In *Proceedings of the 13th Conference on Information and Knowledge Management (CIKM 2004), Nov 8-13, Washington DC*

Unsupervised Question Answering Data Acquisition From Local Corpora

Lucian Vlad Lita

Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
llita@cs.cmu.edu

Jaime Carbonell

Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
jgc@cs.cmu.edu

ABSTRACT

Data-driven approaches in question answering (QA) are increasingly common. Since availability of training data for such approaches is very limited, we propose an unsupervised algorithm that generates high quality question-answer pairs from local corpora. The algorithm is ontology independent, requiring very small seed data as its starting point. Two alternating views of the data make learning possible: 1) question types are viewed as relations between entities and 2) question types are described by their corresponding question-answer pairs. These two aspects of the data allow us to construct an unsupervised algorithm that acquires high precision question-answer pairs. We show the quality of the acquired data for different question types and perform a task-based evaluation. With each iteration, pairs acquired by the unsupervised algorithm are used as training data to a simple QA system. Performance increases with the number of question-answer pairs acquired confirming the robustness of the unsupervised algorithm. We introduce the notion of *semantic drift* and show that it is a desirable quality in training data for question answering systems.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and SoftwareQuestion Answering (fact retrieval) systems

General Terms

Algorithms, Experimentation

Keywords

data acquisition, question answering, unsupervised learning, semantic drift

CIKM'04, November 8–13, 2004, Washington, DC, USA.

1. INTRODUCTION

Over the past few years, question answering (QA) has evolved from successful pipeline architectures [16, 9, 2], to systems that incorporate complex reasoning mechanisms and planning [15, 17] as well as large amounts of training data [7, 6]. Data driven approaches to open domain question answering are becoming increasingly common [11, 1, 19, 5]. Statistical, as well as NLP systems heavily rely on the availability of large corpora [6], knowledge resources [8, 10, 12], and training data in the form of questions and corresponding answers.

An important issue in question answering is the limited amount of high quality training data. Standard question-answer datasets emerged through past TREC [21] and CLEF [13] competitions. However, the size of these datasets is still very small, bringing the number of overall question-answer pairs to under five thousand. Less standard datasets, such as trivia question databases are typically too specific in terms of content or format in order to be used as training data for a QA system.

Considerable effort has been put into building ontologies for question answering – originally created to support a limited set of factoid questions. However these ontologies are not standard across systems. They are also being continuously extended to cover increasingly many types of questions from different domains. Existing question-answer pairs cannot reasonably cover most question ontologies available today. For statistical learning, it is often necessary for each category/type in an ontology to have a minimum number of question instances associated with it, in order to be able to derive a corresponding model. Larger datasets are required for data-driven systems to be able to accurately make use of these ontologies.

We observe that manual acquisition of question-answer pairs is very expensive and highly subjective. However it is imperative to obtain large amounts of training data for data-driven methods. To overcome this problem, we propose an unsupervised algorithm for high precision question-answer data acquisition from local corpora. The algorithm requires a very small seed data and is compatible with existing question ontologies – i.e. makes no assumptions about question types. It is resource independent and does not assume the availability of specific pre-processing tools. The approach is also language independent and can be applied to any cor-

pus given an appropriate seed/starting point. We acquire a large set of questions and answers, and we demonstrate their quality through task-based evaluation. We show that even one of the simplest data driven extraction methods can obtain results comparable to the top five performing systems at TREC.

We introduce the notion of *semantic drift* and argue that it is a desirable quality when building question answering systems.

2. RELATED WORK

Recent research has been focusing on two dimensions of data acquisition: acquiring redundant passages to support existing questions and acquiring supporting data for answering new questions.

Gathering redundant passages is likely to boost the confidence of correct answers: Dumais et all [4] make use of the high redundancy of the web and retrieve passages presumed to contain a correct answer. This work supports the intuitive argument that more relevant passages entail higher QA performance. The approach is based on the assumptions that most questions have answers on the web and that the most frequent answer is the correct answer. However, it is less appropriate for questions with sparse supporting web data, multiple meanings, or based on subjective assertions. Furthermore, learning extraction models solely from web data [18] is likely to saturate soon after acquiring very simple but highly redundant patterns/features.

The second dimension consists of acquiring data to support answering new questions. Girju et all [7] propose a supervised algorithm for part-whole relations based on 20,000 manually inspected sentences and on 53,944 manually annotated relations. They report an $F1$ measure of about 90 in answering questions based on part-whole relations. Fleischman et all [6] also propose a supervised algorithm that uses part of speech patterns and a large corpus. The algorithm extracts semantic relations for *Who-is* type questions and builds an offline question-answer database.

Unsupervised models in named entity tagging are closely related to our approach. Collins et all [3] thoroughly explore unsupervised models ranging from a decision list approach [22] to an expectation maximization approach and good results are presented in terms of precision. However, the issue of recall in unsupervised named entity tagging needs to be further investigated.

In recent years, learning components have started to be more frequent in question answering [1, 19, 5]. Although the field is still dominated by knowledge-intensive approaches, components such as question classification, answer extraction, and answer verification are beginning to be addressed through statistical methods. Moreover, current research [11] shows that it is possible to successfully learn answering strategies directly from question-answer pairs through an instance based approach.

3. APPROACH

In this paper we view questions as collections of entities and relations among them. The missing piece of information – the required answer – is usually in the form of an unknown relation or an unknown entity. Consider the following examples:

A *invented* **Q**
A *is a part of* **Q**
Q *is in* **A**

These statements contain the entities **Q** and **A**, as well as relations between them. In contrast, questions usually consist of incomplete collections of relations and entities. The answering process involves finding the missing element. Some questions may contain all the entities but lack the relation itself:

- What is the connection between **A** and **Q**?
- How are **A** and **Q** related?

while other questions might contain the relation and lack one of the entities involved:

- Who invented **Q**?
- What does **Q** contain?
- Where is **Q**?

where **Q** denotes the entity present in the question and **A** denotes the required answer. We will focus on questions whose answers are missing entities. Relations will also be referred to as **question types** (e.g. *who-invented*, *where-is*), since they usually determine specific answer seeking strategies in most question answering systems.

QA ontologies often include question types as well as answer types. We stress the distinction between answer type and question type: different question types (e.g. *who-invented*, *who-is-the-leader-of*, *who-controls*) may produce answers of the same type (e.g. *person*). For simplicity many existing ontologies often consider question types as specializations of answer types: *leader-of* would be a specialization or refinement of answer type *person*.

The approach presented in this paper is independent of specific ontologies since we adopt the view that a question type can be directly described through the data: question-answer pairs. For each question type, question-answer pairs (**Q**,**A**) that fit the relation are acquired from the local corpus. Given enough high-quality question-answer pairs, a QA system can be trained to answer similar questions.

Many question answering systems use questions and answers as training data in order to construct or improve their answer seeking strategies. In this paper, we focus on the process of acquiring high quality question-answer pairs rather than arguing how to incorporate them into a specific QA system.

3.1 The Unsupervised Algorithm

A relation can be defined as a set of high precision context patterns. The patterns are in fact alternate forms of expressing a concept – for example the relation *who-hired* may occur in raw text as "Y *was hired by* X", "Y *is employed by* X" etc

The same relation can also be defined indirectly through a set of entity pairs. Each pair is an instance of that particular relation. Sample instances of relation *who-wrote* are: (*Hesse*, *The Glass Bead Game*) and (*Jefferson*, *The Declaration of Independence*). Since the relations correspond to question types, the entity pairs can be viewed as question-answer pairs:

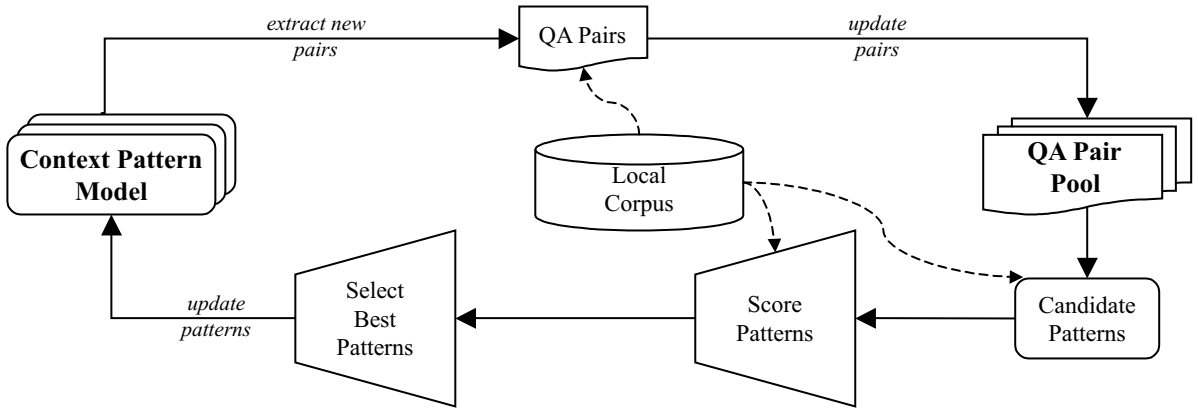

Figure 1: Unsupervised QA data acquisition. During each iteration, question-answer pairs of the same type are used to extract highly correlated context patterns. In turn, the patterns are used to generate more question-answer pairs.

- Who wrote "The Glass Bead Game"?
- Who wrote the Declaration of Independence?

We present an unsupervised algorithm (figure 1) that iterates trough these two alternate views of relations: a set of patterns (the *Context Pattern Model*) and a set of question-answer pairs (the *QA Pair Pool*). The algorithm acquires question-answer pairs while at the same time improving the high precision pattern set.

1. Start with a seed of context patterns $\{T\}$ or question-answer pairs $\{(Q, A)\}$
2. Apply the context patterns $\{T\}$ and extract question-answer pairs $\{(Q, A)\}'$ from the local corpus
3. Using the local corpus, extract a set of candidate context patterns $\{T\}'$ that co-occur with $\{(Q, A)\}'$
4. Score the candidate contexts $\{T\}'$ according to a conservative relevance criterion.
5. Select the top K candidate contexts $\{T\}''$
6. Update the model $\{T\}$ with selected contexts $\{T\}''$
7. Return to step 1

3.2 Selection Criterion

Each iteration, the Context Pattern Model is updated to contain a subset of the candidate context patterns that have the best scores. Scoring must be based on a criterion that maximizes the correlation of a pattern with the existing question-answer instances in the QA Pair Pool.

The selection criterion used in this paper is the **F1** measure of a pattern T at iteration i. For clarity, we consider the precision and recall of pattern T – which can be thought of as as a query in the local corpus – relative to the known "correct" pair set, QA Pair Pool. Given the QA Pair Pool $\{(Q, A)\}$ during the i^{th} iteration, a candidate context pattern T has a precision and recall:

$$R(T,i) = \frac{PoolCoverage(T,i)}{|Pool(i)|}$$

$$P(T,i) = \frac{PoolCoverage(T,i)}{CorpusCoverage(T)}$$

where $PoolCoverage(T,i)$ is the number of pairs known to be "correct" (i.e. extracted so far and stored in the QA Pair Pool) that were extracted using pattern T as a query in the corpus at iteration i. $CorpusCov(T)$ represents the number of distinct pairs that pattern T can extract from the corpus at iteration i, and $|Pool(i)|$ is the size of the QA Pair Pool at iteration i.

The **F1** measure based on pool coverage and corpus coverage is:

$$F1(T,i) = \frac{2 \cdot P(T,i) \cdot R(T,i)}{P(T,i) + R(T,i)}$$

At iteration $i+1$, we select the K candidate patterns with highest top F1 score and use them to update the Context Pattern Model.

In order to intuitively illustrate corpus coverage and pool coverage, consider the question type *who-invented*. The goal of the unsupervised algorithm is to extract as many pairs of inventors and objects invented as possible. The algorithm is considering whether to include the pattern "*A, father of Q*" into the Context Pattern Model. The pattern can be used to extract relevant pairs such as *(Farnsworth, television)*, but also noisy pairs such as *(Michael, John)*. The recall is high is high since many inventors are referred to as parents of their own inventions and consequently this pattern can extract many known pairs inventors-inventions from the corpus: i.e. pairs already in the QA Pair Pool have a high correlation with this pattern. However, the precision is low since the pattern occurs very frequently in the local corpus. As shown in this example, the pattern "*A, father of Q*" is often a manifestation of other relations beside *who-invented*. The corpus coverage of our pattern is high, but only a very small percentage of pair instances actually refer to the inventor-invention relation.

We have explored other selection criteria based on *pool coverage*. These criteria are faster to compute, but very of-

ten the algorithm diverges quickly from the original question type. One particular criterion that yields results similar to the F1 measure has been successfully used in semantic lexicon extraction [20]:

$$score_p(T,i) = \frac{PoolCoverage(T,i)}{CorpusCoverage(T,i)} \cdot \log PoolCoverage(T,i)$$

Intuitively, pattern T obtains a high score if a high percentage of the pairs it extracts are already in the QA Pair Pool, or if it extracts a moderate number of pairs already in the QA Pair Pool and it extracts lots of them.

3.3 Starting and Stopping Criteria

The algorithm can be initialized either with a small set of patterns in the Context Pattern Model or a set of questions-answer pairs of the same type in the QA Pair Pool. The former approach can be better controlled and has the potential of being more precise, while the later approach can be automated more easily.

A moderate-size validation dataset could be used as the stopping criterion for the algorithm, determining when the question-answer pairs are becoming detrimental as training data to a QA system. When the question-answer pairs extracted are completely deviating from the original relation expressed in the seed, they will most likely not improve the performance of a question answering system, since there is nothing new to be learned. The advantage of a validation set is that the acquisition of question-answer pairs based on different relations will have flexible stopping criteria and the process can be tailored for specific QA systems, rather than imposing a threshold on learning saturation. The disadvantage consists in the fact that standard QA datasets contain very few questions and cover a limited number of question types.

Since using a reasonable-size validation set is not yet feasible, a set of parameters in the unsupervised algorithm can be learned in order to control how much questions deviate from the original relation. The set of parameters can consist of number of iterations, number of extracted pairs, or a threshold on pattern extraction precision.

4. SEMANTIC DRIFT

Often times questions are either ambiguous or are formulated awkwardly. For example, the question "*Who invented The Muppets?*" is conceptually equivalent to the question "*Who is the creator of The Muppets?*". The latter formulation is more frequently observed than the former when expressing the connection between Jim Henson and The Muppets. Intuitively, this shows that multiple relations may belong to a larger semantic class.

Unsupervised algorithms generally experience a degradation in the quality of the data/model over time. Traditionally this is viewed as a negative phenomenon, since it introduces noise. This degradation also varies with the seed data and the corpus being used and is not easily controlled. This phenomenon also occurs in the unsupervised question-answer pair acquisition algorithm. In practice, conservatively incorporating this noise into the answer extraction model increases the performance.

The very nature of the algorithm dictates that new context patterns will enhance the model after each iteration. We tend to think of these patterns as semantically equivalent. However, in time they tend to diverge from the original relation. We will refer to this unsupervised algorithm inherent property as **semantic drift**. This property reflects a tradeoff between enriching the original semantic relation and noise in the acquired question-answer pairs.

In our previous example, the answer model starts with the notion of *invention*, accumulating context patterns and question-answer pairs that support the original relation. However, through several iterations, the following context patterns are noticed[1]:

$$< inventor\ of > \longrightarrow$$
$$< creator\ of > \longrightarrow$$
$$< producer\ of > \longrightarrow$$
$$< father\ of > \longrightarrow$$
$$< maker\ of >$$

While the notions of *creator-of* and *producer-of* could be considered similar to the original relation (*inventor-of*), the subsequent divergence is too generic to produce relevant question-answer pairs.

Similarly, the relation *winner-of* drifts into context patterns referring to people who are *expected to win* and have not won yet, while the relation *writer-of* drifts to include patterns about publishers, editors and literary works: (i.e. *A , whose novel Q*).

Ideally, semantic drift can be used to add slightly divergent question-answer pairs to the question pool. However, a critical aspect is the stopping criterion, necessary for deciding when data becomes too noisy to be added to the model. As previously mentioned, a moderate-size validation dataset or a set of learned parameters correlated with noise can be used as the stopping criterion for the algorithm, finding a balance between semantic drift and noise.

5. EXPERIMENTS

From the TREC 9, 10, and 11 collections, we identified a set of 90 non-definition questions of the form:

$$Who \quad < verb > \quad Q\ ?$$

We consider each $< verb >$ to represent a specific relation. For each relation described by individual verbs, we automatically employ the unsupervised algorithm in order to generate training question-answer pairs. None of the original TREC question instances were used in the unsupervised learning part of the experiments.

Most systems employ ontologies and group semantically equivalent questions. However, these ontologies are not standard across QA systems. In order to maintain generalizability, the experiments presented in this paper deliberately do not benefit from semantic correlation between questions. Each verb is treated as a semantic relation by itself. For example the question types *who-invented* and *who-created* are viewed as two different relations. However, correlation between question types can be observed at each iteration through the overlap of question-answer pairs.

The unsupervised experiments are based on the TREC and AQUAINT corpora which consist of several gigabytes of text. No language processing tools or external resources

[1]we ignore similar intermediate patterns such as $<$*the person who invented* $>$ for the purpose of clarity

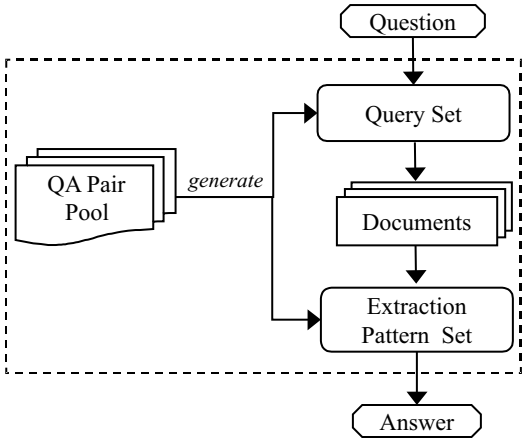

Figure 2: The question answer pairs are used to train a bare-bones question answering system, based mostly on answer extraction.

such as WordNet [14], named entity taggers, part of speech taggers, parsers, gazetteers were used.

The context patterns were limited to a maximum size of a sentence. The starting data for the unsupervised algorithm consists of only one context pattern for each relation:

$$... \quad A \quad <, \ who\ verb > \quad Q \quad ...$$

where A and Q are placeholders for the answer and question terms and *verb* is the verb used to generate the question type. The seed data is extremely simple, but powerful enough that it avoids the human effort that could be put in creating complex and highly precise seeds for each relation. Note that although the seed pattern imposes an ordering on A and Q, the unsupervised algorithm is free from such constraints.

The learned patterns identify exact answers (i.e. proper names). Text snippets which do not have the correct extent – as defined by answer patterns provided by NIST – are considered incorrect answers.

The algorithm was run for each relation, producing up to 2,000 question-answer pairs per question type. For more obscure relations such as *who-found*, the algorithm acquired fewer pairs than for more common relations such as *who-made*.

5.1 Qualitative Analysis

Table 1 shows qualitative results produced by the unsupervised algorithm. Five sample relations are presented with question-answer pair sampling at 1, 10, 100, and 1000 as more data was added to the pool. The specificity varies from very exact questions pairs such as "*Who owns the New Jersey Devils?*" to broader questions more likely to have many correct answers – i.e. "*Who makes small motors?*". In order to show the semantic similarity between two question types as seen through the data, we included both the *invented* and *created* relations.

5.2 Task-Based Evaluation

In order to test the quality of the acquired question-answer pairs, we use them in order to train a baseline QA system with a very simple extraction step. The goal is to indirectly show that an unsupervised approach can acquire question answering data useful in the QA process. We show that performance improves with the number of question-answer pairs acquired and we argue that full-fledged QA systems that employ parsing and named entity tagging can better exploit such training data.

Pair#	Question Term	Answer
	who-invented	
1	dynamite	Alfred Nobel
10	theosophy	Helena Blavatsky
100	dialectical philosophy	Hegel
1,000	television's Twin Peaks	Mark Frost
	who-created	
1	Providence's Waterfire	Barnaby Evans
10	Howdy Doody	Buffalo Bob Smith
100	HBO's acclaimed Mr. Show	Troy Miller
1,000	the invisible holster	Charlie Parrot
	who-makes	
1	small motors	Johnson Electric Holdings
10	ping golf clubs	Karsten Manufacturing corp.
100	removable media data storage devices	Iomega corp.
1,000	all the airbus wings	British Aerospace
	who-owns	
1	The Candlelight Wedding Chapel	Gordon Gust
10	The New Jersey Devils	John Mcmullen
100	the sky diving operation	Steve Stewart
1,000	the ambulance company	Steve Zakheim
	who-founded	
1	Associated Publishers inc.	Mr. Cox
10	Earthlink Network	Sky Dayton
100	Limp Bizkit's label	Jordan Schur
1,000	Macromedia	Marc Canter

Table 1: Sample qualitative results. Question-answer pairs are added to the pool incrementally. We show the 1^{st}, 10^{th}, 100^{th}, $1,000^{th}$ question-answer pairs as they are added to the pool.

The bare-bones QA system (figure 2) consists of a retrieval step and an answer extraction step, where the answers are produced and scored. No query expansion or answer clustering is performed and no feedback loops are present. The retrieval step consists of retrieving raw documents from the web using the Google API (www.google.com/api) and only the keywords present in the question. No outside resource or processing tool was used.

The retrieval step is trained using the high-precision patterns acquired at each iterations during the unsupervised learning. The patterns are added as phrases to queries in order to capture more relevant documents. When new questions are processed, several queries are formed by concatenating the question terms and the high-precision patterns.

These queries are then used to retrieve the top fifty relevant documents.

We did not want to limit the actual answer extraction to the high precision patterns discovered during the unsupervised learning. Although very precise, the recall of this set of patterns would have been too low. Therefor, the answer extraction step is trained by extracting a large number of surface patterns (over 5,000) from the local corpus using the question-answer pairs. These patterns range from highly correlated to the question type to weakly correlated. The patterns are further generalized through regular expressions using elliptical terms.

Each pattern's $F1$ score was computed against the question-answer pairs extracted from the local corpora. When a new question is processed, all generalized patterns are applied to the raw documents. Among the ones that do match, the highest scoring patterns are used to extract the final answer set. A more complex answer clustering and merging method is likely to increase QA performance.

The experiments were evaluated using the following metrics:

1. Mean Reciprocal Rank (MRR) – the average reciprocal rank of the first correct answer for each question. Only the top 5 answers are considered for each question.

$$MRR = \frac{1}{N} \cdot \sum_{i=1}^{N} 1/correct\ answer\ rank_i$$

2. Confidence Weighted Score (CWS), which is a measure combining the percent of correct answers and the confidence of the system in its scoring method. Questions are ordered according to confidence in their corresponding answers.

$$CWS = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{\#\ correct\ up\ to\ question\ i}{i}$$

The candidate answers for the TREC questions were evaluated using the standard answer keys associated with the questions. Although they cover many correct answers, they are by far not complete and do not allow for some variation in the answers. For example if a system produces "*Apple* or "*Apple Computer*" as answers to the question "*Who created the Macintosh computer?*", they are considered wrong answers since they do not match the standard answer keys.

The overall MRR score obtained for the TREC test data is 0.54 and the confidence weighted score is 0.73. Figure 3 shows the overall rank distribution of first correct answers. On this data, the top five performing systems at TREC obtained scores ranging between 0.4 MRR and 0.76 MRR. Figure 4 compares the performance of our straight-forward system (referred to as *QA Pairs*) with the performance of the top five systems at TREC 9, 10 and 11 on the same test data. Note that different systems obtained the top five results in different years. The results are impressive especially when taking into account that the top five systems are full-fledged QA systems incorporating knowledge resources, specialized document retrieval, complex question and passage processing, answer selection and verification. In contrast we focused on a simple answer extraction component of a QA system in order to show the high potential of using more question-answer pairs in training QA systems.

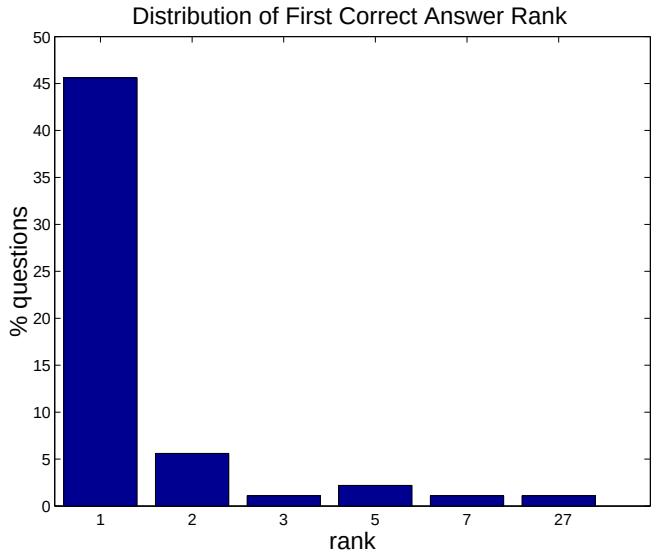

Figure 3: High precision: most correct answers proposed by the system have rank 1.

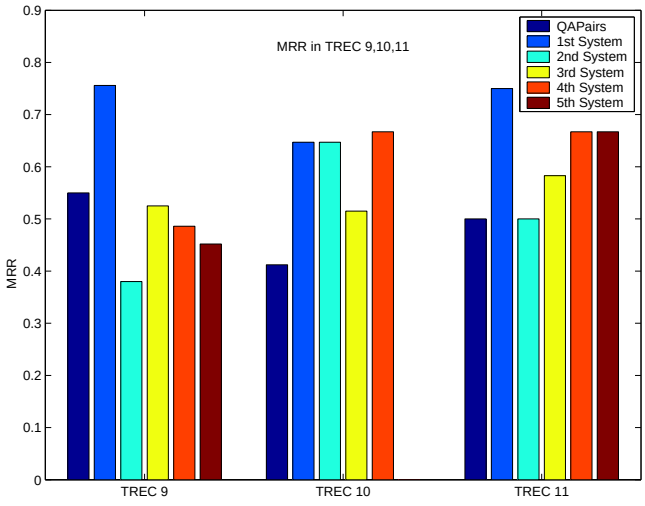

Figure 4: *QA Pairs* compared to the top five system performance at TREC 9, 10 and 11 on the *same* test data used in our experiments.

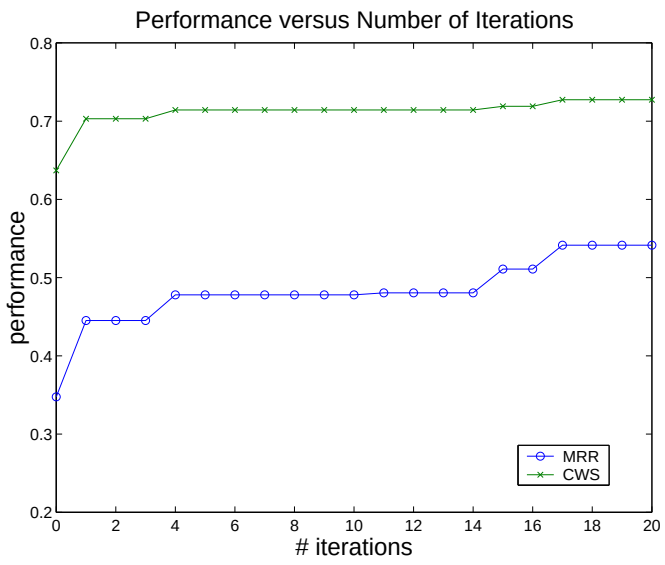

Figure 5: Performance increases with the number of iterations and therefore with the size of the training data.

With each iteration, the unsupervised algorithm acquires more question-answer pairs. At each iteration the bare-bones QA system is re-trained and evaluated. Figure 5 shows that performance improves with each iteration. Semantic drift allows advanced iterations to contribute to the QA system by answering ambiguous questions and capturing answers which are awkwardly stated. However, as more question-answer pairs are added to the pool, they become more obscure and contribute less to learning new patterns.

The fact that performance increases with the acquisition of more question-answer pairs shows that the scoring method is correlated with the number of iterations. The more training data is obtained from the local corpus, the better the answer extraction component performs. This observation further suggests that more complex QA systems can take better advantage of the acquired data.

6. CONCLUSIONS

This paper presents a generic, resource-free, unsupervised algorithm that acquires high quality question-answer pairs from unlabeled data in local corpora. The unsupervised algorithm learns from very small seeds and is compatible with existing question ontologies. The approach is easy to implement and adapt to specific systems.

The acquired question-answer pairs are used in a task-based evaluation to train a very simple data-driven question answering system. Results show that the QA system performance improves with number of training pairs. Moreover, the confidence weighted score also increases with the number of iterations, which indicate that the confidence scores are correlated with answer correctness. These experiments show that the unsupervised algorithm produces high quality training data that can be used by data-driven systems in order to improve performance.

We introduced the notion of *semantic drift* as a desirable property of the unsupervised algorithm. QA systems that use the question-answer pairs will automatically benefit from semantic drift since with each iteration the algorithm diverges slightly from the original relation. The key issue in taking advantage of semantic drift is specifying a good stopping criterion. Because of semantic drift, a question answering system has a higher chance of correctly answering more ambiguous questions than if it is trained on perfect data.

7. FUTURE WORK

Further experiments will selectively incorporate external resources. More accurate noun phrase identification tools can produce better filtering for question-answer pairs. Also, as seen in [6] part of speech can be very helpful in extending the high precision model to include more than context patterns. Semantic analysis as well as phrase expansion may directly contribute to gathering more relevant pairs. Knowledge resources can serve as heuristics and guide semantic drift to quickly cover more relevant variations of the original relation.

The use of specific question ontologies allows the customization of seed data. It also supports a small level of relation clustering in order to obtain more correlated question-answer pairs from the *local* corpus. The use of answer type ontologies is likely to join the seeds for different relations as well as make possible reasonable-size validation sets.

Future work will focus on developing statistical, data-driven answer extraction models that exploit complex relations and better generalize based on semantic drift. We plan to analyze the confidence of automatic seed generation from variable size question datasets. We also plan to experiment with training a fully data-driven instance-based question answering system system [11] with data acquired through unsupervised means from local corpora.

8. ACKNOWLEDGEMENTS

Research presented in this paper has been supported in part by an ARDA grant under Phase I of the AQUAINT program.

9. REFERENCES

[1] C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P. Tilker. Statistical selection of exact answers. *Text Retrieval Conference (TREC)*, 2003.

[2] C. Clarke, G. Cormack, and T. Lynam. Exploiting redundancy in question answering. *International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2001.

[3] M. Collins and Y. Singer. Unsupervised models for named entity classification. *Conference on Empirical Methods in Natural Language Processing (EMNLP)/VLC*, 1999.

[4] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? *International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 2002.

[5] A. Echihabi and D. Marcu. A noisy channel approach to question answering. *Association for Computational Linguistics Conference (ACL)*, 2003.

[6] M. Fleischman, E. Hovy, and A. Echihabi. Offline strategies for online question answering: Answering

questions before they are asked. *Association for Computational Linguistics Conference (ACL)*, 2003.

[7] R. Girju, D. Moldovan, and A. Badulescu. Learning semantic constraints for the automatic discovery of part-whole relations. *Human Language Technology and North American chapter of the Association for Computational Linguistics joint conference (HLT-NAACL)*, 2003.

[8] U. Hermjakob, E. Hovy, and C. Lin. Knowledge-based question answering. *Text Retrieval Conference (TREC)*, 2000.

[9] E. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C. Lin. Question answering in webclopedia. *Text Retrieval Conference (TREC)*, 2000.

[10] E. Hovy, U. Hermjakob, C. Lin, and D. Ravichandran. Using knowledge to facilitate factoid answer pinpointing. *International Conference on Computational Linguistics (COLING)*, 2002.

[11] L.V. Lita and J. Carbonell. Instance-based question answering: A data-driven approach. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

[12] L.V. Lita, W. Hunt, and E. Nyberg. Resource analysis for question answering. *Association for Computational Linguistics Conference (ACL)*, 2004.

[13] B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Penas, V. Peiado, F. Verdejo, and M. de Rijke. The multiple language question answering track at cross-lingual evaluation forum (clef) 2003. *Cross-Lingual Evaluation Forum (CLEF)*, 2003.

[14] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. *International Journal of Lexicography*, 1990.

[15] D. Moldovan, D. Clark, S. Harabagiu, and S. Maiorano. Cogex: A logic prover for question answering.

Association for Computational Linguistics Conference (ACL), 2003.

[16] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. The structure and performance of an open-domain question answering system. *"Association for Computational Linguistics Conference (ACL)*, 2000.

[17] E. Nyberg, T. Mitamura, J. Callan, J. Carbonell, R. Frederking, K. Collins-Thompson, L. Hiyakumoto, Y. Huang, C. Huttenhower, S. Judy, J. Ko, A. Kupsc, L. V. Lita, V. Pedro, D. Svoboda, and B. V. Durme. The javelin question-answering system at trec 2003: A multi strategy approach with dynamic planning. *Text Retrieval Conference (TREC)*, 2003.

[18] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. *Association for Computational Linguistics Conference (ACL)*, 2002.

[19] D. Ravichandran, A. Ittycheriah, and S. Roukos. Automatic derivation of surface text patterns for a maximum entropy based question answering system. *Human Language Technology and North American chapter of the Association for Computational Linguistics joint conference (HLT-NAACL)*, 2003.

[20] M. Thelen and E. Riloff. A bootstrapping method for learning semantic lexicons using extraction pattern contexts. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[21] E. Voorhees. Overview of the text retrieval conference (trec) 2003 question answering track. *Text Retrieval Conference (TREC)*, 2003.

[22] D. Yarowsky. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. *Association for Computational Linguistics Conference (ACL)*, 1994.