hoekstra, Harry van der Hulst, Michael Moortgat (eds.), *Perspectives on Functional Grammar*. Dordrecht: Foris.

Richard O. Duda, Peter E. Hart, David G. Stork, 2001. *Pattern Classification*. John Wiley & Sons.

Jonathan G. Fiscus, 1997. “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)”, In *Proceedings of the European Conference on Speech Technology*, volume 4.

Dan Gusfield, 1993. “Efficient methods for multiple sequence alignment with guaranteed error bounds”, *Bulletin of Mathematical Biology*, 55(1).

Eva Hajicová, 1984. “Topic and focus.” In Jan Horecky (ed.), *Contributions to Functional Syntax, Semantics, and Language Comprehension*.

John C. Henderson, 1999. *Exploiting Diversity for Natural Language Parsing*. Ph.D. thesis, Johns Hopkins University.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Michael Junk, Chin-Yew Lin, 2000. “Question answering in Webclopedia”, In *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*.

Abraham Ittycheriah, Martin Franz, Salim Roukos, 2001. “IBM's statistical question answering system”, *Proceedings of the Tenth Text Retrieval Conference (TREC-10)*.

Beth Levin, 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Marc Light, Gideon S. Mann, Ellen Riloff, Eric Breck, 2001. “Analyses for elucidating current question answering technology”, *Natural Language Engineering* 7(4).

Michael S. Rochemont, 1986. *Focus in Generative Grammar*. Amsterdam: John Benjamins.

Ralph Weischedel, Scott Miller, Ada Brunstein, Robert Granville, Jonathan May, Lance Ramshaw, 2002. “Answering questions through understanding and analysis (AQUA)”. In notebook from *AQUAINT R&D Program Phase 1 Mid-Year Workshop*, Monterey CA, June.