RC22371 (W0203-040) March 13, 2002
Computer Science

IBM Research Report

Statistical Answer-Type Identification in Open-Domain Question Answering

John M. Prager, Jennifer Chu-Carroll
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Krzysztof Czuba
Carnegie-Mellon University
Pittsburgh, PA 15213

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Statistical Answer-Type Identification in Open-Domain Question Answering

John Prager
IBM T.J. Watson Research Center
Yorktown Heights, N.Y. 10598
jprager@us.ibm.com

Jennifer Chu-Carroll
IBM T.J. Watson Research Center
Yorktown Heights, N.Y. 10598
jencc@us.ibm.com

Krzysztof Czuba
Carnegie-Mellon University
Pittsburgh, PA 15213
kczuba@cs.cmu.edu

ABSTRACT

One of the most critical components of a question-answering system is the identification of the type, or semantic class, of the answer sought. Systems today use widely-varying numbers of such classes, but all must map the question to one or more classes in their inventory. In this paper, we present a statistical method of associating question terms with candidate semantic classes that has been shown to achieve a high degree of accuracy and to be applicable to different underlying semantic classifications.

Keywords

Question-Answering, Information Retrieval, Ontologies

1. INTRODUCTION

One critical component of a question-answering (QA) system is the identification of the type, or semantic class, of the answer sought. In all current top-performing QA systems (see e.g [Pasca and Harabagiu, 2001; Clarke et al., 2001]), the identified answer type is used in search and/or answer selection to help select appropriate answers to a given question.

These systems consist essentially of the following stages: 1) identify the answer type by analyzing the question, 2) process the question by elimination of stop-words, including the wh-words that indicate the question type, 3) perform morphological and other expansions on the question, 4) perform a bag-of-words search to return a number of documents or passages, 5) examine these texts for candidate answers of the earlier-identified type, and 6) rank these candidate answers with a metric based on linguistic, logical and statistical features. In our method of Predictive Annotation [Prager et al., 2000], we further utilize the answer type during indexing time by indexing the semantic class labels for each recognized named entity along with the text. During search time, we include in the bag-of-words the desired answer-type, thus guaranteeing that the resulting passages contain at least one candidate answer of the correct semantic class, in the context of the other terms in the question. Regardless of the usage of answer types in the rest of the system, however, answer type identification is integral to the selection of answer candidates in all current prominent question answering systems.

Although the technique of answer type identification is widely employed, there is no generally agreed upon classification used by these systems. The issues influencing the creation of such a set are granularity and coverage. For example, it is clear that for answering "Where" questions, a class of type PLACE is needed. In a context where there are likely to be questions of the kind: "What state …" and "Name a country that …", it seems reasonable that classes STATE and COUNTRY would be useful. However, this argument can be applied to an almost indefinite level of granularity, suggesting classes: CONTINENT, OCEAN, COUNTY, CAPITAL, CITY, TOWN, VILLAGE, ISLAND, PORT, STADIUM, DESERT, MOUNTAIN, RIVER, LAKE, STREET, BUILDING, PLANET and so on, not to mention the different kinds of organization (e.g. FACTORY, MUSEUM, SCHOOL) that can be locations. Furthermore, though no named-entity-based system would get far without the basic notions such as PLACE, PERSON, TIME and NUMBER, with appropriate subclasses, it is not so obvious whether classes such as COLOR, DISEASE and METAL are necessary too.[1]

The number of classes employed by systems participating in the QA track of TRECs 8-10 [Voorhees and Harman, 2000-2] ranged from approximately 10 to 100 (see, e.g. [Ittycheriah et al. 2001] and [Hovy et al. 2001]). However, there is no clear correlation between classes and system performance. We thus conclude that for the foreseeable future, systems will vary in the semantic classes they use, with no guarantee that a close-matching class exists for any given question. This paper addresses the problem of finding the "best" class to match a question's answer type from a system's own inventory of classes. Although in the rest of the paper, the examples draw on the semantic classes used in our system, the algorithm we discuss is independent of this classification and can be easily adopted by other existing systems.

We focus on fact-based questions seeking a named entity whose type is easily (at least to a human) determined from the question, for example, "What is the wingspan of a condor?" and "What metal has the highest melting point?". These questions are by-and-large either asking for the value of a property of an object or to select one item from an enumerable set. "Definitional"

[1] One can transform the coverage problem to a granularity problem by inventing general classes such as PROPERTY and THING.

questions (such as "What is a meerkat?") are not appropriate to this treatment since the question does not indicate the type of the answer being looked for; such questions can be answered by using different approaches, such as Virtual Annotation [Prager et al., 2001].

2. METHODOLOGY

To determine the mapping between an answer type specification in the question (such as **wingspan**) and a semantic class, we had previously employed a WordNet-based manual tagging method. We manually associated with about a hundred mid-level WordNet synsets [Miller, 1995] one or more semantic classes. Given a question term we traversed its hypernym tree until a tagged synset was found, and returned the associated semantic class(es). Thus to answer the TREC8 question "What debts did the Qintex group leave?" we would look up **debts** and find its ancestor **financial condition** which was mapped to our class MONEY.

There were two major problems with this manual tagging method. First, it requires deriving by hand the synset-class mapping, which is a labor-intensive task that resulted in many gaps in coverage (i.e. for some terms, none of its ancestors were associated with any semantic class). This coverage problem can be reduced by applying additional manual effort, but must be revisited whenever semantic classes change. The work must also be replicated if another ontology is used, and may not be feasible if a domain-specific ontology is unavailable. The second problem was that, especially for compound nouns such as **death toll**, not all terms had entries in WordNet. Using just the head noun was not necessarily effective either: the parentage of **toll** consists of the two branches: {**fee**, **fixed charge**, **charge**, **cost**, **expenditure**, **financial loss**, **loss**, **transferred possession**, **possession**} and {**value**, **worth**, **quality**, **attribute**, **abstraction**}. Neither branch is useful in directing a QA system to find numbers of people, or just numbers. Since this approach is neither automatable nor easily extensible, we investigated an alternative statistical approach.

Our new approach follows from the observation that frequently in text, entities are mentioned in conjunction with other entities of the same or similar[2] semantic class. This could be because an entity is contrasted with another entity, is presented in a list with another entity, or shares a relationship with some other entity. For instance, in "What is the wingspan of a condor?", the term **wingspan** occurs frequently with entities labeled LENGTH in the corpus, such as "9-foot". We hypothesize that local co-occurrence counts of the question term T and each semantic class label L_i used by the system will indicate the degree of association between T and L_i, and that those classes highly associated with T would be good candidate classes for T itself.

Our approach specifically targets questions that include a single or multi-word term T that explicitly names the answer type, such as **wingspan**. After extracting T, our goal is to determine, given a system's underlying semantic classification, the most plausible class(es) that represents answers of type T. For example, for "What was the death toll at the eruption of mount Pinatubo?", we extract **death toll** as the term T. Given our semantic classification, the correct classes are either WHOLENO or POPULATION, specifying either a cardinal number or, more specifically, a number recognized as population size.[3]

Our algorithm seeks to determine the semantic class of T based on the co-occurrence counts of T and each semantic class label L_i, relative to total occurrences of L_i in the TREC corpus. First, each class label receives a score based on a scoring function S, which is compared with the expected score for that class. The greater the score for L_i deviates positively from its expected value, the higher the degree of positive association is between T and L_i, and thus the better L_i is as a candidate answer type.

We implemented three variations of the scoring function as follows, where C represents the TREC corpus:

> S1 counts the number of co-occurrences of T with L_i in the same or consecutive lines of each document in C.
>
> S2 counts the number of documents in C where both T and L_i occur in the same sentence.
>
> S3 counts the number of documents in C where both T and L_i occur either adjacent or separated by a single word.

These scoring algorithms differ in two aspects. First, they differ in the window size that defines the context in which we search for co-occurrences. Second, they differ in terms of whether actual co-occurrence counts or the number of documents containing term co-occurrences are used. In S1-S3, all counts are transformed into a deviation from the expected frequency given a uniform distribution of class labels in the corpus. The class labels are then scored and sorted and the top classes may be used as candidate answer types for a QA system. Additionally, we implemented a fourth algorithm, AVERAGE, based on the outcome of the three S_j scoring algorithms.

The specific algorithms used are as follows.

S1 – Occurrences

Using the GREP utility we found all three-line text segments containing T in the middle line, and collected these segments in a file F. We ran our named-entity-recognizing indexer on F to generate an index whose statistics we could query. For each semantic class label L we calculated the ratio

$$R_1 = \frac{\text{occs of L in F} / \text{total terms in F}}{\text{occs of L in C} / \text{total terms in C}} \qquad \ldots\ldots(1)$$

This ratio represents, for each semantic class L, how its distribution in the text segments in F differs from its distribution in the TREC corpus as a whole. The larger this ratio is, the higher the co-occurrence is between T and L, and therefore the likelier L is as a candidate semantic type for the question term T.

[2] I.e. they may share a hypernym or meronym relationship.

[3] Our named entity recognizer/annotator can (and frequently does) tag text strings with multiple semantic labels.

Table 1 Results for "death toll"

	S1 - Occurrences		S2 - Sentences		S3 - Phrases		AVG - Average	
1	POPULATE	10.55	WHOLENO	3.00	WHOLENO	0.96	WHOLENO	0.22
2	SPEED	6.00	SMALLNO	2.74	SMALLNO	0.92	SMALLNO	0.21
3	SMALLNO	4.87	DISEASE	2.65	DISEASE	0.74	DISEASE	0.14
4	WHOLENO	4.86	THING	2.17	SPEED	0.61	LARGENO	0.14
5	LARGENO	4.82	LARGENO	1.98	LARGENO	0.37	SPEED	0.12

Table 2 Results for "cancer"

	S1 - Occurrences		S2 - Sentences		S3 - Phrases		AVG - Average	
1	DISEASE	83.78	CONSTELL	208.32	MEDICAL	24.91	DISEASE	0.39
2	MEDICAL	20.93	DISEASE	108.92	DISEASE	16.13	MEDICAL	0.26
3	CONSTELL	20.29	MEDICAL	53.03	SYMPTOM	2.66	CONSTELL	0.22
4	STAR	13.00	MEDICINE	19.39	ROLE	2.63	STAR	0.05
5	MEDICINE	12.29	SYMPTOM	14.08	STAR	2.63	MEDICINE	0.04

S2 – Sentences

For S2 and S3 we used a utility *doc_cnt* in our GuruQA search-engine package [Prager et al., 2000]; doc_cnt takes a GuruQA query and returns the number of documents that the query matches exactly. For S2 we used the query

```
@win(1 T L)
```

which finds occurrences of T and L within a window size of 1, i.e., when they co-occur in the same sentence. For each semantic class label L, if doc_cnt used as above returned a count of M documents, we calculated the ratio

$$R_2 = \frac{M/1000}{\text{docs containing L in C} / \text{total docs in C}} \quad \ldots\ldots(2)$$

S3 – Phrases

For S3 we used the GuruQA query

```
@win(1 @phr(1 T L))
```

which finds occurrences of T and L in the same phrase with at most one intervening term. For each semantic class label L, if doc_cnt as used above returned a count of N documents, we calculated the ratio

$$R_3 = \frac{N/1000}{\text{docs containing L in C} / \text{total docs in C}} \quad \ldots\ldots\ldots(3)$$

Equivalently, equations (2) and (3) are seeking the L which maximizes

$$|T \text{ and } L| / |L| \quad \ldots\ldots\ldots\ldots(4)$$

for the respective kind of intersection. The factor (1000/total docs in C) is constant and just serves to scale the numbers.

AVERAGE

We gathered the top 5 semantic categories according to each scoring scheme. We noticed that the best-performing scoring algorithm S_j depended on L_i, but in a non-obvious way, and also that the correct L_i for a given T would often occur in two or three of the top-5 lists, even if never in first place. We therefore decided to compute the average of the S_j. Since the scales of the S_j were not directly compatible, we normalized the S_j scores R_{ji} by dividing each score by the sum of the top 5 placed L_i. The AVG score for a given L_i is the average of its normalized R_{ji} scores. The sum of the top 5 AVG scores equals 1 only if the top 5 L_i entries for each of the S_j contain the same 5 L_i, in some permutation.

Scores for sample question-terms are presented in Table 1 and Table 2. It should be noted that for the S_j scores, the absolute values of the numbers are not as meaningful as their relative values, within a column. Values across columns cannot be compared. The absolute values in the AVG column, on the other hand, can be thought of as rough estimates of confidence.

3. EXAMPLES

As illustration of some of the properties and behaviors of our approach, we present in Table 3 the top-5 results for certain selected question terms. The rows in the table are split into three groups to facilitate discussion. For each question term, we show the top 5 semantic classes returned by the system, along with their respective scores using the AVERAGE scoring algorithm,

Table 3 Top 5 Results for Selected Question Terms

Question Term T	1st		2nd		3rd		4th		5th	
	Class	S	Class	S	Class	S	Class	S	Class	S
Wingspan	*LENGTH*	*0.67*	SPEED	0.12	WEIGHT	0.06	POWER	0.05	VEHICLE	0.05
Cancer	*DISEASE*	*0.39*	*MEDICAL*	*0.26*	CONSTELL	0.22	STAR	0.05	MEDICINE	0.04
Culture	*DURADATE*	*0.19*	*NATIONAL*	*0.16*	THING	0.14	ROLE	0.12	COMPOS	0.07
Season	SPORTSTEAM	0.34	*TIMEOFYEAR*	*0.19*	WEIGHT	0.12	TIME	0.10	SPORT	0.06
Bone	MUSICAL	0.38	DISEASE	0.28	MEDICAL	0.12	*BODYPART*	*0.10*	SYMPTOM	0.05
Plant	POWER	0.52	PROVINCE	0.08	ROLE	0.07	WEIGHT	0.07	AREA	0.06
Continent	*CONTIN*	*0.47*	PLACE	0.11	MOUNTAIN	0.10	REGION	0.09	OCEAN	0.08
Country	REGION	0.22	CONTIN	0.14	PLACE	0.10	ROLE	0.10	THING	0.08
State	NICKNAME	0.19	ROLE	0.14	PLACE	0.11	COUNTRY	0.10	POWER	0.10
Movie	*COMPOS*	*0.45*	THING	0.09	CONSTELL	0.09	PERSON	0.09	DURADATE	0.09
Film	*COMPOS*	*0.29*	PERSON	0.10	CONSTELL	0.10	DURADATE	0.10	THING	0.10
Best-selling book	*COMPOS*	*0.67*	PERSON	0.12	THING	0.16	RELIGION	0.05	YEAR	0.05
Novel	*COMPOS*	*0.28*	DURADATE	0.19	RELIGION	0.17	PHONE	0.15	PERSON	0.09
Sitcom	*COMPOS*	*0.60*	THING	0.09	ROLE	0.07	PERSON	0.06	MOST	0.04
Television show	*COMPOS*	*0.50*	MOON	0.17	THING	0.09	ROLE	0.07	PERSON	0.07
Song	*COMPOS*	*0.34*	STAR	0.26	MUSICAL	0.12	PERSON	0.09	THING	0.05

which indicates relative confidence. The semantic classes in italics are those considered correct for the given question term.

The first group of question terms in Table 3 contains examples that show certain system successes and failures. The clear winner for **wingspan** is LENGTH, as desired. Note that the runner-up candidates all are related in reasonable ways, in particular when the word is used in the context of wingspans of aircrafts. The results for **cancer** ("What is the most common cancer?") returned highly relevant semantic classes that represent diseases and medical conditions in first and second places. Interestingly, the two semantic classes in top 5 positions that are considered irrelevant for this example are STAR and CONSTELL(ATION), which correspond to a different, but also fairly common sense of the word. **Culture** ("What culture developed the idea of potlatch?") did well, in our opinion, with a fuzzy category. Its best candidates, DURADATE[4] and NATIONAL, address the temporal and spatial aspects of the question. For the next two question terms, the correct semantic class is found, but not in the top position. For the question term **season** ("What is the busiest air travel season?"), the system found the correct semantic class TIMEOFYEAR in second position. The first place answer is SPORTSTEAM, at first glance an unlikely candidate. However, we believe this is because journalists often compare sports teams by season, or refer to a team's performance in the "1995 baseball season". On the other hand, when the four seasons are mentioned in text, the word "season" is so strongly implied by "spring" and "summer" that it is often not explicitly mentioned. The next two examples, **bone** and **plant**, suffer from problems with polysemy, which we elaborate further in Section 5.

The second group of examples contains three question terms, **continent** ("What continent is Bolivia on?"), **country** ("In what country is a stuck-out tongue a friendly greeting?"), and **state** ("What state produces the best lobster to eat?"). Of these three terms, the system found a suitable semantic class only for **continent** (not considering the generic semantic class PLACE). Similar to the **season** example, we presume that this is because we talk about "the African continent" but not the "French country" or the "Massachusetts state", for example[5]. We will discuss this issue of failure due to stylistic factors in Section 5. However, note that the scores for the top candidates in the **country** and **state** examples are relatively low, compared to most other examples for which the top semantic class is correct.

The final group of examples illustrates a wide range of question terms for which our system returns COMPOS, our generic class for titled works (compositions), as the top candidate semantic class. These question terms include **movie**, **film**, **best-selling book**, **novel**, **sitcom**, **television show**, and **song**. Note that not only did our system return COMPOS for all these examples, it also did so with high confidence. This set of examples illustrates the powerfulness of our methodology --- though it is possible to manually derive pattern-matching rules to associate question terms with candidate semantic classes, we conjecture that terms such as **sitcom** is unlikely to be in a QA-system's pattern inventory for all but those with the most complete coverage.

[4] DURADATE is the intersection of DURATION and DATE – time references that can be viewed either as points in time or extents, such as "The 12th century" and "The Renaissance".

[5] Although admittedly the expression "state of X" is used somewhat, a rough calculation shows it to be used less that 2% of times a state is mentioned.

4. EXPERIMENTAL RESULTS

To evaluate the performance of our algorithm, we chose questions from the TREC9-10 QA tracks[6] that fit our criterion, that a term in the question explicitly states the answer type. These questions fall into two categories: *Set* questions, which seek a member of a set of instances ("What metal has the highest melting point?") and *Property* questions, which seek the value of a property ("What is the wingspan of the condor?").

We applied our algorithm to all qualifying questions using all three scoring functions, plus the average. The algorithm returned the top 5 candidates, and each question received a score equal to the reciprocal rank of the correct answer (or 0 if it failed to show). The scores in the following tables are the mean reciprocal ranks (MRRs) in each test condition. The column labeled N shows the number of questions processed (in Table 5 and Table 7, multiple questions with the same term T are counted only once).

Table 4 MRRs for TREC9 questions, counting instances

	N	S1	S2	S3	AVG
Property	27	.63	.66	.70	.67
Set	106	.42	.55	.62	.60
All	133	.46	.58	.64	.62

Table 5 MRRs for TREC9 questions, counting types

	N	S1	S2	S3	AVG
Property	15	.53	.56	.64	.61
Set	73	.50	.65	.67	.72
All	88	.50	.63	.66	.70

Table 6 MRRs for TREC10 questions, counting instances

	N	S1	S2	S3	AVG
Property	49	.72	.81	.91	.89
Set	103	.33	.42	.48	.46
All	152	.46	.55	.62	.60

Table 7 MRRs for TREC10 questions, counting types

	N	S1	S2	S3	AVG
Property	21	.65	.74	.83	.88
Set	49	.40	.49	.56	.55
All	70	.48	.56	.64	.65

In most cases, *Property* questions fared much better than *Set* questions. Furthermore, S3 universally outperformed S2, which in turn outperformed S1, suggesting that very close proximity is a key feature for success. In about half of the experimental conditions the AVERAGE column performed best. We plan to investigate whether the scoring functions combined with the manual tagging method will lead to further improvement.

[6] These questions are familiar to the community, have answer sets, and are derived (by NIST) from logs of questions from real users.

5. DISCUSSION

The results show that our methodology works very well on the kinds of factual questions in TREC QA - on average, a correct answer type is proposed in first or second place. The errors typically fall into two categories: those resulting from polysemy and those resulting from stylistic or domain-dependent factors. The first category is typified by **plant**, as in "Material called linen is made from what plant?" Our algorithm strongly suggests type POWER, since most mentions of **plant** in the TREC corpus are about power plants and their power output. The second category is typified by **bone**, as in "What is the longest bone in the human body?" Instead of BODYPART, the algorithm suggests MUSICAL (instrument), due to polysemy of words such as "pipe", "horn", "wood", "organ" etc., and DISEASE, since mentions of bone diseases are more common than phrases such as "the ulna bone". Examples such as **season**, **country**, and **state** discussed earlier in the paper also fall into this category. This contrasts with **metal**, which frequently occurs in sentences such as "Among the precious metals, platinum futures showed …". We observe that *Set* questions are much more prone to such problems than *Property* questions.

5.1. Local Context Analysis

It was hypothesized that these problems, particularly those deriving from polysemy, can be ameliorated by using the context of the entire question to disambiguate senses, following the work in Local Context Analysis of Xu and Croft (1996). To test this hypothesis, we used the original questions (from which our focus terms T were derived) to query the corpus and generate document sets. We used hit-lists of size 1000, which resulted in sets of 1000*n* documents, where *n* was the number of questions focusing on a given query term. We used these document sets as substitutes for the TREC corpus C in equations (1)-(3); we indexed the document sets and applied the four scoring functions described in Section 2. Our results were (by type) almost uniformly worse than using the entire corpus.

The reasons for the deterioration appear to be many, but with two main variations, both concerning the frequency of the question term T. Based on equation (4), in order for the score for the correct semantic class L to improve with the application of local context analysis, either 1) T and L must co-occur more frequently in the sub-corpus than in the entire TREC corpus, or 2) L itself must occur less frequently in the sub-corpus than in the whole corpus. When the question term T is frequent, we observe that the number of co-occurrences of T and each semantic class tend to remain fairly constant between the sub-corpus and the main corpus. On the other hand, when T is very infrequent, its co-occurrence with each semantic class is constant – all of its instances occur in the filtered corpus. In either case, the numerator of (4) will be constant (modulo scaling). However, since the filtered corpus is biased toward the subject-matter of the question term, the denominator of (4) will tend to be the maximum possible for highly correlated classes, i.e., |L| will become greater for more relevant semantic classes, resulting in lower scores for the very semantic classes whose scores we are seeking to improve!

5.2. Frequency Correlation

We hypothesize that explicit mention of the semantic class with a term may be correlated with the frequency of that term. Noticeably, the two most frequently occurring terms (**year** and **state**) each got a score of 0, since it usually goes without saying that, for example, 1960 is a year and that California is a state. To validate our hypothesis, we plotted the score for a question term against the document frequency (DF) of the term, with appropriate smoothing. Figure 1 shows the MRR of a moving window of 30 terms against log of DF. Figure 2 shows a "cumulating" MRR for all question terms of equal or lesser DF. This allows one to see what the effective accuracy of the system is if applied to all question terms with a DF below a given threshold.

Despite the obvious kinks in the curves, there is a clear downward trend of MRR with DF, although it appears that two or more competing forces are at play. The steep rise in both figures at log DF of around -5 is an artifact of the smoothing. In fact there is a jump from average performance of about .7 to near-perfect performance for all terms with log DF -5.0 to -3.5. This is illustrated in Figure 3, which is like Figure 2 but only applied to terms with log DF greater than -5 (i.e. occurring in more than 1 in every 150 documents). The cumulating MRR for terms in this range is clearly monotonic.

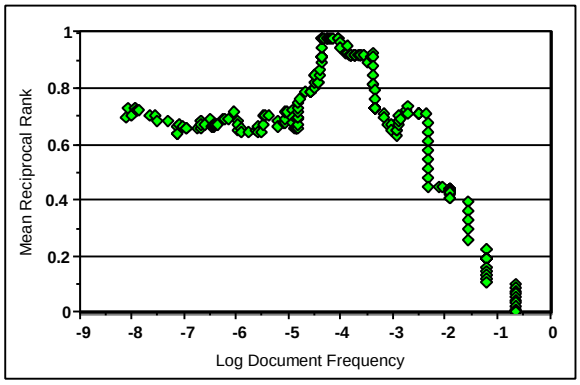

Figure 1. Question terms arranged in increasing order of their DF. Graph shows the MRR of a sliding window of 30 terms against the log DF of the right-most term in the window.

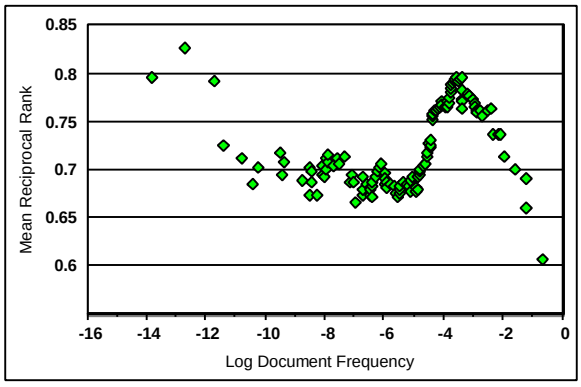

Figure 2. With the same ordering as in Figure 1, the MRR of all of the terms to the left of a given term is plotted.

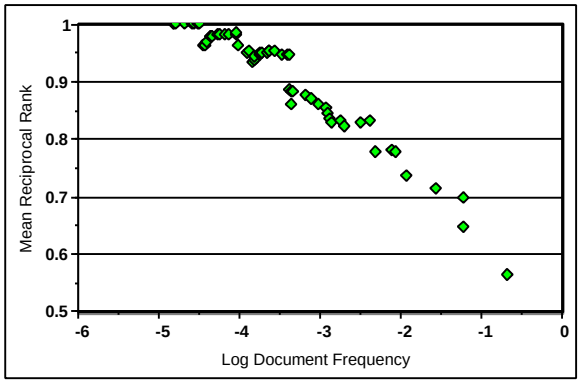

Figure 3. As Figure 2, but only for the most frequent terms.

It seems quite likely from these figures that there are at least two different phenomena, or possibly a single phenomenon with different controlling parameters in different frequency ranges. It is possible there is some relation with Rosch et al.'s [1976] Basic Categories, which is to say that the most natural level of description of an object is both used most frequently and is least in need of a semantic description. A full investigation of these phenomena is beyond the scope of this paper.

We can use the results presented in these figures to choose a performance threshold for deciding when to apply our algorithm, based on how frequently the question term occurs in text.. For example, if we choose a threshold of log DF = -2.5, corresponding to a frequency of less than 1 document in 12, we can predict an MRR of greater than .6 for any such question term, or an average MRR of about.75 across all such terms.

6. SUMMARY

We presented a statistical method of determining the appropriate semantic types corresponding to a term given any underlying semantic classification. When utilized by a QA system, our algorithm establishes the identity of a candidate answer type sought by a natural language question. We have shown that our algorithm, on average, proposes a correct answer type in either first or second place on a subset of the TREC9-10 questions. Furthermore, for especially the more frequent terms, the rarer the term, the better the algorithm performance. These figures provide a thresholding mechanism for a given desired success rate, based on the corpus document frequency of the term.

7. ACKNOWLEDGMENTS

We would like to thank Dave Ferrucci and the anonymous reviewers for their very helpful comments and suggestions. This work was supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program under contract number MDA904-01-C-0988.

8. REFERENCES

[1] Clarke, L.A., Cormack, G.V. and Lynam, T.R. "Exploiting Redundancy in Question Answering", *Proceedings of SIGIR 2001*, New Orleans, LA, 2001.

[2] Hovy, H., Gerber, L., Hermjakob, U. Junk, M. & Lin, C-Y. "Question Answering in Webclopedia. In *Proceedings of the 9th Text Retrieval Conference (TREC9)*, Gaithersburg, MD, 2001.

[3] Ittycheriah, A., Franz, M., Zhu, W-J., Ratnaparkui, A. and Mammone, R.J. "Question Answering Using Maximum Entropy Components", *Proceedings NAACL* , Pittsburgh, PA, 2001.

[4] Mihalcea, R. and Moldovan, D. "A Method for Word Sense Disambiguation of Unrestricted Text", *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, pp. 152-158, College Park, MD, 1999.

[5] Miller, G. "WordNet: A Lexical Database for English", *Communications of the ACM* 38(11) pp. 39-41, 1995.

[6] Pasca, M.A. and Harabagiu, S.M. "High Performance Question/Answering", *Proceedings of SIGIR 2001*, New Orleans, LA, 2001.

[7] Prager, J.M., Brown, E.W., Coden, A.R., and Radev, D.R. "Question-Answering by Predictive Annotation", *Proceedings of SIGIR 2000*, pp. 184-191, Athens, Greece, 2000.

[8] Prager, J.M., Radev, D.R. and Czuba, K. "Answering What-Is Questions by Virtual Annotation", *Proceedings of HLT 2001*, pp. 26-30, San Diego, CA 2001.

[9] Radev, D.R., Prager, J.M. and Samn, V. "Ranking Suspected Answers to Natural Language Questions using Predictive Annotation", *Proceedings of ANLP'00*, Seattle, WA, 2000.

[10] Rosch, E. et al. "Basic Objects in Natural Categories", *Cognitive Psychology* 8, pp. 382-439, 1976.

[11] TREC8 - "The Eighth Text Retrieval Conference", E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD, 2000.

[12] TREC9 - "The Ninth Text Retrieval Conference", E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD, 2001.

[13] TREC2001 - "The Tenth Text Retrieval Conference", E.M. Voorhees and D.K. Harman Eds., NIST, Gaithersburg, MD, to appear in 2002.

[14] Xu, J. & Croft, W.B. "Query expansion using local and global document analysis", *Proceedings of SIGIR '96*., pp 4-11, 1996.