Evaluating Entity Models on the TREC Question Answering Task

Hema Raghavan and James Allan
{hema,allan}@cs.umass.edu

ABSTRACT

We propose *entity models*, a representation of the language used to describe a named entity (person, organization, or location). The model is purely statistical and constructed from snippets of text surrounding mentions of an entity. We evaluate the effectiveness of entity models for fact-based question answering. The results obtained on question answering are promising indicating that entity models contain useful information which would aid textual data mining and other related tasks.

1. INTRODUCTION

To find out who someone is, we ask friends, read books, search libraries, browse the Web, etc., looking for information that describes the person. The more information we have gathered, the better a picture we develop. We might find out the person's career, what they are known for, who they have associated with, when they lived, and so on. Our picture of a person's "meaning" is constructed from numerous passages of text.

Inspired by that idea, we propose *entity models*, models of people, places, and other entities, based on how they are described. Our model is completely unstructured and based only on the text in our corpus. We do not employ any deep natural language processing beyond currently available off the shelf techniques for locating likely names nor do we use a knowledge base to improve our representation.

Fundamentally, an entity model is a collection of words or phrases that describe the way that a name is discussed. We collect all references to a name and consider the text surrounding the mention. That data provides us with an estimate of the likelihood that a word will be used in the context of a person. Our hypothesis is that the high frequency words will provide a useful representation of who a person is.

We evaluate the effectiveness of our model on the TREC question answering track. Our modeling approach provides an interesting new way to represent a person (or other entity), and it has broad applicability, with question answering being only one such technique. Since our model uses the text directly, without deep processing, we expect that it can be ported to new domains (i.e., not just news) with little difficulty. We demonstrate that the model is reasonably successful at the question answering task, suggesting that they may help in an even broader set of activities.

In the next Section we overview related work towards modeling entities. Following that, in Section 2, we formally describe entity models. Section 3 discusses the application of entity models to question answering and Section 4 defines the set-up for the same. We discuss our results in Section 5 and conclude and outline future directions for the work in Section 6.

1.1 Related work

Conrad and Utt [9] used fixed-sized windows around a name to build a pseudo-document of text that represents the name. They used these pseudo-documents as a method to retrieve names in response to queries and as a way to find connections between names. In their case, queries were names of people and organizations and one could retrieve the names of other people and organizations associated with the name that formed the query.

Conrad and Utt's pseudo-documents form a type of model that is very similar to the entity models that we develop here, and which clearly demonstrated some potential value in the approach. Our work introduces a more rigorous evaluation of the model using the TREC Question Answering task to verify the utility of an entity model. Additionally, Conrad and Utt built their system to handle only two types of entities–people and organizations. We extend that framework to 24 types of entities, including places, animals, substances, dates etc. They considered only one retrieval model in their paper and no evaluation of the retrieval performance. We consider several retrieval models and present an evaluation of each.

Researchers have studied lexical, phrasal, co-occurrence and dependency relations that can be pulled out from spans of text [13, 8]. Pseudo-relevance feedback methods find words that are related to the query term to improve the effectiveness of retrieval [25, 19, 15] . Those uses of statistics and others like them are similar to our building of Entity Models in that they find related words. We know of no work that viewed the probability distribution of text around a word as a model of its meaning. *Use theory*[24] states that the meaning of a word depends on how it is used in language. Therefore, words with similar local contexts are related. We extend that concept to *entities*, that is, the meaning of an entity is determined by the words in the local context of that word. For example, one would expect the words *information* and *retrieval* to often occur in the context of *Gerard Salton*.

As regards the question answering aspect of this paper, there has been a significant amount of research in the field. The TREC-8, TREC-9, and TREC-2002 proceedings reflect the progress and the state of the art in question answering. The best systems [22, 23] rely on a large amount of syntactic and semantic analysis of questions and answer candidates . They also make use of databases of factual

Submission for SIGIR 2004. Please do not cite or distribute his draft paper.

information [11], thesauri [18] and several hand-crafted rules. In this paper we use question answering as a means to evaluate entity models, without the use of any knowledge bases. We are looking at the question answering task, not from the perspective of achieving the best possible performance on a given TREC track, but with the view of finding underlying statistical properties of text which would be useful for question answering. We believe that an entity language model represents a unit of information that has interesting statistical properties, that can be put to several uses one of which is question answering.

2. ENTITY LANGUAGE MODELS

We define an entity model to be a collection of words or phrases that are likely to be used to describe the named entity. For example, an entity model for *George W. Bush* would have *president*, *republican*, *conservative*, and other such words with high frequency. It would also include names of strongly associated people (e.g., *Dick Cheney*), places (*Texas*), actions (*cut taxes*), and so on.

Given a large corpus of text, we construct a model for a named entity E as follows. First we find all occurrences of E in the corpus. *Snippets* of text consisting of $\pm n$ words around the mention of E are extracted. All snippets in the corpus that arise from the mention E are pooled together to form a pseudo-document of entity E. We naively pool all mentions of E together. For example two mentions of *Ford*, one which refers to the organization (company) and one which refers to the person *Ford* would contribute to the same entity model. One could do some kind of coreference resolution. However we left that for future work. If a name had multiple words, the entire multi-word string was treated as a single mention. Hence, a contiguous occurence of *Henry Ford* contributed only to the *Henry Ford* model, and not to the *Henry* and *Ford* models seperately.

The pseudo document obtained in this way gives us a maximum likelihood estimate or raw term frequency for any word that appears around mentions of E. Different document models may be suited to different tasks. Depending on the document model (vector space or probabilistic), collection statistics and smoothing can be used to obtain the final representation or entity model. We have used the two terms *entity models* and *pseudo-documents* interchangeably in this work. However, the meaning is clear from the context in which these two terms appear.

Some entities have a few snippets contributing to their model, and others can have several thousands. Sometimes several sources may contribute almost identical snippets to the pseudo document. The addition of a large number of similar snippets may cause problems due to over counting of some words. It may be possible to use some more optimal subset of the snippets to construct the entity language models. However for this work we did not resort to any such techniques, simplicity and Ockham's razor being our rule of thumb.

One way of looking at our approach is that we are transforming the original corpus into a new corpus comprised of entity models – one for each entity. One can then apply standard retrieval , clustering, classification and other techniques on this new corpus to tasks which are more oriented towards solving knowledge discovery and data mining problems.

3. QA TASK DEFINITION AND APPROACH

3.1 The TREC QA track

Since 1999, TREC has had a Question Answering Track [21] which evaluates a system's ability to answer open domain, closed-class questions. Systems are expected to retrieve small snippets of text, which contain answers to the different questions. For each question, a system is expected to retrieve exactly one answer. Some questions have no answer in the database. Systems are expected to report 'NIL' for those types of questions. Systems are compared on the basis of the number of answers they got right, and the accuracy (precision and recall) with which they return 'NIL'.

3.2 Our Approach

Our approach to question answering is simple. In traditional information retrieval, the basic unit of information we are trying to retrieve is a document. In question answering the basic unit of information that one is trying to retrieve is an exact answer. Each entity is a potential candidate for an answer.

In this paper we view the problem of question answering as a problem of retrieval on the corpus composed of entity models. Retrieval may be performed using either vector-space methods, or language modeling methods. The pseudo-documents or entity models described above contain snippets of text which convey information about an entity. An entity model can be considered to be a unigram model of words describing an entity. Additionally n-gram, phrasal and other contextual information may be obtained from the snippets of text that make up the entity model. Thus, a large number of retrieval techniques may be applied on the entity model corpus. In this way we transform the original QA problem into an information retrieval problem, and then use standard information retrieval techniques to solve the original problem. For example consider the TREC question *1744. What car company invented the Edsel?*, the answers to which is *Ford*. The words *car*, *company* and *edsel* occur with sufficiently high frequency in the pseudo-document for *Ford* and therefore the answer is retrieved by our system.

3.3 Retrieval Models

In this paper we use three standard information retrieval techniques implemented in the LEMUR[5] toolkit.

The first one is the traditional vector-space model, which computes vectors for documents and queries, and the similarity of a document to a query can be measured by any similarity metric that is applicable in vector space like the cosine measure or the Euclidean distance. We experiment with two different vector space models with TF*IDF weights. For one system we use the log of the term frequency for TF values, and for the other system we use OKAPI BM25 weights [19].

The second search engine that we use is INQUERY [7], which allows for structured queries. The underlying mechanism of INQUERY is a belief network and it permits the use of boolean operators such as #and, #or, and contextual operators like #n and #phrase.

Operator	Action
#and	AND terms in the scope of the operator
#or	OR the terms in the scope of the operator
#not	NEGATE terms in the scope of the operator
#sum	Mean of the beliefs of the arguments
#n	A match is found if all words in the operator are found in order with less than n words between adjacent words
#phrase	Value is a function of the beliefs returned by the #3 and #sum operators
#syn	The argument terms are synonymous

The intuition behind using INQUERY as one of the search engines is that it allows us to experiment with a variety of queries, which may prove useful in querying our pseudo document database.

We experiment with two models from the language modeling domain – the query likelihood model [17], and Relevance Model number one[15]. The query likelihood model assumes that the user generates a query (a question in our case) that is most representative of the document (an entity) that represents his or her information need. The system estimates the document that is a most likely representative of that ideal document as follows:

$$\text{arg max}_d P(d|q) = \text{arg max}_d P(q|d)P(d) \qquad (1)$$

The prior P(d) is assumed to be uniform. The original query likelihood model was first proposed by Ponte and Croft [17] and produces results that are comparable with traditional TF-IDF like systems. Language models are becoming increasingly popular in the information retrieval community as they fit in a nice mathematical framework.

A probabilistic model that has proven to be more effective for IR than the query likelihood model is the relevance model. The relevance model builds on the classic probabilistic model [20] of retrieval which suggests that the optimal ranking of documents is one where the ranking is done by the ratio $P(d|R)/P(d|NR)$, where R and NR are relevant and non-relevant classes of documents respectively. In the generative approach the only notion of relevance is from the original query and therefore it is assumed that $P(d|R) \approx P(d|q)$. In the original paper on relevance models, Lavrenko and Croft [15] describe two ways of estimating the probability of relevance. In this paper we use Relevance model (method 1) for which $P(w, q)$ is estimated as

$$P(w, q_1 ... q_n) \quad = \quad \sum p(d)p(w|d) \prod_{i=1}^{m} p(q|d) \qquad (2)$$

For both methods–Query Likelihood and Relevance Model 1 – the similarity of the document to the query is given by the Kullback Liebler divergence of the relevance model or query model with respect to the document model.

$$KL(R||D) = \sum_w P(w|R) log \frac{P(w|R)}{P(w|D)} \qquad (3)$$

The KL divergence or relative entropy is a measure of how two probability distributions differ from each other. Documents are ranked in increasing order of their KL-divergence from the query model or relevance model as the case may be.

4. EXPERIMENTAL SETUP

4.1 Data

We use the AQUAINT corpus which was used for the TREC 2002 and TREC 2003 evaluations, and which consists of about 3GB of newswire text from three sources–New York Times News Service, Associated Press News Service, and Xinhua News Service. There are about a million documents in the whole corpus. For the question answering evaluation we used 500 questions from the TREC-2002 main task, and 380 factoid questions from the TREC-2003 main task. The TREC-2002 questions were used for training and development and the TREC 2003 questions were used as a held out test set.

4.2 Question Answering evaluation

TREC systems are expected to return exactly one *[answer string,-doc id]* pair for each question. The *answer string* contains the exact answer to the question, and the *doc-id* is that of a document that supports the answer. Determining whether a string returned by a system is an acceptable answer is subjective and different human assessors may vary in their opinion. For evaluation, TREC pools the results of all participants, and evaluators are asked to judge the pooled list of answers for each question, and mark each answer as acceptable or not. From the accepted answers, a list of patterns or regular expressions is constructed for each question and that list is used for subsequent evaluations. We used the list of patterns as provided by TREC to get a good idea of what the correct answer was. However that list is not exhaustive as it only allows for valid answers that were retrieved by at least one of the TREC systems that year. If there was a valid answer that no system retrieved that year, then that answer would be left out from the list. Therefore, we also went through the answers as provided by our system to correct the patterns to be able to accept correct answers that our system retrieved and were not retrieved by any system that year. (We had to make almost no changes).

For the TREC-2003 evaluation questions, NIST has provided a list of 380 answer patterns for the main task to TREC participants. None of these 380 questions have NIL as the correct answer. The website [4] describes this list to be "less than official". [1]

For each of our experiments we report the accuracy at rank one, that is, the fraction of times that a system retrieved the correct answer at rank one. We also report the Mean Reciprocal Rank, which is the mean of the reciprocal of the rank at which the answer was found. We used the top 10 retrieved answers in our MRR computations, i.e., if an answer was not retrieved in the top 10 list of answers for a given query, it contributed a value of 0 to the MRR score (that is, its reciprocal rank was 0). The MRR score as we used it is given as

$$\begin{aligned}
MRR &= \frac{1}{T}\sum_{i=1}^{T} 1/R' \qquad (4)\\
R' &= R \text{ if } R < 11 \text{ else}\\
R' &= 0\\
&\text{where}\\
R &= \text{The rank at which the answer was found}\\
T &= \text{Total number of questions in the evaluation}
\end{aligned}$$

4.3 Construction of the Entity Models

We ran IdentiFinder$^{\text{TM}}$[6] (version 5.0) on the AQUAINT corpus. IdentiFinder$^{\text{TM}}$is an off the shelf tool for named entity tagging. It identifies 24 different categories of entities and has a Hidden Markov Model at its core. A list of the categories that it can tag is given in Appendix A. IdentiFinder$^{\text{TM}}$recognized 2,122,126 unique entities in the AQUAINT corpus. We used a value of $n = 10$ and extracted 20 word snippets around each mention of an entity. For each unique name we built a pseudo-document by collapsing all the snippets corresponding to that name together, giving us a total of 2,122,126 pseudo documents or Entity Models. We indexed this collection of pseudo documents using the LEMUR toolkit [5]. Stopwords were removed, and the Krovetz stemmer was used during indexing.

As mentioned earlier, we collapse all identical mentions of a name together without any coreference resolution, and without even considering the type of the entity. However, we retain some type

[1] We went through this list of patterns that NIST had provided and our answers. Again we found that we hardly returned any answers that were not covered by the original list. For TREC-2003, we were able to complete the verification of answer patterns for only 200 factoid questions. For the remaining 180 we stuck to the regular expressions exactly as provided by NIST. This list did not have any questions with the answer as NIL.

information for each entity model. Each snippet that contributes to a model of a given name is associated with a mention of that name, and that mention has a type assigned to it by IdentiFinder^{TM}. We retain the frequency information of the type of each mention that contributed to the model.

4.4 Retrieval Implementations

We used the LEMUR toolkit [5] to implement all of our retrieval models. The parameters for the query likelihood model and for relevance models were determined by a search on the TREC-2002 question set. We denote the vector space retrieval methods as TFIDF and BM25, the Inquery retrieval runs as INQ, query likelihood by QL, and relevance models by RM.

We used Inquery with the purpose of benefiting from structured queries. For the purposes of this paper we generate structured queries from the questions in the following way. Using the Brill [3] tagger, we automatically extract noun phrases from the question and add them to the original question, enclosed in the #phrase operator.

For example the question *What is the chemical formula of sulphur dioxide?* is transformed to *#sum(#phrase(chemical formula) #phrase(sulphur dioxide));*

For the rest of this paper we denote INQUERY retrieval runs as INQ.

4.5 Question Classification

For Question Classification we used a Support Vector Machine (SVM)[10] based classifier. That Support Vector Machines are known to work well for question classification has been shown by Zhang and Lee [12]. Zhang and Lee used a string kernel. Our classifier is much simpler, and is constructed using SVM Light[1].

Our question classifier categorizes entities into categories as defined by the BBN question ontology. This ontology fits neatly with BBN's IdentiFinder^{TM}as a named entity extraction toolkit. A list of the 31 categories in the BBN ontology is given in Appendix B. The classifier uses unigrams, bigrams and hypernym expansions of the headword via WordNet [2] as features, a simple radial basis function (RBF) kernel and has an accuracy of about 83%. It is trained on labeled TREC-8,9 and 10 question sets provided by BBN.

4.6 Making use of the Entity classes

The ranked list obtained in the above experiments is a ranking over all 2,122,126 entities, and does not take into consideration the class of the question, or the class of the entity as recognized by IdentiFinder ^{TM}. We can improve the ranked list by taking into consideration these two pieces of information.

For each entity in the ranked list we have the frequency of the types it was categorized as. For example, an entity A might have have been tagged 5 times in the corpus as a *person* and 3 times as a *work of art*. This may be representative of the true category. However, one must remember that we may find that different mentions of the same entity occur as different types even for an entity which has only one true category. This is because the named entity tagger can make mistakes. Therefore, for each entity we have a count of the number of times it occurs as any given type. We could make use of these frequencies in different ways. The simplest way to use these counts would be to match the most frequent category of the entity with the type of the question. Results using that approach are denoted as TFIDF′, BM25′, INQ′, LM′ and RM′ in table 1. Note that the list of question classes (Appendix B), and IdentiFinder categories are not exactly the same. Hence we used this matching technique only for questions that were classified into a category that matched an IdentiFinder class. For other questions, like *definition* questions we did not make use of this technique and simply did a ranking over all possible entities as before.

4.7 Processing of NILs

To handle those questions whose correct answer is NIL, we used a threshold on the scores as returned by the information retrieval system, such that only documents with scores above the threshold are considered in the ranking. For a given query if there are no documents above that threshold the system returns a NIL.

Since we have NIL-type questions only for TREC-2002, we did not have a held-out test set of these types of questions. Our results for NIL type questions which we discuss in Section 5 indicate that it is indeed hard to arrive at a threshold to detect such questions.

5. RESULTS

5.1 TREC-2002

Table 1 shows the accuracy at rank one and the mean reciprocal rank computed using the top 10 retrieved entities for each of the retrieval models. We observe that with no language processing, that is, without the use of a question classifier or without the use of the entity classes as provided by IdentiFinder ^{TM}, we obtain a very low accuracy. However, our approach where we match the entity type with the question type almost doubles our performance as shown in figure 1

System	Accuracy(%)	MRR
TFIDF	7.4	0.11
BM25	7.4	0.11
INQ	6.8	0.10
QL	4.8	0.08
RM1	7.2	0.10
TFIDF′	14.6	0.18
BM25′	14.6	0.18
INQ′	11	0.13
QL′	10.2	0.15
RM1′	14.4	0.18

Figure 1: Performance of 5 different retrieval models on the TREC-2002 task. Accuracy is the the number of answers at rank 1 for the 500 training questions. MRR is computed using the top 10 retrieved entities

	TFIDF′	BM25′	INQ′	LM′	RM′
TFIDF′	X	-	+	+	-
BM25′	X	X	+	+	-
INQ′	X	X	X	-	+
LM′	X	X	X	X	+
RM′	X	X	X	X	X

Figure 2: Significance tests, doing pairwise comparisons of retrieval models. + indicates that the Null hypothesis was rejected, X indicates that no test was done, and - indicates that we cannot reject the null hypothesis

We performed a two tailed t test at 95% confidence to compare the performance of TFIDF′,BM25′, INQ′,LM′ and RM′. The results are shown in figure 2. From Figures 1 and 2 we observe that TFIDF′, BM25′ and RM′ perform almost identically.

Qid	Question	Answer	Type
1401	What is the democratic party symbol?	donkey	definition
1403	When was the internal combustion engine invented?	1867	date
1442	What is the chemical formula for sulphur dioxide?	so2	definition
1491	What was the name of Sherlock Holmes' brother?	mycroft	person
1625	What is the deepest lake in the world	baikal	location
1657	What do the French call the English Channel?	la manche	other

Figure 3: Some questions for which we retrieved the answer at rank one and for which others possibly had trouble

The results indicate that the unigram vector space model does much better on average than INQ', the system which uses contextual information. When we look at the accuracy on different types of questions, we see that some methods do better than others on certain categories of questions. For example a unigram approach is significantly better for *person* and *date* type of questions (We did a two tailed t-test at 95% confidence to compare the performance of INQ′ and TFIDF′ on *person* and *date* type of questions). However, contextual information is useful for certain classes of questions-especially *definition* type questions.

This leads us to an approach which makes use of a combination of retrieval mechanisms depending on the class of the question. Based on our results, we choose an approach which uses TFIDF′ (or a unigram entity model) for *person, date, location, fac, gpe* and *location* type questions and INQ′ for all other types of questions. The results for the combined approach are shown in the last column of Figure 8.

If we consider the fact that we are automatically able to classify questions and that we do better on some categories than others, we can argue that our method can be used for only those types of questions for which it works well. From the table in Figure 8 it is obvious that our method works well for *person* and *date* types of questions, obtaining accuracies of 22% and 25% respectively on each of these two types. If we use our system, only for queries classified as *person* and *date*, we can obtain an accuracy of 24.5% on that set of queries, with an MRR score of 0.28.

Our goal is to demonstrate that entity models are a simple and useful technique for this type of question answering, but it is nonetheless instructive to compare its effectiveness with more complex methods. This approach does not result in a high quality question answering system, as can be seen by comparing it to TREC results. The TREC-2002 proceedings tabulate the accuracy of the top 15 systems. The table in Figure 4 shows the performance of the top performing system (LCCmain2002), the system which was ranked 15th that year (pqas22) and a system (BBN2002C) that had an accuracy somewhere in-between the accuracies of the first two. The Appendix of the TREC-2002 proceedings gives the number of questions correctly answered by each system that participated in the track that year . We obtained the accuracy of each system on the main task from there. Figure 5 shows the accuracies of each of the systems sorted in decreasing order of accuracy. Our system, based on Entity Models would be about middle of the order, if rankings were done in the order of accuracy. The median performance is around 0.2, and our system has an accuracy of about 0.16.

System	Accuracy	NIL Precision	NIL Recall
LCCmain2002	0.83	0.578	0.804
BBN2002C	0.284	0.182	0.087
pqas22	0.266	0.145	0.674

Figure 4: Performance of 3 of the top 15 systems at TREC-2002

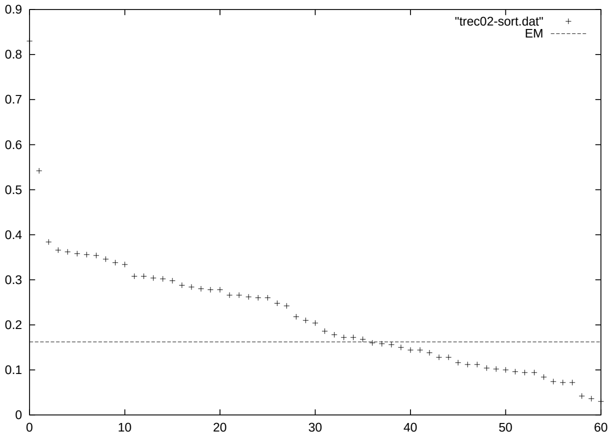

Figure 5: The above graph shows the number of questions answered exactly by the different TREC participants, sorted by decreasing order of accuracy. The horizontal line indicates our systems performance. COMB′ retrieves 81 questions of 500.

We then analyzed the judgments as provided by NIST. The TREC website has a judgment set for each set of questions. The judgment set contains the document and answer string pairs from all submissions to the track that year, for all the questions. It also contains a judgment which is a numerical value indicating whether the answer was acceptable or not. We analyzed the judgment set to find questions for which less than 5 systems had received a correct answer (a judgment of greater than 0), and for which our system had obtained the answer at rank one. There were 16 such questions. A few examples are shown in Figure3.

Consider the question No. 1491, *What was the name of Sherlock Holmes' brother?*. The correct answer is *Mycroft*. The question is classified as a *person* type of question, and we saw that unigram models perform best for this type. The top 5 terms of the entity model of *Mycroft* sorted by their maximum likelihood probabilities are shown below.

Probability	**Word**
0.122807	holmes
0.0877193	mycroft
0.0526316	sherlock
0.0526316	brother
0.0350877	wild

The above values indicate that certain answers can be found by simply considering the statistics of words present in the window around the mention of a word, without the application of any natural language understanding. Definition questions seem to benefit from the use of noun phrases. In an earlier Section we had shown how we constructed structured queries from the original question using the example of question *1442 - What is the chemical formula for sulphur dioxide?*. It is apparent from Figure 5.1 that there are some potential benefits to that approach. The phrase *sulphur dioxide* occurs with high frequency in the pseudo document of *so2*, making it highly ranked for the query *#sum(#phrase(chemical formula) #phrase(sulphur dioxide));*

february 12 xinhua sulphur dioxide so2 emission considerable reduce europe 48 percent decline transboundary flux so2 decade surpass 30 percent program focus controlling sulfur dioxide so2 emission main pollutant atmosphere plan industrial source discharg so2 meet national standard 2000 ensure atmospheric pollution index api sulfur dioxide so2 nitrogen oxide nox total suspend particle tsp atmospheric pollution index index api sulfur dioxide so2 nitrogen oxide nox total suspend particle tsp vapor kilogram sulphur dioxide so2 estimate fuel sale airline sugges nitrogen oxide pollution vehicle concentrate sulphur dioxide so2 carbon dioxide co2 measure flue gas desulfurize equipe offset anticipate so2 project include

Figure 6: Pseudo document for so2

5.2 TREC-2003

We now go on to discuss results that we obtained on the TREC-2003 set of questions. Figure 7 shows how each of the retrieval methods performed on TREC-2003.

System	Accuracy	MRR
TFIDF	4.7	0.08
INQ	4.4	0.08
RM	4.7	0.08
TFIDF′	7.3	0.11
INQ′	7.1	0.10
RM′	7.3	0.11

Figure 7: Performance of 5 different retrieval models on the TREC-2003 task. Accuracy is the the number of answers at rank 1 for 380 factoid questions. MRR is computed using the top 10 retrieved entities

Unigram models continue to outperform INQ′. However, when we group performance on the basis of categories as assigned by our classifier we observe that some of our observations from the previous section still hold. Our result, indicating that the unigram language model performs well for questions that are classified (automatically) as *date, person* or *gpe* type of questions, continues to hold on the TREC-2003 set of questions too. The need for contextual information to obtain answers of questions classified as *definition* type of questions also remains. In the TREC-2003 set of questions we are able to obtain about 20% of the answers at rank one for *person* type of questions. This is consistent with the performance on TREC-2002. However, in TREC-2003 only 9.2% of the questions are classified as *person* type of questions. Hence we do not get an overall improvement in accuracy. *Definition* type questions continue to benefit from contextual information. On the TREC-2003 set of questions we are able to obtain answers for nearly 10% of the definition questions at rank 1 using COMB′.

We analyzed the accuracies for TREC-2003 systems just as we did for TREC-2002. The median performance is around 0.17 for a value of accuracy (the proportion of the answers answered correctly). Our system achieves around 0.11 accuracy.

In the previous section we proposed that we could use our Entity Model methodology for only those queries that were classified as *person* and *date* by the question classifier. If we followed that approach one would obtain an accuracy of 11% on the set of questions classified as *date* or *person* and an MRR score of 0.16.

5.3 NIL accuracy

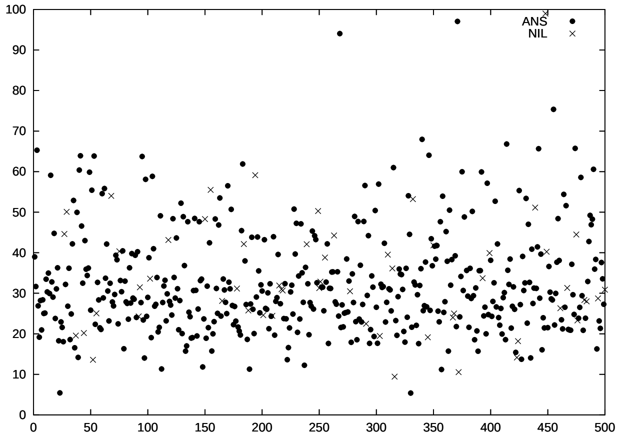

Figure 10: Distribution of scores for NIL and ANS questions using TFIDF retrieval

Fgure 10 shows the distribution of scores for two categories of questions– those whose answers were present in the database (ANS), and those whose answers were not (NIL). There is no clear threshold on the scores which would allow one to discriminate between the two classes. Therefore, we report NIL only in the case that not a single document was retrieved by our system. This is equivalent to setting a threshold of zero.

Using the above approach, TFIDF′ and RM′ did not report any answers as NIL. Hence they had a precision and recall of zero on questions whose correct answer is NIL. However, INQ′ and COMB′ reported 6 and 5 questions respectively as questions having the answer NIL. All 6 questions were correctly answered as NIL by INQ′, giving it a NIL recall of 10.52% and a Precision of 100%. COMB′ had similar values at 9.6% recall and 100% precision.

From the TREC-2002 proceedings and from figure 4 we ascertained that even the best systems do badly on NIL accuracy, indicating that it is indeed difficult to predict when an answer is not found in the database. Systems that achieved high NIL precision had poor recall, and those that had high recall had very low precision.

6. CONCLUSION AND FUTURE WORK

We have shown that entity models can often be used to answer questions. Using a very simple model of entities—the text surrounding mentions of the entity as found by a named entity recognizer—we have answered around a quarter of some classes of questions accurately. We are confident that improvements in snippet selection, disambiguation of frequently occurring entities, entity reference resolution, and tagging of a larger range of entity types, could all contribute to improved performance on the question answering task. We do not expect that entity models could be vastly more successful or that they could achieve the accuracy of more knowledge- and processing-intensive approaches.

However, our primary goal was to demonstrate that entity models are a useful representation, that they do indeed capture useful information about the entity. We believe that we have successfully demonstrated that and—in addition to trying to make them more

Question class	%	Accuracy at Rank 1			
	of Total	INQ	TFIDF	RM	COMBIN
animal	1.4	0(.1)	0(0.2)	0(0.07)	0(0.1)
bio	0.2	1(1)	1(1)	1(1)	1(1)
cardinal	1.6	0(.05)	0(0.06)	0(0.05)	0(0.05)
date	21.8	0.16(0.22)	0.23(0.29)+	0.25(0.31)	0.31(0.25)
definition	11.6	0.12(0.14)	0.05(0.08)	0.03(0.06)	0.12(0.14)
fac	1.8	0(0.04)	0.55(0.57)	0.22(0.3)	0.22(0.3)
gpe	19.2	0.02(0.06)	0.08(0.12)+	0.06(0.11)	0.06(0.11)
language	0.4	0.5(0.5)	0.5(0.5)	0.5(0.5)	0.5(0.5)
location	4.4	0.09(0.14)	0.09(0.13)	0.13(0.16)	0.13(0.16)
money	1.4	0.28(0.35)	0.14(0.14)	0.14(0.14)	0.28(0.35)
nationality	0.2	0(0.16)	0(0)	0(0)	0(0.16)
organization	3.2	0.06(0.09)	0.12(0.15)	0.12(0.19)	0.12(0.19)
other	4.4	0.13(0.14)	0.04(0.06)	0.04(0.02)	0.13(0.14)
percent	0.6	0(0)	0(0)	0(0)	0(0)
person	18.6	0.13(0.17)	0.20(0.23)+	0.22(0.25)	0.22(0.25)
plant	0.2	0(0.11)	0(0.5)	0(0.2)	0(0.11)
product	0.6	0.33(0.33)	0.33(0.33)	0(0.16)	0.33(0.33)
quantity	9	0.11(0.12)	0.07(0.09)	0.08(0.09)	0.11(0.12)
substance	0.4	0(0.166)	0.5(0.75)	0.5(0.55)	0(0.166)
work-of-art	1	0(0)	0(0)	0(0)	0(0)
Total	100	0.11(0.12)	0.14(0.18)	0.14(0.18)	0.16(0.20)

Figure 8: Accuracy by category on the TREC-2002 questions. Plus signs indicate statistical significance. Numbers in brackets are the MRR scores for each category

Question class	Total	Accuracy at Rank 1			
		INQ′	TFIDF′	RM′	COMBIN
animal	2.1	0.125(0.142)	0.125(0.156)	0(0.11)	0.125(0.142)
cardinal	11.8	0.02(0.045)	0.022(0.055)	0.04(0.09)	0.02(0.045)
cause-effect-influence	2.6	0(0.25)	0(0)	0(0)	0(0.25)
contact-info	2.6	0(0.25)	0(0)	0(0.2)	0(0.25)
date	13.9	0.05(0.11)	0.11(0.14)	0.10(0.15)	0.10(0.15)
definition	12.9	0.10(0.12)	0.06(0.09)	0.04(0.06)	0.10(0.12)
fac	2.4	0.11(0.17)	0(0.17)	0.2(0.30)	0.2(0.30)
gpe	13.4	0.06(0.08)	0.05(0.09)	0.07(0.10)	0.07(0.10)
language	2.6	0(0)	0(0)	0(0)	0(0)
location	3.1	0.08(0.14)	0.25(0.27)	0.25(0.3)	0.25(0.3)
money	7.8	0(0)	0(0)	0(0)	0(0)
organization	4.7	0.05(0.09)	0.11(0.17)	0.11(0.20)	0.05(0.09)
other	11.6	0(0.016)	0(0.01)	0(0.018)	0(0.016)
percent	2.6	0.5(0.5)	0(0.08)	0(0)	0.5 (0.5)
person	9.2	0.03(0.20)	0.14(0.20)	0.14(0.20)	.14(0.20)
quantity	11	0.07(0.09)	0.04(0.06)	0.02(0.04)	0.07(0.09)
substance	2.6	0(0)	0(0)	0(0)	0(0)
use	2.6	0(0.125)	0(0.14)	0(0)	0(0.125)
work-of-art	1.3	0.4(0.5)	0.4(0.46)	0.4(0.425)	0.4(0.5)
TOTAL	100	.071(0.10)	0.073(0.11)	0.073(0.11)	0.09(0.13)

Figure 9: Accuracy by category on the TREC-2003 questions. Numbers in brackets are the MRR score for each category.

accurate—hope to move into new applications of entity models. For example, a comparison of entity models can provide probabilistic links between them that can then be incorporated into appropriate data mining activities [16, 14]. We mentioned *use theory* in the Introduction. Use Theory states that if two words are used in similar contexts they probably are related. In the same way if *tennis* appears often in the context of *Monica Seles* and *Pete Sampras*, we could learn that they were related, and the word *tennis* in some way describes the similarity between them.

The broad focus of our continuing work is to improve the quality of entity models and to explore their broader utility. On the first point, we expect that sentence parsing may help select which text is most descriptive of an entity and that phrases may provide more focused descriptions. We are interested in finding a way to break very frequently occurring entities (e.g., the name of the US president) into ”aspects” or ”topics” so that the entity is not washed out by so many concepts.For example *Arnold Schwarzenegger* would clearly have two aspects *Movies* and *Politics*. Another area for research is the construction of good entity models. Determining which phrases should contribute to the model, getting rid of repitions, dealing with document length problems in the pseudo document database, and constructing entity models with interesting statistical properties for the task at hand is an open research problem.

We believe that entity models provide an intriguing alternate viewpoint of a collection. We have shown that they have potential for a task such as question answering and expect they will be more broadly useful. We are exploring additional ways that entity models can be used, including connecting them into data mining systems, incorporating them into news tracking systems, and leveraging them for summarization of the personalities involved in a story.

APPENDIX

A. IDENTIFINDER ™CATEGORIES

ANIMAL CONTACT INFO DISEASE
EVENT FAC GAME
GPE LANGUAGE LAW
LOCATION NATIONALITY ORGANIZATION
PERSON PLANT PRODUCT
SUBSTANCE WORK OF ART CARDINAL
MONEY ORDINAL PERCENT
QUANTITY DATE TIME

B. QUESTION CLASSES

ANIMAL BIO CARDINAL
CAUSE-EFFECT CONTACT-INFO DATE
DEFINTION DISEASE EVENT
FAC FAC_DESC GAME
GPE LANGUAGE LOCATION
MONEY NATIONALITY ORGANIZATION
ORG_DESC OTHER PERCENT
PERSON PLANT PRODUCT
PRODUCT-DESC QUANTITY REASON
SUBSTANCE TIME USE

C. ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903 and in part by Advanced Research and Development Activity under contract number MDA904-01-C-0984. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

D. REFERENCES

[1] http://svmlight.joachims.org/.

[2] http://www.cogsci.princeton.edu/ wn/.

[3] http://www.cs.jhu.edu/ brill/.

[4] http://www.trec.nist.gov.

[5] The lemur toolkit, http://www.cs.cmu.edu/lemur.

[6] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[7] J. P. Callan, W. B. Croft, and S. M. Harding. The INQUERY retrieval system. In *Proceedings of DEXA-92, 3rd International Conference on Database and Expert Systems Applications*, pages 78–83, 1992.

[8] K. Church, W.Gale, P.Hanks, and D.Hindle. Using statistics in lexical analysis. In *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pages 115–164, 1991.

[9] J. G. Conrad and M. H. Utt. A system for discovering relationships by feature extraction from text databases. In *Proc. of the 17th ACM-SIGIR Conference*, pages 260–270, 1994.

[10] N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, 2000.

[11] S. H. Dan. Falcon: Boosting knowledge for answer engines.

[12] W. S. L. Dell Zhang. Question classification using support vector machines. In *Proc. of the 26th ACM-SIGIR Conference*, pages 260–270, 2003.

[13] S. Haas and R. Losee Jr. Looking in text windows: Their size and composition. *Information Processing and Management*, 50:619–629, 1994.

[14] M. Hearst. Untangling text data mining. In *Proc. of ACL*, 1999.

[15] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Research and Development in Information Retrieval*, pages 120–127, 2001.

[16] U. Y. Nahm and R. J. Mooney. Mining soft-matching rules from textual data. In *IJCAI*, pages 979–986, 2001.

[17] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

[18] J. M. Prager, J. Chu-Carroll, and K. Czuba. Use of wordnet hypernyms for answering what-is questions. In *Text REtrieval Conference*, 2001.

[19] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-2. In *The Second Text retrieval Conference*, pages 21–34, 1994.

[20] S. E. Robertson. The probability ranking principle in ir. pages 281–286, 1997.

[21] In *The Eighth Text REtrieval Conference (TREC 8)*. NIST, 1999. NIST Special Publication 500-246.

[22] In *The Eleventh Text REtrieval Conference (TREC 11)*. NIST, 2002. NIST Special Publication 500-246.

[23] In *The Twelfth Text REtrieval Conference (TREC 12)*. NIST, 2003. NIST Special Publication 500-246.

[24] L. Wittgenstein. *Philosophical Investigations*. Basil Blackwell and Mott Ltd., Oxford, England, 1953.

[25] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, 2000.