Offline Strategies for Online Question Answering: Answering Questions Before They Are Asked

Michael Fleischman, Eduard Hovy,
Abdessamad Echihabi
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
{fleisch, hovy, echihabi} @ISI.edu

Abstract

Recent work in Question Answering has focused on web-based systems that extract answers using simple lexico-syntactic patterns. We present an alternative strategy in which patterns are used to extract highly precise relational information offline, creating a data repository that is used to efficiently answer questions. We evaluate our strategy on a challenging subset of questions, i.e. "Who is …" questions, against a state of the art web-based Question Answering system. Results indicate that the extracted relations answer 25% more questions correctly and do so three orders of magnitude faster than the state of the art system.

1 Introduction

Many of the recent advances in Question Answering have followed from the insight that systems can benefit by exploiting the redundancy of information in large corpora. Brill et al. (2001) describe using the vast amount of data available on the World Wide Web to achieve impressive performance with relatively simple techniques. While the Web is a powerful resource, its usefulness in Question Answering is not without limits.

The Web, while nearly infinite in content, is not a complete repository of useful information. Most newspaper texts, for example, do not remain accessible on the Web for more than a few weeks. Further, while Information Retrieval techniques are relatively successful at managing the vast quantity of text available on the Web, the exactness required of Question Answering systems makes them too slow and impractical for ordinary users.

In order to combat these inadequacies, we propose a strategy in which information is extracted automatically from electronic texts offline, and stored for quick and easy access. We borrow techniques from Text Mining in order to extract semantic relations (e.g., concept-instance relations) between lexical items. We enhance these techniques by increasing the yield and precision of the relations that we extract.

Our strategy is to collect a large sample of newspaper text (15GB) and use multiple part of speech patterns to extract the semantic relations. We then filter out the noise from these extracted relations using a machine-learned classifier. This process generates a high precision repository of information that can be accessed quickly and easily.

We test the feasibility of this strategy on one semantic relation and a challenging subset of questions, i.e., "Who is …" questions, in which either a concept is presented and an instance is requested (e.g., "Who is the mayor of Boston?"), or an instance is presented and a concept is requested (e.g., "Who is Jennifer Capriati?"). By choosing this subset of questions we are able to focus only on answers given by concept-instance relationships. While this paper examines only this type of relation, the techniques we propose are easily extensible to other question types.

Evaluations are conducted using a set of "Who is …" questions collected over the period of a few

months from the commercial question-based search engine www.askJeeves.com. We extract approximately 2,000,000 concept-instance relations from newspaper text using syntactic patterns and machine-learned filters (e.g., "president Bill Clinton" and "Bill Clinton, president of the USA,"). We then compare answers based on these relations to answers given by TextMap (Hermjakob et al., 2002), a state of the art web-based question answering system. Finally, we discuss the results of this evaluation and the implications and limitations of our strategy.

2 Related Work

A great deal of work has examined the problem of extracting semantic relations from unstructured text. Hearst (1992) examined extracting hyponym data by taking advantage of lexical patterns in text. Using patterns involving the phrase "such as", she reports finding only 46 relations in 20M of *New York Times* text. Berland and Charniak (1999) extract "part-of" relations between lexical items in text, achieving only 55% accuracy with their method. Finally, Mann (2002) describes a method for extracting instances from text that takes advantage of part of speech patterns involving proper nouns. Mann reports extracting 200,000 concept-instance pairs from 1GB of *Associated Press* text, only 60% of which were found to be legitimate descriptions.

These studies indicate two distinct problems associated with using patterns to extract semantic information from text. First, the patterns yield only a small amount of the information that may be present in a text (the Recall problem). Second, only a small fraction of the information that the patterns yield is reliable (the Precision problem).

3 Relation Extraction

Our approach follows closely from Mann (2002). However, we extend this work by directly addressing the two problems stated above. In order to address the Recall problem, we extend the list of patterns used for extraction to take advantage of appositions. Further, following Banko and Brill (2001), we increase our yield by increasing the amount of data used by an order of magnitude over previously published work. Finally, in order to address the Precision problem, we use machine learning techniques to filter the output of the part of speech patterns, thus purifying the extracted instances.

3.1 Data Collection and Preprocessing

Approximately 15GB of newspaper text was collected from: the TREC 9 corpus (~3.5GB), the TREC 2002 corpus (~3.5GB), Yahoo! News (.5GB), the *AP* newswire (~2GB), the *Los Angeles Times* (~.5GB), the *New York Times* (~2GB), *Reuters* (~.8GB), the *Wall Street Journal* (~1.2GB), and various online news websites (~.7GB). The text was cleaned of HTML (when necessary), word and sentence segmented, and part of speech tagged using Brill's tagger (Brill, 1994).

3.2 Extraction Patterns

Part of speech patterns were generated to take advantage of two syntactic constructions that often indicate concept-instance relationships: common noun/proper noun constructions (CN/PN) and appositions (APOS). Mann (2002) notes that concept-instance relationships are often expressed by a syntactic pattern in which a proper noun follows immediately after a common noun. Such patterns (e.g. "president George Bush") are very productive and occur 40 times more often than patterns employed by Hearst (1992). Table 1 shows the regular expression used to extract such patterns along with examples of extracted patterns.

${NNP}*${VBG}*${JJ}*${NN}+${NNP}+
trainer/NN Victor/NNP Valle/NNP
ABC/NN spokesman/NN Tom/NNP Mackin/NNP
official/NN Radio/NNP Vilnius/NNP
German/NNP expert/NN Rriedhart/NNP
Dumez/NN Investment/NNP

Table 1. The regular expression used to extract CN/PN patterns (common noun followed by proper noun). Examples of extracted text are presented below. Text in bold indicates that the example is judged illegitimate.

${NNP}+\s*,\/,\s*${DT}*${JJ}*${NN}+(?:of\/IN)* \s*${NNP}*${NN}*${IN}*${DT}*${NNP}* ${NN}*${IN}*${NN}*${NNP}*,\/,
Stevens/NNP ,/, president/NN of/IN the/DT firm/NN ,/,
Elliott/NNP Hirst/NNP ,/, md/NN of/IN Oldham/NNP Signs/NNP ,/,
George/NNP McPeck/NNP,/, an/DT engineer/NN from/IN Peru/NN,/,
Marc/NNP Jonson/NNP,/, police/NN chief/NN of/IN Chamblee/NN ,/,
David/NNP Werner/NNP ,/, a/DT real/JJ estate/NN investor/NN ,/,

Table 2. The regular expression used to extract APOS patterns (syntactic appositions). Examples of extracted text are presented below. Text in bold indicates that the example is judged illegitimate.

In addition to the CN/PN pattern of Mann (2002), we extracted syntactic appositions (APOS). This pattern detects phrases such as "Bill Gates, chairman of Microsoft,". Table 2 shows the regular expression used to extract appositions and examples of extracted patterns. These regular expressions are not meant to be exhaustive of all possible varieties of patterns construed as CN/PN or APOS. They are "quick and dirty" implementations meant to extract a large proportion of the patterns in a text, acknowledging that some bad examples may leak through.

3.3 Filtering

The concept-instance pairs extracted using the above patterns are very noisy. In samples of approximately 5000 pairs, 79% of the APOS extracted relations were legitimate, and only 45% of the CN/PN extracted relations were legitimate. This noise is primarily due to overgeneralization of the patterns (e.g., "Berlin Wall, the end of the Cold War,") and to errors in the part of speech tagger (e.g., "Winnebago/CN Industries/PN"). Further, some extracted relations were considered either incomplete (e.g., "political commentator Mr. Bruce") or too general (e.g., "meeting site Bourbon Street") to be useful. For the purposes of learning a filter, these patterns were treated as illegitimate.

In order to filter out these noisy concept-instance pairs, 5000 outputs from each pattern were hand tagged as either legitimate or illegitimate, and used to train a binary classifier. The annotated examples were split into a training set (4000 examples), a validation set (500 examples); and a held out test set (500 examples). The WEKA machine learning package (Witten and Frank, 1999) was used to test the performance of various learning and meta-learning algorithms, including Naïve Bayes, Decision Tree, Decision List, Support Vector Machines, Boosting, and Bagging.

Table 4 shows the list of features used to describe each concept-instance pair for training the CN/PN filter. Features are split between those that deal with the entire pattern, only the concept, only the instance, and the pattern's overall orthography. The most powerful of these features examines an Ontology in order to exploit semantic information about the concept's head. This semantic information is found by examining the super-concept relations of the concept head in the 110,000 node Omega Ontology (Hovy et al., in prep.).

Feature Type	**Pattern Features**
Binary	${JJ}+${NN}+${NNP}+
Binary	${NNP}+${JJ}+${NN}+${NNP}+
Binary	${NNP}+${NN}+${NNP}+
Binary	${NNP}+${VBG}+${JJ}+${NN}+${NNP}+
Binary	${NNP}+${VBG}+${NN}+${NNP}+
Binary	${NN}+${NNP}+
Binary	${VBG}+${JJ}+${NN}+${NNP}+
Binary	${VBG}+${NN}+${NNP}+
	Concept Features
Binary	Concept head ends in "er"
Binary	Concept head ends in "or"
Binary	Concept head ends in "ess"
Binary	Concept head ends in "ist"
Binary	Concept head ends in "man"
Binary	Concept head ends in "person"
Binary	Concept head ends in "ant"
Binary	Concept head ends in "ial"
Binary	Concept head ends in "ate"
Binary	Concept head ends in "ary"
Binary	Concept head ends in "iot"
Binary	Concept head ends in "ing"
Binary	Concept head is-a occupation
Binary	Concept head is-a person
Binary	Concept head is-a organization
Binary	Concept head is-a company
Binary	Concept includes digits
Binary	Concept has non-word
Binary	Concept head in general list
Integer	Frequency of concept head in CN/PN
Integer	Frequency of concept head in APOS
	Instance Features
Integer	Number of lexical items in instance
Binary	Instance contains honorific
Binary	Instance contains common name
Binary	Instance ends in honorific
Binary	Instance ends in common name
Binary	Instance ends in determiner
	Case Features
Integer	Instance: # of lexical items all Caps
Integer	Instance: # of lexical items start w/ Caps
Binary	Instance: All lexical items start w/ Caps
Binary	Instance: All lexical items all Caps
Integer	Concept: # of lexical items all Caps
Integer	Concept: # of lexical items start w/ Caps
Binary	Concept: All lexical items start w/ Caps
Binary	Concept: All lexical items all Caps
Integer	Total # of lexical items all Caps
Integer	Total # of lexical items start w/ Caps

Table 4. Features used to train CN/PN pattern filter. Pattern features address aspects of the entire pattern, Concept features look only at the concept, Instance features examine elements of the instance, and Case features deal only with the orthography of the lexical items.

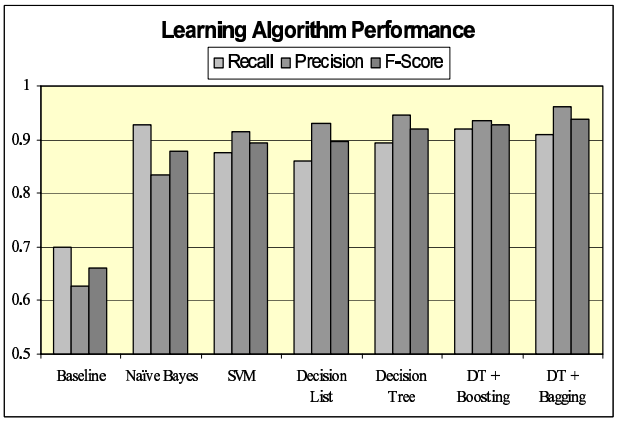

Figure 1. Performance of machine learning algorithms on a validation set of 500 examples extracted using the CN/PN pattern. Algorithms are compared to a baseline in which only concepts that inherit from "Human" or "Occupation" in Omega pass through the filter.

4 Extraction Results

4.1 Machine Learning Results

Figure 1 shows the performance of different machine learning algorithms, trained on 4000 extracted CN/PN concept-instance pairs, and tested on a validation set of 500. Naïve Bayes, Support Vector Machine, Decision List and Decision Tree algorithms were all evaluated and the Decision Tree algorithm (which scored highest of all the algorithms) was further tested with Boosting and Bagging meta-learning techniques. The algorithms are compared to a baseline filter that accepts concept-instance pairs if and only if the concept head is a descendent of either the concept "Human" or the concept "Occupation" in Omega. It is clear from the figure that the Decision Tree algorithm plus Bagging gives the highest precision and overall F-score. All subsequent experiments are run using this technique.[1]

Since high precision is the most important criterion for the filter, we also examine the performance of the classifier as it is applied with a threshold. Thus, a probability cutoff is set such that only positive classifications that exceed this cutoff are actually classified as legitimate. Figure 2 shows a plot of the precision/recall tradeoff as this threshold is changed. As the threshold is raised, precision increases while recall decreases. Based on this graph we choose to set the threshold at 0.9.

Applying the Decision Tree algorithm with Bagging, using the pre-determined threshold, to the held out test set of 500 examples extracted with the CN/PN pattern yields a precision of .95 and a recall of .718. Under these same conditions, but applied to a held out test set of 500 examples extracted with the APOS pattern, the filter has a precision of .95 and a recall of .92.

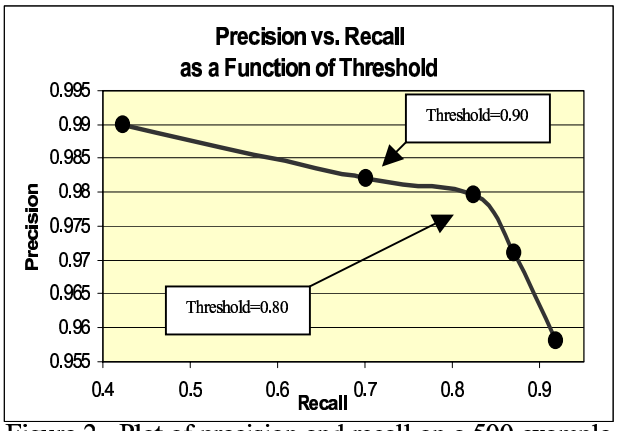

Figure 2. Plot of precision and recall on a 500 example validation set as a threshold cutoff for positive classification is changed. As the threshold is increased, precision increases while recall decreases. At the 0.9 threshold value, precision/recall on the validation set is 0.98/0.7, on a held out test set it is 0.95/0.72.

4.2 Final Extraction Results

The CN/PN and APOS filters were used to extract concept-instance pairs from unstructured text. The approximately 15GB of newspaper text (described above) was passed through the regular expression patterns and filtered through their appropriate learned classifier. The output of this process is approximately 2,000,000 concept-instance pairs. Approximately 930,000 of these are unique pairs, comprised of nearly 500,000 unique instances[2], paired with over 450,000 unique concepts[3] (e.g.,

[1] Precision and Recall here refer only to the output of the extraction patterns. Thus, 100% recall indicates that all legitimate concept-instance pairs that were extracted using the patterns, were classified as legitimate by the filter. It does not indicate that all concept-instance information in the text was extracted. Precision is to be understood similarly.

[2] Uniqueness of instances is judged here solely on the basis of surface orthography. Thus, "Bill Clinton" and "William Clinton" are considered two distinct instances. The effects of collapsing such cases will be considered in future work.

[3] As with instances, concept uniqueness is judged solely on the basis of orthography. Thus, "Steven Spielberg" and "J. Edgar Hoover" are both considered instances of the single concept

"sultry screen actress"), which can be categorized based on nearly 100,000 unique complex concept heads (e.g., "screen actress") and about 14,000 unique simple concept heads (e.g., "actress"). Table 3 shows examples of this output.

A sample of 100 concept-instance pairs was randomly selected from the 2,000,000 extracted pairs and hand annotated. 93% of these were judged legitimate concept-instance pairs.

Concept head	Concept	Instance
Producer	Executive producer	Av Westin
Newspaper	Military newspaper	Red Star
Expert	Menopause expert	Morris Notwlovitz
Flutist	Flutist	James Galway

Table 3. Example of concept-instance repository. Table shows extracted relations indexed by concept head, complete concept, and instance.

5 Question Answering Evaluation

A large number of questions were collected over the period of a few months from www.askJeeves.com. 100 questions of the form "Who is *x*" were randomly selected from this set. The questions queried concept-instance relations through both instance centered queries (e.g., "Who is Jennifer Capriati?") and concept centered queries (e.g., "Who is the mayor of Boston?"). Answers to these questions were then automatically generated both by look-up in the 2,000,000 extracted concept-instance pairs and by TextMap, a state of the art web-based Question Answering system which ranked among the top 10 systems in the TREC 11 Question Answering track (Hermjakob et al., 2002).

Although both systems supply multiple possible answers for a question, evaluations were conducted on only one answer.[4] For TextMap, this answer is just the output with highest confidence, i.e., the system's first answer. For the extracted instances, the answer was that concept-instance pair that appeared most frequently in the list of extracted examples. If all pairs appear with equal frequency, a selection is made at random.

Answers for both systems are then classified by hand into three categories based upon their information content.[5] Answers that unequivocally identify an instance's celebrity (e.g., "Jennifer Capriati is a tennis star") are marked correct. Answers that provide some, but insufficient, evidence to identify the instance's celebrity (e.g., "Jennifer Capriati is a defending champion") are marked partially correct. Answers that provide no information to identify the instance's celebrity (e.g., "Jennifer Capriati is a daughter") are marked incorrect.[6] Table 5 shows example answers and judgments for both systems.

	State of the Art		Extraction	
	Answer	Mark	Answer	Mark
Who is Nadia Comaneci?	U.S. citizen	P	Romanian Gymnast	C
Who is Lilian Thuram?	News page	I	French defender	P
Who is the mayor of Wash., D.C.?	Anthony Williams	C	*no answer found*	I

Table 5. Example answers and judgments of a state of the art system and look-up method using extracted concept-instance pairs on questions collected online. Ratings were judged as either correct (C), partially correct (P), or incorrect (I).

6 Question Answering Results

Results of this comparison are presented in Figure 3. The simple look-up of extracted concept-instance pairs generated 8% more partially correct answers and 25% more entirely correct answers than TextMap. Also, 21% of the questions that TextMap answered incorrectly, were answered partially correctly using the extracted pairs; and 36% of the questions that TextMap answered incorrectly, were answered entirely correctly using the extracted pairs. This suggests that over half of the questions that TextMap got wrong could have benefited from information in the concept-instance pairs. Finally, while the look-up of extracted pairs took approximately ten seconds for all 100 questions, TextMap took approximately 9 hours.

"director." See Fleischman and Hovy (2002) for techniques useful in disambiguating such instances.

[4] Integration of multiple answers is an open research question and is not addressed in this work.

[5] Evaluation of such "definition questions" is an active research challenge and the subject of a recent TREC pilot study. While the criteria presented here are not ideal, they are consistent, and sufficient for a system comparison.

[6] While TextMap is guaranteed to return some answer for every question posed, there is no guarantee that an answer will be found amongst the extracted concept-instance pairs. When such a case arises, the look-up method's answer is counted as incorrect.

This difference represents a time speed up of three orders of magnitude.

There are a number of reasons why the state of the art system performed poorly compared to the simple extraction method. First, as mentioned above, the lack of newspaper text on the web means that TextMap did not have access to the same information-rich resources that the extraction method exploited. Further, the simplicity of the extraction method makes it more resilient to the noise (such as parser error) that is introduced by the many modules employed by TextMap. And finally, because it is designed to answer any type of question, not just "Who is…" questions, TextMap is not as precise as the extraction technique. This is due to both its lack of tailor made patterns for specific question types, as well as, its inability to filter those patterns with high precision.

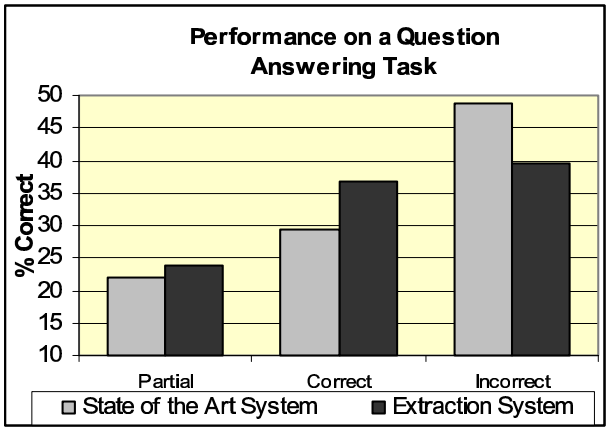

Figure 3. Evaluation results for the state of the art system and look-up method using extracted concept-instance pairs on 100 "Who is …" questions collected online. Results are grouped by category: partially correct, entirely correct, and entirely incorrect.

7 Discussion and Future Work

The information repository approach to Question Answering offers possibilities of increased speed and accuracy for current systems. By collecting information offline, on text not readily available to search engines, and storing it to be accessible quickly and easily, Question Answering systems will be able to operate more efficiently and more effectively.

In order to achieve real-time, accurate Question Answering, repositories of data much larger than that described here must be generated. We imagine huge data warehouses where each repository contains relations, such as birthplace-of, location-of, creator-of, etc. These repositories would be automatically filled by a system that continuously watches various online news sources, scouring them for useful information.

Such a system would have a large library of extraction patterns for many different types of relations. These patterns could be manually generated, such as the ones described here, or learned from text, as described in Ravichandran and Hovy (2002). Each pattern would have a machine-learned filter in order to insure high precision output relations. These relations would then be stored in repositories that could be quickly and easily searched to answer user queries. [7]

In this way, we envision a system similar to (Lin et al., 2002). However, instead of relying on costly structured databases and pain stakingly generated wrappers, repositories are automatically filled with information from many different patterns. Access to these repositories does not require wrapper generation, because all information is stored in easily accessible natural language text. The key here is the use of learned filters which insure that the information in the repository is clean and reliable.

Such a system is not meant to be complete by itself, however. Many aspects of Question Answering remain to be addressed. For example, question classification is necessary in order to determine which repositories (i.e., which relations) are associated with which questions.

Further, many question types require post processing. Even for "Who is …" questions multiple answers need to be integrated before final output is presented. An interesting corollary to using this offline strategy is that each extracted instance has with it a frequency distribution of associated concepts (e.g., for "Bill Clinton": 105 "US president"; 52 "candidate"; 4 "nominee"). This distribution can be used in conjunction with time/stamp information to formulate mini biographies as answers to "Who is …" questions.

We believe that generating and maintaining information repositories will advance many aspects of Natural Language Processing. Their uses in

[7] An important addition to this system would be the inclusion of time/date stamp and data source information. For, while "George Bush" is "president" today, he will not be forever.

data driven Question Answering are clear. In addition, concept-instance pairs could be useful in disambiguating references in text, which is a challenge in Machine Translation and Text Summarization.

In order to facilitate further research, we have made the extracted pairs described here publicly available at www.isi.edu/~fleisch/instances.txt.gz. In order to maximize the utility of these pairs, we are integrating them into an Ontology, where they can be more efficiently stored, cross-correlated, and shared.

Acknowledgments

The authors would like to thank Miruna Ticrea for her valuable help with training the classifier. We would also like to thank Andrew Philpot for his work on integrating instances into the Omega Ontology, and Daniel Marcu whose comments and ideas were invaluable.

References

Michelle Banko, Eric Brill. 2001. Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proceedings of the Association for Computational Linguistics*, Toulouse, France.

Matthew Berland and Eugene Charniak. 1999. Finding Parts in Very Large Corpora. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. College Park, Maryland.

Eric Brill. 1994. Some advances in rule based part of speech tagging. *Proc. of AAAI.* Seattle, Washington.

Eric Brill, Jimmy Lin, Michele Banko, Susan Dumais, and Andrew Ng. 2001. Data-Intensive Question Answering. *Proceedings of the 2001 Text REtrieval Conference (TREC 2001)*, Gaithersburg, MD.

Michael Fleischman and Eduard Hovy. 2002. Fine Grained Classification of Named Entities. *19th International Conference on Computational Linguistics (COLING)*. Taipei, Taiwan.

Ulf Hermjakob, Abdessamad Echihabi, and Daniel Marcu. 2002. Natural Language Based Reformulation Resource and Web Exploitation for Question Answering. In *Proceedings of the TREC-2002 Conference, NIST*. Gaithersburg, MD.

Marti Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France.

Jimmy Lin, Aaron Fernandes, Boris Katz, Gregory Marton, and Stefanie Tellex. 2002. Extracting Answers from the Web Using Data Annotation and Data Mining Techniques. *Proceedings of the 2002 Text REtrieval Conference (TREC 2002)* Gaithersburg, MD.

Gideon S. Mann. 2002. Fine-Grained Proper Noun Ontologies for Question Answering. *SemaNet'02: Building and Using Semantic Networks*, Taipei, Taiwan.

Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a Question Answering system. *Proceedings of the 40th ACL conference*. Philadelphia, PA.

I. Witten and E. Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with JAVA implementations*. Morgan Kaufmann, San Francisco, CA.