On-Demand Information Extraction

Satoshi Sekine
Computer Science Department
New York University
715 Broadway, 7th floor
New York, NY 10003 USA
sekine@cs.nyu.edu

Abstract

At present, adapting an Information Extraction system to new topics is an expensive and slow process, requiring some knowledge engineering for each new topic. We propose a new paradigm of Information Extraction which operates 'on demand' in response to a user's query. On-demand Information Extraction (ODIE) aims to completely eliminate the customization effort. Given a user's query, the system will automatically create patterns to extract salient relations in the text of the topic, and build tables from the extracted information using paraphrase discovery technology. It relies on recent advances in pattern discovery, paraphrase discovery, and extended named entity tagging. We report on experimental results in which the system created useful tables for many topics, demonstrating the feasibility of this approach.

1 Introduction

Most of the world's information is recorded, passed down, and transmitted between people in text form. Implicit in most types of text are regularities of information structure - events which are reported many times, about different individuals, in different forms, such as layoffs or mergers and acquisitions in news articles. The goal of information extraction (IE) is to extract such information: to make these regular structures explicit, in forms such as tabular databases. Once the information structures are explicit, they can be processed in many ways: to mine information, to search for specific information, to generate graphical displays and other summaries.

However, at present, a great deal of knowledge for automatic Information Extraction must be coded by hand to move a system to a new topic. For example, at the later MUC evaluations, system developers spent one month for the knowledge engineering to customize the system to the given test topic. Research over the last decade has shown how some of this knowledge can be obtained from annotated corpora, but this still requires a large amount of annotation in preparation for a new task. Improving portability - being able to adapt to a new topic with minimal effort – is necessary to make Information Extraction technology useful for real users and, we believe, lead to a breakthrough for the application of the technology.

We propose 'On-demand information extraction (ODIE)': a system which *automatically identifies the most salient structures and extracts the information on the topic the user demands*. This new IE paradigm becomes feasible due to recent developments in machine learning for NLP, in particular unsupervised learning methods, and it is created on top of a range of basic language analysis tools, including POS taggers, dependency analyzers, and extended Named Entity taggers.

2 Overview

The basic functionality of the system is the following. The user types a query / topic description in keywords (for example, "merge" or "merger"). Then tables will be created automatically in several minutes, rather than in a month of human labor. These tables are expected to show information about the salient relations for the topic.

Figure 1 describes the components and how this system works. There are six major components in the system. We will briefly describe each component and how the data is processed; then, in the next section, four important components will be described in more detail.

Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 731–738,
Sydney, July 2006. ©2006 Association for Computational Linguistics

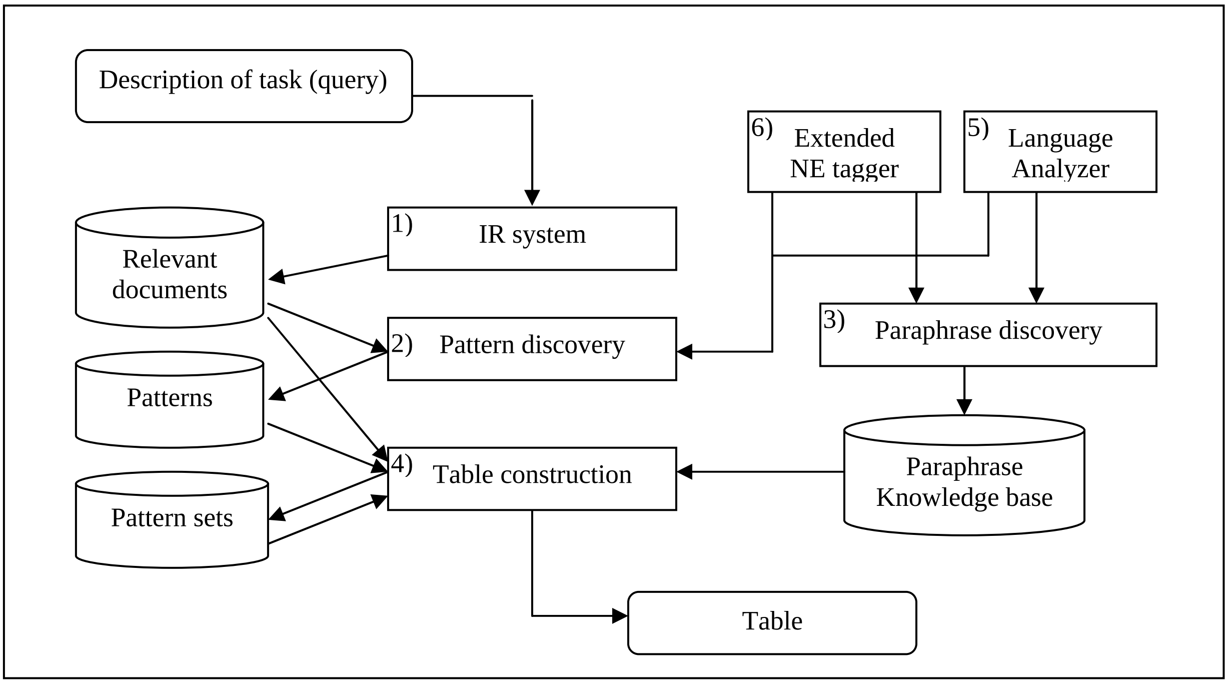


Figure 1. System overview

1) <u>IR system</u>: Based on the query given by the user, it retrieves relevant documents from the document database. We used a simple TF/IDF IR system we developed.

2) <u>Pattern discovery</u>: First, the texts in the retrieved documents are analyzed using a POS tagger, a dependency analyzer and an Extended NE (Named Entity) tagger, which will be described later. Then this component extracts sub-trees of dependency trees which are relatively frequent in the retrieved documents compared to the entire corpus. It counts the frequencies in the retrieved texts of all subtrees with more than a certain number of nodes and uses TF/IDF methods to score them. The top-ranking sub-trees which contain NEs will be called *patterns*, which are expected to indicate salient relationships of the topic and will be used in the later components.

3) <u>Paraphrase discovery</u>: In order to find semantic relationships between patterns, i.e. to find patterns which should be used to build the same table, we use paraphrase discovery techniques. The paraphrase discovery was conducted offline and created a paraphrase knowledge base.

4) <u>Table construction</u>: In this component, the patterns created in (2) are linked based on the paraphrase knowledge base created by (3), producing sets of patterns which are semantically equivalent. Once the sets of patterns are created, these patterns are applied to the documents retrieved by the IR system (1). The matched patterns pull out the entity instances and these entities are aligned to build the final tables.

5) <u>Language analyzers</u>: We use a POS tagger and a dependency analyzer to analyze the text. The analyzed texts are used in pattern discovery and paraphrase discovery.

6) <u>Extended NE tagger</u>: Most of the participants in events are likely to be Named Entities. However, the traditional NE categories are not sufficient to cover most participants of various events. For example, the standard MUC's 7 NE categories (i.e. person, location, organization, percent, money, time and date) miss product names (e.g. Windows XP, Boeing 747), event names (Olympics, World War II), numerical expressions other than monetary expressions, etc. We used the Extended NE categories with 140 categories and a tagger based on the categories.

3 Details of Components

In this section, four important components will be described in detail. Prior work related to each component is explained and the techniques used in our system are presented.

3.1 Pattern Discovery

The pattern discovery component is responsible for discovering salient patterns for the topic. The patterns will be extracted from the documents relevant to the topic which are gathered by an IR system.

Several unsupervised pattern discovery techniques have been proposed, e.g. (Riloff 96), (Agichtein and Gravano 00) and (Yangarber et al. 00). Most recently we (Sudo et al. 03) proposed a method which is triggered by a user query to discover important patterns fully automatically. In this work, three different representation models for IE patterns were compared, and the sub-tree model was found more effective compared to the predicate-argument model and the chain model. In the sub-tree model, any connected part of a dependency tree for a sentence can be considered as a pattern. As it counts all possible sub-trees from all sentences in the retrieved documents, the computation is very expensive. This problem was solved by requiring that the sub-trees contain a predicate (verb) and restricting the number of nodes. It was implemented using the sub-tree counting algorithm proposed by (Abe et al. 02). The patterns are scored based on the relative frequency of the pattern in the retrieved documents (f_r) and in the entire corpus (f_{all}). The formula uses the TF/IDF idea (Formula 1). The system ignores very frequent patterns, as those patterns are so common that they are not likely to be important to any particular topic, and also very rare patterns, as most of those patterns are noise.

$$score(t : subtree) = \frac{f_r(t)}{\log(f_{all}(t) + c)} \qquad (1)$$

The scoring function sorts all patterns which contain at least one extended NE and the top 100 patterns are selected for later processing. Figure 2 shows examples of the discovered patterns for the "merger and acquisition" topic. Chunks are shown in brackets and extended NEs are shown in upper case words. (COM means "company" and MNY means "money")

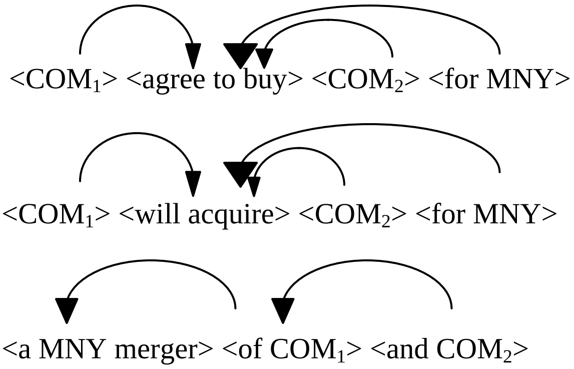


Figure 2. Pattern examples

3.2 Paraphrase Discovery

The role of the paraphrase discovery component is to link the patterns which mean the same thing for the task. Recently there has been a growing amount of research on automatic paraphrase discovery. For example, (Barzilay 01) proposed a method to extract paraphrases from parallel translations derived from one original document. We proposed to find paraphrases from multiple newspapers reporting the same event, using shared Named Entities to align the phrases (Shinyama et al. 02). We also proposed a method to find paraphrases in the context of two Named Entity instances in a large un-annotated corpus (Sekine 05). The phrases connecting two NEs are grouped based on two types of evidence. One is the identity of the NE instance pairs, as multiple instances of the same NE pair (e.g. Yahoo! and Overture) are likely to refer to the same relationship (e.g. acquisition). The other type of evidence is the keywords in the phrase. If we gather a lot of phrases connecting NE's of the same two NE types (e.g. company and company), we can cluster these phrases and find some typical expressions (e.g. merge, acquisition, buy). The phrases are clustered based on these two types of evidence and sets of paraphrases are created.

Basically, we used the paraphrases found by the approach mentioned above. For example, the expressions in Figure 2 are identified as paraphrases by this method; so these three patterns will be placed in the same pattern set.

Note that there is an alternative method of paraphrase discovery, using a hand crafted synonym dictionary like WordNet (WordNet Home page). However, we found that the coverage of WordNet for a particular topic is not sufficient. For example, no synset covers any combinations of the main words in Figure 2, namely “buy”, “acquire” and “merger”. Furthermore, even if these words are found as synonyms, there is the additional task of linking expressions. For example, if one of the expressions is “reject the merger”, it shouldn’t be a paraphrase of “acquire”.

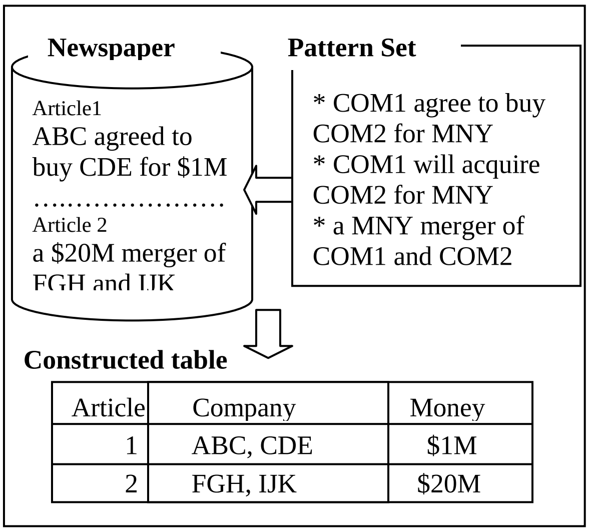
3.3 Extended NE tagging

Named Entities (NE) were first introduced by the MUC evaluations (Grishman and Sundheim 96). As the MUCs concentrated on business and military topics, the important entity types were limited to a few classes of names and numerical expressions. However, along with the development of Information Extraction and Question Answering technologies, people realized that there should be more and finer categories for NE. We proposed one of those extended NE sets (Sekine 02). It includes 140 hierarchical categories. For example, the categories include Company, Company group, Military, Government, Political party, and International Organization as subcategories of Organization. Also, new categories are introduced such as Vehicle, Food, Award, Religion, Language, Offense, Art and so on as subcategories of Product, as well as Event, Natural Object, Vocation, Unit, Weight, Temperature, Number of people and so on. We used a rule-based tagger developed to tag the 140 categories for this experiment.

Note that, in the proposed method, the slots of the final table will be filled in only with instances of these extended Named Entities. Most common nouns, verbs or sentences can’t be entries in the table. This is obviously a limitation of the proposed method; however, as the categories are designed to provide good coverage for a factoid type QA system, most interesting types of entities are covered by the categories.

3.4 Table Construction

Basically the table construction is done by applying the discovered patterns to the original corpus. The discovered patterns are grouped into pattern set using discovered paraphrase knowledge. Once the pattern sets are built, a table is created for each pattern set. We gather all NE instances matched by one of the patterns in the set. These instances are put in the same column of the table for the pattern set. When creating tables, we impose some restrictions in order to reduce the number of meaningless tables and to gather the same relations in one table. We require columns to have at least three filled instances and delete tables with fewer than three rows. These thresholds are empirically determined using training data.



Article	Company	Money
1	ABC, CDE	$1M
2	FGH, IJK	$20M

Figure 3. Table Construction

4 Experiments

4.1 Data and Processing

We conducted the experiments using the 1995 New York Times as the corpus. The queries used for system development and threshold tuning were created by the authors, while queries based on the set of event types in the ACE extraction evaluations were used for testing. A total of 31 test queries were used; we discarded several queries which were ambiguous or uncertain. The test queries were derived from the example sentences for each event type in the ACE guidelines. Examples of queries are shown in the Appendix.

At the moment, the whole process takes about 15 minutes on average for each query on a Pentium 2.80GHz processor running Linux. The corpus was analyzed in advance by a POS tagger, NE tagger and dependency analyzer. The processing and counting of sub-trees takes the majority (more than 90%) of the time. We believe we can easily make it faster by programming techniques, for example, using distributed computing.

4.2 Result and Evaluation

Out of 31 queries, the system is unable to build any tables for 11 queries. The major reason is that the IR component can't find enough newspaper articles on the topic. It retrieved only a few articles for topics like "born", "divorce" or "injure" from The New York Times. For the moment, we will focus on the 20 queries for which tables were built. The Appendix shows some examples of queries and the generated tables. In total, 127 tables are created for the 20 topics, with one to thirteen tables for each topic. The number of columns in a table ranges from 2 to 10, including the document ID column, and the average number of columns is 3.0. The number of rows in a table range from 3 to 125, and the average number of rows is 16.9. The created tables are usually not fully filled; the average rate is 20.0%.

In order to measure the potential and the usefulness of the proposed method, we evaluate the result based on three measures: usefulness, argument role coverage, and correctness. For the usefulness evaluation, we manually reviewed the tables to determine whether a useful table is included or not. This is inevitably subjective, as the user does not specify in advance what table rows and columns are expected. We asked a subject to judge usefulness in three grades; A) very useful – for the query, many people might want to use this table for the further investigation of the topic, B) useful – at least, for some purpose, some people might want to use this table for further investigation and C) not useful – no one will be interested in using this table for further investigation. The argument role coverage measures the percentage of the roles specified for each ACE event type which appeared as a column in one or more of the created tables for that event type. The correctness was measured based on whether a row of a table reflects the correct information. As it is impossible to evaluate all the data, the evaluation data are selected randomly.

Table 1 shows the usefulness evaluation result. Out of 20 topics, two topics are judged very useful and twelve are judged useful. The very useful topics are "fine" (Q4 in the appendix) and "acquit" (not shown in the appendix). Compared to the results in the 'useful' category, the tables for these two topics have more slots filled and the NE types of the fillers have fewer mistakes. The topics in the "not useful" category are "appeal", "execute", "fired", "pardon", "release" and "trial". These are again topics with very few relevant articles. By increasing the corpus size or improving the IR component, we may be able to improve the performance for these topics. The majority category, "useful", has 12 topics. Five of them can be found in the appendix (all those besides Q4). For these topics, the number of relevant articles in the corpus is relatively high and interesting relations are found. The examples in the appendix are selected from larger tables with many columns. Although there are columns that cannot be filled for every event instance, we found that the more columns that are filled in, the more useful and interesting the information is.

Table 1. Usefulness evaluation result

Evaluation	Number of topics
Very useful	2
Useful	12
Not useful	6

For the 14 "very useful" and "useful" topics, the role coverage was measured. Some of the roles in the ACE task can be filled by different types of Named Entities, for example, the "defendant" of a "sentence" event can be a Person, Organization or GPE. However, the system creates tables based on NE types; e.g. for the "sentence" event, a Person column is created, in which most of the fillers are defendants. In such cases, we regard the column as covering the role. Out of 63 roles for the 14 event types, 38 are found in the created tables, for a role coverage of 60.3%. Note that, by lowering the thresholds, the coverage can be increased to as much as 90% (some roles can't be found because of Extended NE limitations or the rare appearance of roles) but with some sacrifice of precision.

Table 2 shows the correctness evaluation results. We randomly select 100 table rows among the topics which were judged "very useful" or "useful", and determine the correctness of the information by reading the newspaper articles the information was extracted from. Out of 100 rows, 84 rows have correct information in all slots. 4

rows have some incorrect information in some of the columns, and 12 contain wrong information. Most errors are due to NE tagging errors (11 NE errors out of 16 errors). These errors include instances of people which are tagged as other categories, and so on. Also, by looking at the actual articles, we found that co-reference resolution could help to fill in more information. Because the important information is repeatedly mentioned in newspaper articles, referential expressions are often used. For example, in a sentence "In 1968 he was elected mayor of Indianapolis.", we could not extract "he" at the moment. We plan to add coreference resolution in the near future. Other sources of error include:

- The role of the entity is confused, i.e. victim and murderer
- Different kinds of events are found in one table, e.g., the victory of Jack Nicklaus was found in the political election query (as both of them use terms like "win")
- An unrelated but often collocate entity was included. For example, Year period expressions are found in "fine" events, as there are many expressions like "He was sentenced 3 years and fined $1,000".

Table 2. Correctness evaluation result

Evaluation	Number of rows
Correct	84
Partially correct	4
Incorrect	12

5 Related Work

As far as the authors know, there is no system similar to ODIE. Several methods have been proposed to produce IE patterns automatically to facilitate IE knowledge creation, as is described in Section 3.1. But those are not targeting the fully automatic creation of a complete IE system for a new topic.

There exists another strategy to extend the range of IE systems. It involves trying to cover a wide variety of topics with a large inventory of relations and events. It is not certain if there are only a limited number of topics in the world, but there are a limited number of high-interest topics, so this may be a reasonable solution from an engineering point of view. This line of research was first proposed by (Aone and Ramos-Santacruz 00) and the ACE evaluations of event detection follow this line (ACE Home Page).

An unsupervised learning method has been applied to a more restricted IE task, Relation Discovery. (Hasegawa et al. 2004) used large corpora and an Extended Named Entity tagger to find novel relations and their participants. However, the results are limited to a pair of participants and because of the nature of the procedure, the discovered relations are static relations like a country and its presidents rather than events.

Topic-oriented summarization, currently pursued by the DUC evaluations (DUC Home Page), is also closely related. The systems are trying to create summaries based on the specified topic for a manually prepared set of documents. In this case, if the result is suitable to present in table format, it can be handled by ODIE. Our previous study (Sekine and Nobata 03) found that about one third of randomly constructed similar newspaper article clusters are well-suited to be presented in table format, and another one third of the clusters can be acceptably expressed in table format. This suggests there is a big potential where an ODIE-type system can be beneficial.

6 Future Work

We demonstrated a new paradigm of Information Extraction technology and showed the potential of this method. However, there are problems to be solved to advance the technology. One of them is the coverage of the extracted information. Although we have created useful tables for some topics, there are event instances which are not found. This problem is mostly due to the inadequate performance of the language analyzers (information retrieval component, dependency analyzer or Extended NE tagger) and the lack of a coreference analyzer. Even though there are possible applications with limited coverage, it will be essential to enhance these components and add coreference in order to increase coverage. Also, there are basic domain limitations. We made the system "on-demand" for any topic, but currently only within regular news domains. As configured, the system would not work on other domains such as a medical, legal, or patent domain, mainly due to the design of the extended NE hierarchy. While specific hierarchies could be incorporated for new domains, it will also be desirable to integrate bootstrapping techniques for rapid incremental additions to the hierarchy. Also at the moment, table column labels are simply Extended

NE categories, and do not indicate the role. We would like to investigate this problem in the future.

7 Conclusion

In this paper, we proposed "On-demand Information Extraction (ODIE)". It is a system which automatically identifies the most salient structures and extracts the information on whatever topic the user demands. It relies on recent advances in NLP technologies; unsupervised learning and several advanced NLP analyzers. Although it is at a preliminary stage, we developed a prototype system which has created useful tables for many topics and demonstrates the feasibility of this approach.

8 Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency under Contract HR0011-06-C-0023 and by the National Science Foundation under Grant IIS-0325657. This paper does not necessarily reflect the position of the U.S. Government.

We would like to thank Prof. Ralph Grishman, Dr. Kiyoshi Sudo, Dr. Chikashi Nobata, Mr. Takaaki Hasegawa, Mr. Koji Murakami and Mr. Yusuke Shinyama for useful comments, discussion.

References

ACE Home Page:
http://www.ldc.upenn.edu/Projects/ace

DUC Home Page: http://duc.nist.gov

WordNet Home Page: http://wordnet.princeton.edu/

Kenji Abe, Shinji Kawasone, Tatsuya Asai, Hiroki Arimura and Setsuo Arikawa. 2002. "Optimized Substructure Discovery for Semi-structured Data". In Proceedings of the 6th European Conference on Principles and Practice of Knowledge in Database (PKDD-02)

Chinatsu Aone; Mila Ramos-Santacruz. 2000. "REES: A Large-Scale Relation and Event Extraction System" In Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-00)

Eugene Agichtein and L. Gravano. 2000. "Snowball: Extracting Relations from Large Plaintext Collectionss". In Proceedings of the 5th ACM International Conference on Digital Libraries (DL-00)

Regina Barzilay and Kathleen McKeown. 2001. "Extracting Paraphrases from a Parallel Corpus. In Proceedings of the Annual Meeting of Association of Computational Linguistics/ and European Chapter of Association of Computational Linguistics (ACL/EACL-01)

Ralph Grishman and Beth Sundheim.1996. "Message Understanding Conference - 6: A Brief History", in Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)

Takaaki Hasegawa, Satoshi Sekine and Ralph Grishman 2004. "Discovering Relations among Named Entities from Large Corpora", In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL-04)

Ellen Riloff. 1996. "Automatically Generating Extraction Patterns from Untagged Text". In Proceedings of Thirteen National Conference on Artificial Intelligence (AAAI-96)

Satoshi Sekine, Kiyoshi Sudo and Chikashi Nobata. 2002 "Extended Named Entity Hierarchy" In Proceefings of the third International Conference on Language Resources and Evaluation (LREC-02)

Satoshi Sekine and Chikashi Nobata. 2003. "A survey for Multi-Document Summarization" In the proceedings of Text Summarization Workshop.

Satoshi Sekine. 2005. "Automatic Paraphrase Discovery based on Context and Keywords between NE Pairs". In Proceedings of International Workshop on Paraphrase (IWP-05)

Yusuke Shinyama, Satoshi Sekine and Kiyoshi Sudo. 2002. "Automatic Paraphrase Acquisition from News Articles". In Proceedings of the Human Language Technology Conference (HLT-02)

Kiyoshi Sudo, Satsohi Sekine and Ralph Grishman. 2003. "An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition". In Proceedings of the Annual Meeting of Association of Computational Linguistics (ACL-03)

Roman Yangarber, Ralph Grishman, Pasi Tapanainen and Silja Huttunen. 2000. "Unsupervised Discovery of Scenario-Level Patterns for Information Extraction". In Proceedings of 18th International Conference on Computational Linguistics (COLING-00)

Appendix: Sample queries and tables
(Note that this is only a part of created tables)

Q1: acquire, acquisition, merge, merger, buy purchase

docid	MONEY	COMPANY	DATE
nyt950714.0324	About $3 billion	PNC Bank Corp., Midlantic Corp.	
nyt950831.0485	$900 million	Ceridian Corp., Comdata Holdings Corp.	Last week
nyt950909.0449	About $1.6 billion	Bank South Corp	
nyt951010.0389	$3.1 billion	CoreStates Financial Corp.	
nyt951113.0483	$286 million	Potash Corp.	Last month
nyt951113.0483	$400 million	Chemicals Inc.	Last year

Q2: convict, guilty

docid	PERSON	DATE	AGE
nyt950207.0001	Fleiss	Dec. 2	28
nyt950327.0402	Gerald_Amirault	1986	41
nyt950720.0145	Hedayat_Eslaminia	1988	
nyt950731.0138	James McNally, James Johnson Bey, Jose Prieto, Patterson	1993, 1991, this year, 1984	
nyt951229.0525	Kane	Last year	

Q3: elect

Docid	POSITION TITLE	PERSON	DATE
nyt950404.0197	president	Havel	Dec. 29, 1989
nyt950916.0222	president	Ronald Reagan	1980
nyt951120.0355	president	Aleksander Kwasniewski	

Q4: fine

Docid	PERSON	MONEY	DATE
nyt950420.0056	Van Halen	$1,000	
nyt950525.0024	Derek Meredith	$300	
nyt950704.0016	Tarango	At least $15,500	
nyt951025.0501	Hamilton	$12,000	This week
nyt951209.0115	Wheatley	Approximately $2,000	

Q5: arrest jail incarcerate imprison

Docid	PERSON	YEAR PERIOD
nyt950817.0544	Nguyen Tan Tri	Four years
nyt951018.0762	Wolf	Six years
nyt951218.0091	Carlos Mendoza-Lugo	One year

Q6: sentence

Docid	PERSON	YEAR PERIOD
nyt950412.0448	Mitchell Antar	Four years
nyt950421.0509	MacDonald	14 years
nyt950622.0512	Aramony	Three years
nyt950814.0106	Obasanjo	25 years