System Demonstration of On-Demand Information Extraction

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

Akira Oda [1)]
Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho,
Toyohashi, Aichi 441-3580 Japan
oda@ss.ics.tut.ac.jp

Abstract

In this paper, we will describe ODIE, the On-Demand Information Extraction system. Given a user's query, the system will produce tables of the salient information about the topic in structured form. It produces the tables in less than one minute without any knowledge engineering by hand, i.e. pattern creation or paraphrase knowledge creation, which was the largest obstacle in traditional IE. This demonstration is based on the idea and technologies reported in (Sekine 06). A substantial speed-up over the previous system (which required about 15 minutes to analyze one year of newspaper) was achieved through a new approach to handling pattern candidates; now less than one minute is required when using 11 years of newspaper corpus. In addition, functionality was added to facilitate investigation of the extracted information.

1 Introduction

The goal of information extraction (IE) is to extract information about events in structured form from unstructured texts. In traditional IE, a great deal of knowledge for the systems must be coded by hand in advance. For example, in the later MUC evaluations, system developers spent one month for the knowledge engineering to customize the system to the given test topic. Improving portability is necessary to make Information Extraction technology useful for real users and, we believe, lead to a breakthrough for the application of the technology.

1) This work was conducted when the first author was a junior research scientist at New York University.

Sekine (Sekine 06) proposed '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 is created on top of a range of basic language analysis tools, including POS taggers, dependency analyzers, and extended Named Entity taggers. This paper describes the demonstration system of the new IE paradigm, which incorporates some new ideas to make the system practical.

2 Algorithm Overview

We will present an overview of the algorithm in this section. The details can be found in (Sekine 06).

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

There are six major components in the system.

1) IR system: 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) Pattern discovery: The texts are analyzed using a POS tagger, a dependency analyzer and an Extended Named Entity (ENE) tagger, which will be explained in (5). Then sub-trees of dependency trees which are relatively frequent in the retrieved documents compared to the entire corpus are identified. The sub-trees to be used must satisfy some restrictions, including having

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 17–20,
Prague, June 2007.

between 2 and 6 nodes, having a predicate or nominalization as the head of the sub-tree, and having at least one NE. We introduced upper and lower frequency bounds for the sub-trees to be used, as we found the medium frequency sub-trees to be the most useful and least noisy. We compute a score for each pattern based on its frequency in the retrieved documents and in the entire collection. The top scoring sub-trees will be called *patterns*, which are expected to indicate salient relationships of the topic and which will be used in the later components. We pre-compute such information as much as possible in order to enable usably prompt response to queries.

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 lexical knowledge such as WordNet and paraphrase discovery techniques. The paraphrase discovery was conducted off-line 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 from the sentences and these entities are aligned to build the final tables.

5) <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 with 140 categories and a tagger developed for these categories.

3 Speed-enhancing technology

The largest computational load in this system is the extraction and scoring of the topic-relevant sub-trees. In the previous system, 1,000 top-scoring sub-trees are extracted from all possible (on the order of hundreds of thousands) sub-trees in the top 200 relevant articles. This computation took about 14 minutes out of the total 15 minutes of the entire process. The difficulty is that the set of top articles is not predictable, as the input is arbitrary and hence the list of sub-trees is not predictable, too. Although a state-of-the-art tree mining algorithm (Abe et al. 02) was used, the computation is still impracticable for a real system.

The solution we propose in this paper is to pre-compute all possibly useful sub-trees in order to reduce runtime. We enumerate all possible sub-trees in the entire corpus and store them in a database with frequency and location information. To reduce the size of the database, we filter the patterns, keeping only those satisfying the constraints on frequency and existence of predicate and named entities. However, it is still a big challenge, because in this system, we use 11 years of newspaper (AQUAINT corpus, with duplicate articles removed) instead of the one year of newspaper (New York Times 95) used in the previous system. With this idea, the response time of the demonstration system is reduced significantly.

The statistics of the corpus and sub-trees are as follows. The entire corpus includes 1,031,124 articles and 24,953,026 sentences. The frequency thresholds for sub-trees to be used is set to more than 10 and less than 10,000; i.e. sub-trees of those frequencies in the corpus are expected to contain most of the salient relationships with minimum noise. The sub-trees with frequency less than 11 account for a very large portion of the data; 97.5% of types and 66.3% of instances, as shown in Table 1. The sub-trees of frequency of 10,001 or more are relatively small; only 76 kinds and only 2.5% of the instances.

Frequency	10,001 or more	10,000-11	10 or less
# of type	76	975,269	38,158,887
	~0.0%	2.5%	97.5%
# of instance	2,313,347	29,257,437	62,097,271
	2.5%	31.2%	66.3%

Table 1. Frequency of sub-trees

We assign ID numbers to all 1 million sub-trees and 25 million sentences and those are mutually linked in a database. Also, 60 million NE occurrences in the sub-trees are identified and linked to the sub-tree and sentence IDs. In the process, the sentences found by the IR component are identified. Then the sub-trees linked to those sentences are gathered and the scores are calculated. Those processes can be done by manipulation of the database in a very short time. The top sub-trees are used to create the output tables using NE occurrence IDs linked to the sub-trees and sentences.

4 A Demonstration

In this section, a simple demonstration scenario is presented with an example. Figure 1 shows the initial page. The user types in any keywords in the query box. This can be anything, but as a traditional IR system is used for the search, the keywords have to include expressions which are normally used in relevant documents. Examples of such keywords are “merge, acquisition, purchase”, “meet, meeting, summit” and “elect, election”, which were derived from ACE event types.

Then, normally within one minute, the system produces tables, such as those shown in Figure 2. All extracted tables are listed. Each table contains sentence ID, document ID and information extracted from the sentence. Some cells are empty if the information can’t be extracted.

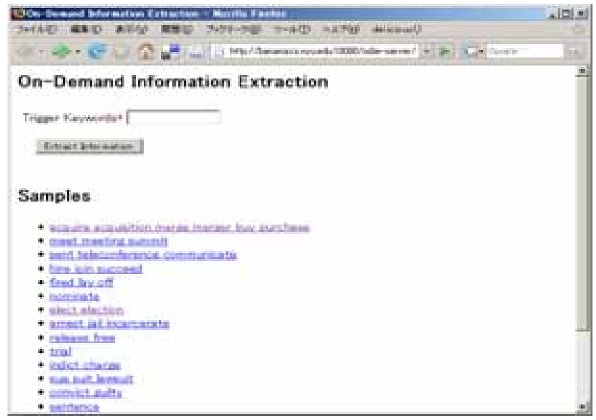

Figure 1. Screenshot of the initial page

5 Evaluation

The evaluation was conducted using scenarios based on 20 of the ACE event types. The accuracy of the extracted information was evaluated by judges for 100 rows selected at random. Of these rows, 66 were judged to be on target and correct. Another 10 were judged to be correct and related to the topic, but did not include the essential information of the topic. The remaining 24 included NE errors and totally irrelevant information (in some cases due to word sense ambiguity; e.g. “fine” weather vs.“fine” as a financial penalty).

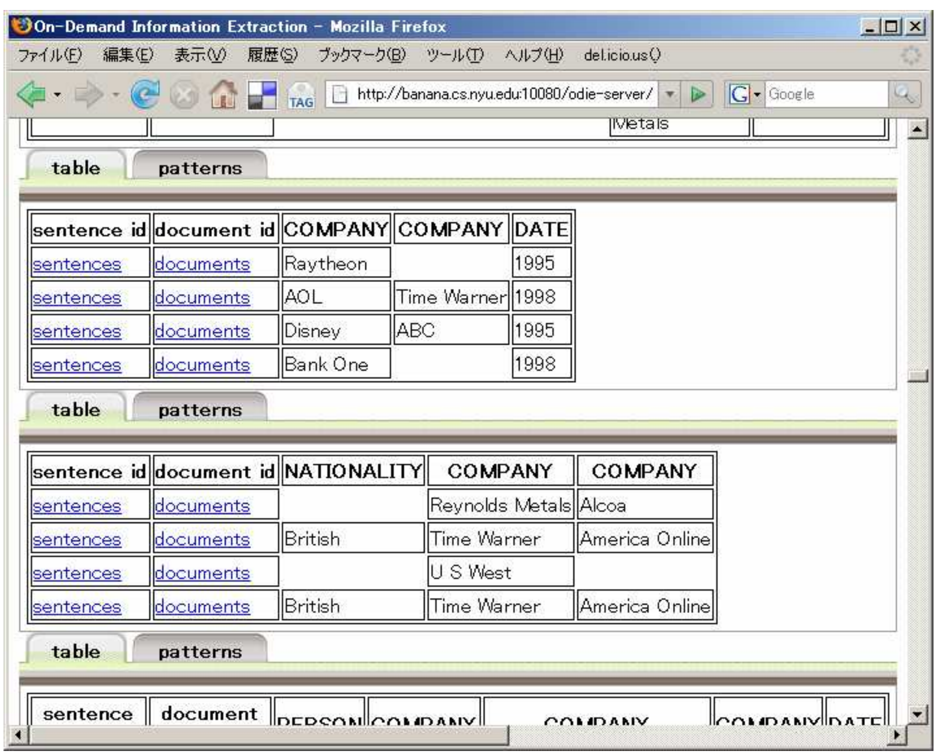

Figure 2. Screenshot of produced tables

6 Other Functionality

Functionality is provided to facilitate the user's access to the extracted information. Figure 3 shows a screenshot of the document from which the information was extracted. Also the patterns used to create each table can be found by clicking the tab "patterns" (shown in Figure 4). This could help the user to understand the nature of the table. The information includes the frequency of the pattern in the retrieved documents and in the entire corpus, and the pattern's score.

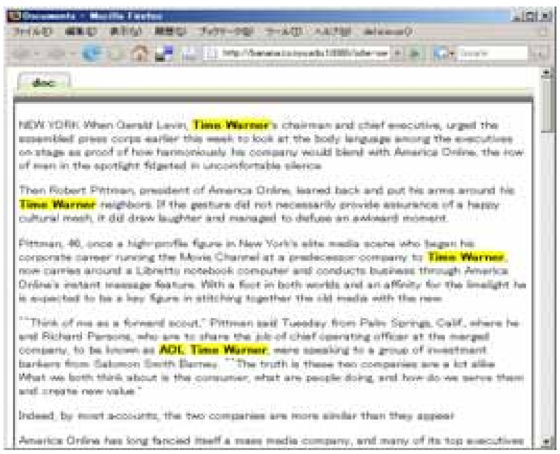

Figure 3. Screenshot of document view

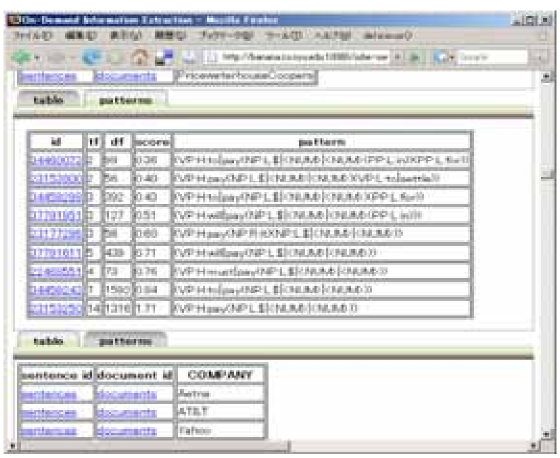

Figure 4. Screenshot of pattern information

7 Future Work

We demonstrated the On-Demand Information Extraction system, which provides usable response time for a large corpus. We still have several improvements to be made in the future. One is to include more advanced and accurate natural language technologies to improve the accuracy and coverage. For example, we did not use a coreference analyzer, and hence information which was expressed using pronouns or other anaphoric expressions can not be extracted. Also, more semantic knowledge including synonym, paraphrase or inference knowledge should be included. The output table has to be more clearly organized. In particular, we can't display role information as column headings. The keyword input requirement is very inconvenient. For good performance, the current system requires several keywords occurring in relevant documents; this is an obvious limitation. On the other hand, there are systems which don't need any user input to create the structured information (Banko et al. 07) (Shinyama and Sekine 06). The latter system tries to identify all possible structural relations from a large set of unstructured documents. However, the user's information needs are not predictable and the question of whether we can create structured information for all possible needs is still a big challenge.

Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency as part of the Translingual Information Detection, Extraction and Summarization (TIDES) program, under Grant N66001-001-1-8917 from the Space and Naval Warfare Systems Center, San Diego, and by the National Science Foundation under Grant IIS-00325657. This paper does not necessarily reflect the position of the U.S. Government. We would like to thank our colleagues at New York University, who provided useful suggestions and discussions, including, Prof. Ralph Grishman and Mr. Yusuke Shinyama.

References

Kenji Abe, Shinji Kawasone, Tatsuya Asai, Hiroki Arimura and Setsuo Arikawa. 2002. "Optimized Substructure Discovery for Semi-structured Data". PKDD-02.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead and Oren Etzioni. 2007. "Open Information Extraction from Web". IJCAI-07.

Satoshi Sekine. 2006. "On-Demand Information Extraction". COLING-ACL-06.

Yusuke Shinyama and Satoshi Sekine, 2006. "Preemptive Information Extraction using Unrestricted Relation Discovery". HLT-NAACL-2006.