Mining and Re-ranking for Answering Biographical Queries on the Web

Donghui Feng **Deepak Ravichandran** **Eduard Hovy**

Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
{donghui, hovy}@isi.edu

Abstract

The rapid growth of the Web has made itself a huge and valuable knowledge base. Among them, biographical information is of great interest to society. However, there has not been an efficient and complete approach to automated biography creation by querying the web. This paper describes an automatic web-based question answering system for biographical queries. Ad-hoc improvements on pattern learning approaches are proposed for mining biographical knowledge. Using bootstrapping, our approach learns surface text patterns from the web, and applies the learned patterns to extract relevant information. To reduce human labeling cost, we propose a new IDF-inspired re-ranking approach and compare it with pattern's precision-based re-ranking approach. A comparative study of the two re-ranking models is conducted. The tested system produces promising results for answering biographical queries.

Introduction

The rapid growth of the Web has made itself a huge repository of large amounts of valuable information. Among them, biography information is of great interest to society. Given a person's name, the challenge is to determine automatically relevant biographical information such as, birth date, birth place, job, spouse, and so on. For example, students need to learn about historical figures in class; people are interested to know such knowledge about politicians and movie stars.

However, till now there has not been an efficient and complete approach to be able to correctly retrieve required biography information. Most databases about people's information are limited and cannot return required results for people in different areas. As the large corpora on internet become available, we can have the access to a large biography source.

In this paper, we propose to intelligently mine and re-rank human biographical information from the web. For famous people, the web contains lots of useful articles related to him/her, as given by search engines. However, to acquire the biographical information, humans have to read the retrieved documents and extract the desired information manually, which is very tedious, time- and cost-consuming. Automated or semi-automated methods can help; they can be inserted immediately after search engines retrieve relevant documents.

We have built a system that automatically extracts some biographical information from the web. Some such information is already available in structured format like tables, and can be extracted using information integration techniques (Katz et al., 2002; Lerman et al., 2004). But much of most biographical information is scattered throughout documents in narrative form.

1. Albert Einstein was born at Ulm, in Württemberg, Germany, on March 14, 1879.
2. Albert Einstein (1879-1955), German-born American theoretical physicist, theories of relativity, philosopher.
3. Albert Einstein was born in 1879 in Ulm, Germany.
4. Albert Einstein's Birthday - March 14, 1879.

Figure 1. Example sentences for birth date expressions.

Born in Ulm, Germany in 1879, Albert Einstein is still considered one of the greatest scientific and mathematical geniuses in history. In 1905, at the age of 26, he set forth his theory of relativity which discards the concept of time and space as absolute entities, and views them as relative to moving frames of reference. ... In 1950, he presented his unified field theory, which attempts to explain gravitation, electromagnetism, and subatomic phenomena in one set of laws. He completed its mathematical formulation in 1953, just two years before his death in 1955 at the age of 76.

Figure 2. An example of long distance dependency.

Natural language poses several fairly challenging problems to be solved by biography extractors. First, such knowledge can be expressed in different ways. For example, for the birth date of Albert Einstein, Figure 1 shows some different expressions collected from the web. An efficient extraction system should be able to handle all the sentences appropriately. Second, the required biographical information is not always expressed in one sentence. Figure 2 gives an example, in which the name "Albert Einstein" appears only once, while most of the interesting facts occur in subsequent sentences and are related via coreference. Coreference resolution is a well-

known difficult problem for language processing systems. Third, information extracted from the web may be ambiguous and even contradictory. Fourth, it is possible that a name might refer to distinct individuals (e.g., a politician and a rock singer are both named Paul Simon, and the two Presidents George Bush pose a very current problem). In this case, we may obtain from the extraction engine mixed information about the different individuals. Correctly re-ranking the candidate answers is highly desirable.

In this paper, we define biographical information (initially) using 5 attributes, namely, birth date, birth place, death date, death place, and spouse. To solve the problem of paraphrasing, we employ and extend a bootstrapping approach, reported in (Ravichandran & Hovy, 2002), to learn surface text patterns from the web. For extraction of desired biographical information, the learned patterns are applied to articles returned by the search engine (Google in our case). This provides an N-best list of each biography field value. To reduce human labeling cost, we propose a new IDF-inspired re-ranking approach and compare it with the pattern's precision-based approach.

The rest of this paper is organized as follows: Section 2 describes related work. In Section 3, we present our methodology for web-based question answering system, including the mining and re-ranking approaches. Section 4 gives experimental results and discusses related issues. The paper concludes with Section 5.

Related Work

Automatic or semi-automatic extraction techniques on a given corpus have been given much attention in information extraction and question answering communities (Manning, 1993; Hearst, 1998; Cao et al., 2001; Lin and Pantel, 2001; Hasegawa et al., 2004). The Automatic Content Extraction (ACE) [1] program has explored the detection and characterization of Entities, Relations, and Events. In recent years, the main approaches have also been applied to molecular biology (Saric et al., 2004; Rosario and Hearst, 2004) and scientific patents (Wu et al., 2003). As mentioned above, some researchers in Question Answering, e.g., the SMART system (Katz et al., 2002), and Information Integration (Lerman et al., 2004; Thakkar et al., 2004) have focused on extracting information from well-structured sources to build specialized databases. However, till now there has been limited work focused on the domain of automatic biography extraction, especially from the Web.

Craven et al. (2000) proposes to construct a knowledge base from the World Wide Web. Their work is to transform the World Wide Web into a computer-understandable knowledge base and requires the analysis of the relations between web pages. It differs from ours since we use the Web only as the text source for automatically learning patterns and extracting biography information.

Typically, supervised machine learning algorithms require a set of annotated training data. Ravichandran and Hovy (2002) show how this work can be reduced by providing the search engine with only a few pairs of examples as 'seed data'. Each pair consists of a 'question term' (in our case, the person's name) and an 'answer term' (in our case, the value of the relation being studied, such as "March 14, 1879" for the relation Birthdate). Then the approach can learn paraphrasing patterns from the articles returned by the search engine.

However, their method exhibits several shortcomings when applied to biography information extraction. First, it requires both the question term and the answer term to appear in the same sentence, which is not suitable for biographies, where coreference is common. We extend their algorithm by adding a simple form of coreference resolution. Second, when determining the length of the answer phrase, their approach searches a fixed range of 50 bytes and requires the answer tag to be replaced by only one word. This does not work for biography information, especially for dates and places, in which the phrase may include several words. We use BBN's IdentiFinder [2] to determine phrasal boundaries for dates, places, and spouses. Third, for variations in writing of dates, places, and person names, we improve their approach by creating three types of variation constructor. Finally, their approach doesn't investigate any features that the answers may have, and doesn't provide an appropriate re-ranking model, which is essential when a list of candidate answers is returned. We propose a new IDF-inspired re-ranking approach and compare it with the pattern's precision-based approach. We investigate re-ranking by both precision scores and AIDF scores to optimize performance. We also discuss to extract biographical information including multiple correct answers with inherent underlying relationships.

We present our mining and re-ranking models in detail in Section 3.

Methodology

Mining Candidate Answers

Biography Attributes. A person's biography may include information of a variety of aspects. As mentioned, we initially work with the 5 attributes Birthdate, Birthplace, Deathdate, Deathplace, and Spouse. Although at present we have only extracted these 5 attributes, the approach presented here can be similarly applied to other attributes.

Surface Pattern Learning. In our approach, mining biography information can be divided into 3 phases, shown in Figure 3. First, the seed data (question and answer term pairs) are provided to the WWW search engine (Google).

1 http://www.ldc.upenn.edu/Projects/ACE/

2 http://www.bbn.com/For_Commercial_Customers/Unstructured_Data/Identifinder.html

Using the articles returned, the surface text patterns are learned. We tag the sentences with <_QT> for question term and <_AT> for answer term, obtaining, for example, the sentence patterns in Figure 4 from the sentences in Figure 1. We follow the procedure of Ravichandran and Hovy (2002) to apply Suffix Trees to find the longest common string (LCS) in each case. However, as mentioned, we use the phrasal boundaries of IdentiFinder to delimit multi-word units. Using the patterns so learned, we are in a position to mine personal bios from the Web. New queries with person names are sent to the search engine to get retrieved articles for bio extraction and re-ranking.

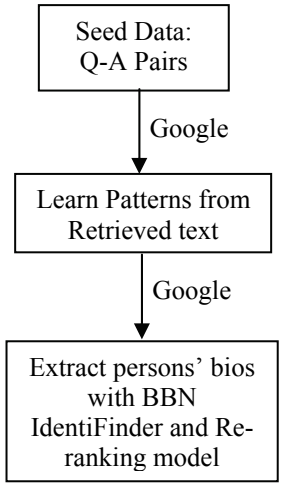

Figure 3. Procedure of biography extraction.

1. <_QT> was born at Ulm, in Württemberg, Germany, on <_AT>.
2. <_QT> (<_AT>-1955), German-born American theoretical physicist, theories of relativity, philosopher.
3. <_QT> was born in <_AT> in Ulm, Germany.
4. <_QT> 's Birthday - <_AT>.

Figure 4. Surface pattern samples.

Re-ranking Candidate Answers

We apply learned text patterns above to extract new person's biographical answers from the web. A re-ranking approach is required to re-rank all the candidate answers.

Suppose we have N candidate answers in the final N-best list, $A_i, i = 1,..., N$, For each candidate answer A_i , there are M_i patterns that produce this answer. We investigate both a precision-based re-ranking model and an IDF-inspired one.

Acquiring Precision by Supervised Learning. Candidate answers can be re-ranked based on the patterns' precision scores. Each pattern could acquire a precision score by supervised learning. We manually determined correct values for the training sets of biography attributes and compute each pattern' precision score using Formula 1:

$$precision(ptn) = \frac{C_a}{C_o} \quad (1)$$

where C_a is the total number of patterns containing the desired answer term and C_o is the total number of patterns with answer terms replaced with any word or phrase.

From this, we get five sets of patterns associated with their precision scores corresponding to the 5 biography attributes. The list of candidate answers could be re-ranked with these patterns' precision scores according to Formula 2.

$$Score(A_i) = \sum_{j=1}^{M_i} precision(ptn_j) \quad (2)$$

All the candidate answers are ranked in descending order according to this score.

IDF-inspired Heuristics. The precision-based re-ranking requires a training set to compute each pattern's precision score. Both the human labeling cost and the computing cost are not negligible, especially when the extraction of multiple values is required.

This drives us to look at the patterns themselves. Intuition tells us that patterns with more informative words can produce more accurate results than the patterns with general words. IDF (Inverse Document Frequency) score is a good measure for the information content of a term. We design a re-ranking model based on patterns' AvgIDF scores, which is an average of all terms' IDF values.

Suppose the pattern is composed of N words, $w_i, i = 1,..., N$, we define the pattern's Average IDF (AIDF) score using Formula 3.

$$AvgIDF(ptn_i) = \frac{1}{N} \sum_{j=1}^{N} IDF(w_j) \quad (3)$$

In this formula, a traditional definition of IDF is given as:

$$IDF(w_j) = \log(\frac{|D|}{DF(w_j)}) \quad (4)$$

where |D| is the total number of documents and $DF(w_j)$ counts the document frequency of term w_j. It equals to the number of documents that term wj appears at least once.

Inpired by the work in (Joachims, 1997), we give another definition of IDF as in formula (5). Here the definition of our IDF not only counts how many documents this term appears, but also how many times it appears in that document. The IDF value can be computed using Formula 5.

$$IDF(w_j) = \frac{|D|}{\sum_{d \in D} DF(w_j, d)} \quad (5)$$

Similar to Formula 2, re-ranking of the candidate answer list can be completed using a voting scheme on patterns' AIDF scores. The final score for each candidate answer, A_i by this re-ranking model can be expressed with Formula 6.

$$Score(A_i) = \sum_{j=1}^{M_i} AvgIDF(ptn_j) \quad (6)$$

Hereafter, we have both AIDF and precision scores for each pattern. We compare the two approaches' performances in next section.

Experiments

In Section 3, we discussed our surface pattern learning algorithm and the re-ranking models. In this section, we describe the system's performance in detail.

Experimental Setup

Initially, we need to provide the seed data to the search engine (Google) to obtain high-yield articles. The key point here is to try to make the question and answer term pair unique. For example, for birthdate, given the pair (Clinton, 1946), the search engine can find many articles in which these two terms correctly refer to birthdate. But birthplace is more difficult since it is possible that the person also lived there for some time, or had some other relationship to it, such as being the mayor or local beauty queen. The articles retrieved using these seed terms may include sentences that express such other relations in addition to birthplace. In this situation, the extracted samples and patterns will be more error-prone. Therefore, on average, we take about 5 to 25 data pairs as seed data, and use the top 1000 articles returned by the search engine to learn patterns.

To score the patterns' precision, we require a set of annotated biography data. We use a list of 10 people, with all their biography attributes manually annotated (all values included). We then compare the attribute values extracted from the top 1000 articles to get each pattern precision score. The learned patterns' AIDF scores are calculated based on the TREC 9 corpus.

For testing data, we collect 50 people and divide them into 5 categories to extract different biographical values. We then manually annotated the testing sets with their biographical data as the gold standard. To evaluate the system's performance, we use the accuracy score for the top N answers, which we interpret as measuring the probability of the correct answers appearing among the top N answers.

Supposing there are in total M persons in the test set, $H_1, H_2, ..., H_M$, the evaluation function is defined as follows:

$$\lambda_i = \begin{cases} 1 & \text{if the correct value appears in the top N list} \\ 0 & \text{Otherwise} \end{cases}$$

$i = 1, ..., M$.

Then the system's performance is in Formula 7.

$$Accuracy_{topN} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i \quad (7)$$

Examples

Using an average of 14 seed pairs, we learned an average of 200 patterns for each of the 5 attributes. We then applied them to the testing set with new people to construct a biography database automatically. The results are encouraging, though further refinement and filtering are needed, as discussed below. Some examples of the extracted biography information appear in Figure 7. As shown in the figure, each attribute has a top-10 candidate list ordered in the sum of patterns' precision scores computed with Formula 2. The list items in bold font indicate the true answers for each attribute.

Results Analysis

We tested the system's performance based on the evaluation metrics discussed above. Figure 5 and Figure 6 give accuracy scores for the 5 biography attributes with different re-ranking models as described in Formula 2 and Formula 6.

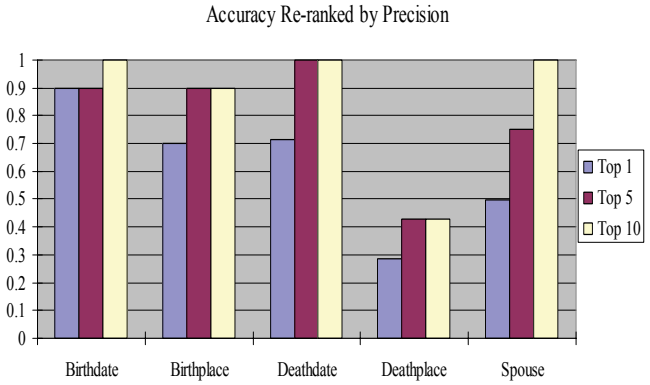

Figure 5. Top N accuracy re-ranked by precision score.

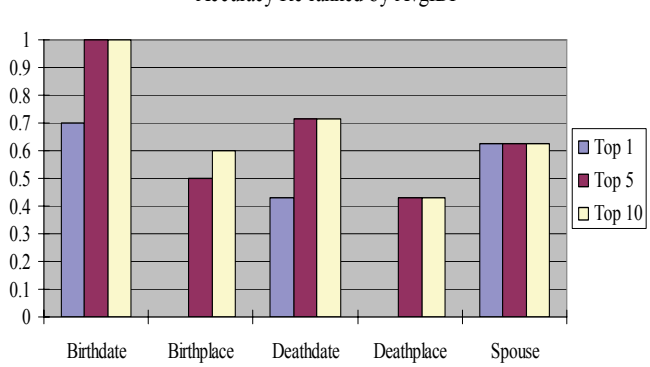

Figure 6. Top N accuracy re-ranked by AIDF.

From these two figures, we can see for the results, generally the performances re-ranked by the patterns' precision scores are better than those by the AIDF score. Also the performances of dates (Birthdate and Deathdate) are better than those of places (Birthplace and Deathplace). This is understandable. Because date has less irregular variations than place and it's relatively easier to learn patterns and extract information. Of all the five attributes, the Deathplace has the worst performance. One of the reasons is that typically there are fewer articles and

Person Name	Bithdate	Bithplace	Spouse
Ronald Reagan	1.8515 **1911** - 2004 1.3980 **1911** 1.2457 **February 6 , 1911** 1.0700 5 TRILLION 0.8726 today 0.8196 150 ; 2004 0.7683 **2 / 6 / 1911** 0.6867 1985 0.6617 1932 0.6383 **Feb**	1.7603 **Illinois** in 1911 1.2720 Screen Actors Guild 1.0531 **Tampico** 0.9185 **Tampico , Illinois** 0.8849 **Illinois** . 0.8167 United States 0.8148 1911 in **Tampico , Illinois** 0.6670 America 0.6547 ' s 0.6066 American	0.8704 **Nancy** 0.5517 **Jane Wyman** 0.4930 **Nancy Davis** 0.2840 Gorbachev 0.2462 Carter 0.1575 Bush 0.1397 Dubya 0.1147 Paul Kengor 0.1133 George Bush 0.1117 Jimmy Carter
Louis Armstrong	2.8571 **8 / 4 / 1901** 1.7871 **August 4** 1.6980 July 4 1.5260 **Aug** 1.3998 04 00 : 00 : 00 EST 0.9997 **1901** - 1971 0.8511 1900 - 71 0.8213 2 - 26 - 1926 0.8091 1900 - 1971 0.7835 11 - 16 - 1926	3.0414 **New Orleans** 1.1539 Chicago 1.1185 Jazz 0.9445 Storyville 0.8520 **New Orleans** on Aug 0.7097 New York City 0.5202 New York 0.4770 U . 0.4768 Wonderful World 0.4151 Ella Fitzgerald	0.4408 **Lucille Wilson** 0.4392 **Lillian Hardin** 0.425 pianist **Lillian Hardin** 0.1470 Oliver 0.1344 Ella 0.1235 **Lil** 0.1210 Charlie Parker 0.1210 Lena Horne 0.1169 **Lucille** 0.1036 King Oliver
Thor Heyerdahl	1.8800 **1914** - 2002 1.3987 **October 6 , 1914** 1.1291 6 1.1095 **1914** 1.0349 16 0.8673 150 ; 2002 0.8593 19 0.8291 3 : 1 0.8197 06.02.14 - 18.04.02 0.8149 Summer 2002	1.9353 **Norway** 1.1100 Easter Island 0.8892 Southern **Norway** 0.8148 **Norway** , 0.6313 Pacific 0.5699 Kon - Tiki 0.5372 southern Norwegian 0.4778 Kon - Tiki Fame 0.4446 Azerbaijan 0.4155 Canary Islands	0.2027 Kon - Tiki Man 0.1842 Betty Blair 0.1670 Kon - Tiki 0.1667 Early Cradle 0.1077 www 0.0986 Edwin N 0.0932 **Jacqueline** .09292 Geographic Diversity 0.0929 Arne Skjolsvold 0.0929 Per Lilliestrom
Niels Bohr	3.6131 **1885** - 1962 2.9127 Oct . 2.7014 **October 7** 2.5003 **1885** 1.3389 June 1 , 1987 1.2895 **7 Oct 1885** 1.1617 **7th October 1885** 1.1021 **7th October** 1.0695 25 th 0.7062 **10 / 7 / 1885** Died : 11 / 18 /	1.8172 **Denmark** 0.8148 Christian , 0.7461 University of Copenhagen 0.6208 Manchester 0.5604 ' s 0.5140 Sweden 0.4925 United States 0.4855 Nobel Prize 0.4431 L - R 0.3801 U .	0.3776 Aage 0.2668 Einstein 0.1925 Werner Heisenberg 0.1445 Rutherford 0.1212 Harald 0.1060 Ellen Adler 0.1002 **Margrethe** 0.0986 Blaise Pascal 0.0929 Rimestad 0.0929 James Rainwater

Figure 7. Examples of extracted biography information.

resources for the event of death than birth of famous people.

Discussions

Having come this far, several problems call out to be resolved. An important problem is that some biography attributes have multiple values. We can classify them into three categories based on the values' features:

- *Geographical Values*. The attribute may have multiple values, which are ordered in space. If the attribute is a place, say, birthplace, deathplace or workplace, the returned answer may be the geographical sub- or super-region of the desired answer. For example, if Edinburgh is the correct birthplace, then Edinburgh, Scotland, and U.K. may all three be correct in some situations. We may use a geography reasoner to integrate all the answers.
- *Precedence Values*. The attribute may have multiple values, which are ordered in time. For example, a person may have different spouses. The patterns may look like "the first XX", "the second XX", and "the third XX". An example pattern list of such values on the attribute Spouse is shown in Figure 8.
- *Discrete Values*. The attribute may have several values, but they are not ordered either in time or space. For example, a person may have several job titles

(1)	<_QT> MARRIED <_AT>
(2)	<_QT> AND <_AT> DIVORCED
(3)	<_QT> AND HIS WIFE <_AT>
(4)	<_QT> MARRIED <_AT> ,
(5)	<_AT> ' S RELATIONSHIP WITH <_QT>
(6)	<_AT> AND <_QT> WERE MARRIED
(7)	<_QT> AND <_AT> SEPARATED
(8)	<_AT> , <_QT> ' S FIRST WIFE .
(9)	<_AT> , <_QT> ' S SECOND WIFE
(10)	<_AT> ' S DECISION TO DUMP <_QT>
(11)	<_QT> ' S FIRST MARRIAGE , TO <_AT>
(12)	PARTNERSHIP BETWEEN <_QT> AND <_AT> .
(13)	<_QT> AND HIS FIRST WIFE , <_AT>
(14)	<_QT> ' S MARRIAGE TO <_AT>
(15)	< AT> WAS < QT> ' S SECOND WIFE

Figure 8. Patterns for multiple-values (spouse).

simultaneously, or have appeared in several movies.Here the requisite patterns would search for lists, and hence require certain orthographic markers such as commas or bullets.

So far, we have investigated only 5 biographical attributes. Our approach can be applied to other attributes such as job title, children, important accomplishments, etc.-anything for which suitably unique seed example pairs can be created. The algorithm is not dependent on the features of the attribute. As long as the system is given a small set of high-quality seed data, it can automatically learn the high-yield patterns and apply them to mine a person's biographical information later.

For each learned pattern, we also investigate the relation between its AIDF score and precision score. As expected, the larger a pattern's AIDF score is, the more information is carried in this pattern and it has a higher precision score. However, when the AIDF score is 0, the pattern's precision may not be 0. This is understandable. For example, for the pattern, "<_QT> (<_AT>", the AIDF score has no score for diacriticals such as parentheses, while they may be critical in our patterns and yield correct answers.

We have begun to investigate the situation where the attribute has multiple correct values (for example, the spouses of Ronald Reagan). Unfortunately, it is not the case that either precision or AIDF scores always clearly differentiate all the good values from all the spurious matches. Even using negative training data has not proven useful. We therefore continue to look for methods, possibly along the lines of learning ancillary patterns that should hold only in the case where multiple values exist, such as phrases using "both" or "all". A more knowledge-intensive approach would employ inferences associated with types to perform fact-checking (the spouse of a man must be a woman) and to suggest useful additional questions/patterns.

Conclusions

Biography information is of great interest to society, and an automated biography creation engine, even if it does not produce polished prose, may be useful for many applications. In this paper we outline some advances and some remaining short-term problems. Longer-term, the problem of outdated and/or changing information will become worse as the web grows, and necessitate additional research in web document trustworthiness. Fortunately, this question is already receiving some attention.

References

Cao. Y., Li, H., and Li, S. 2003. Learning and Exploiting Non-Consecutive String Patterns for Information Extraction, *Microsoft Research Asia Technical Report*. 2003.

Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., and Slattery, S. 2000. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence,* 118(1–2):69–113.

Hasegawa, T., Sekine, S., and Grishman, R. 2004. Discovering Relations among Named Entities form Large Corpora. In *Proceedings of ACL-2004.*

Hearst, M.A. 1998. Automated Discovery of WordNet Relations. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database.* MIT Press.

Joachims, T. 1997, A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In Proceedings of *ICML-1997*.

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A.J., and Temelkuran, B. 2002. Omnibase: Uniform Access to Heterogeneous Data for Question Answering. In *Proceedings of the 7th International Workshop NLDB 2002.*

Lerman, K., Getoor, L., Minton, S., and Knoblock, C.A. 2004. Using the structure of web sites for automatic segmentation of tables. In *Proceedings of ACM SIG on Management of Data (SIGMOD-2004)*.

Lin, D. and Pantel, P. 2001. DIRT. Discovery of Inference Rules from Text. *In Proceedings of ACM SIGKDD 2001.*

Manning, C.D. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of ACL-1993*.

Ravichandran, D. and Hovy, E.H. 2002. Learning Surface Text Patterns for a Question Answering System. In *Proceedings of ACL-2002.*

Rosario, B. and Hearst, M. 2004. Classifying Semantic Relations in Bioscience Text. In *Proceedings of ACL-2004.*

Saric, J., Jensen, L.J., Bork, P., Ouzounova, R., and Rojas, I. 2004. Extracting Regulatory Gene Expression Networks From Pubmed. In *Proceedings of ACL-2004.*

Thakkar, S., Ambite, J.L., and Knoblock, C.A. 2004. A data integration approach to automatically composing and optimizing web services. In *Proceedings of 2004 ICAPS.*

Wu, T., Holzman, L.E., Pottenger, W.M., and Phelps, D.J. 2003. A Supervised Learning Algorithm for the Discovery of Finite State Automata for Information Extraction from Textual Data. In *Proceedings of the Textmine '03 Workshop, Third SIAM International Conference on Data Mining.*