# Unsupervised Learning of mDTD-based Web Information Extraction Patterns

**Dongseok Kim, Hanmin Jung, and Gary Geunbae Lee**
Pohang University of Science & Technology
San31, Hyoja, Pohang, 790-784, Korea
{dskim, jhm, gblee}@nlp.postech.ac.kr

## ABSTRACT

This paper presents a new information extraction pattern, called modified Document Type Definition (mDTD), which relies on analytical interpretation to identify target information from the textual fragments of Web documents. We develop two major extensions from conventional SGML DTD: Concerning syntax, we introduce an extended content model with type-specific operators and keywords. Resulting mDTD can represent HTML structures and Web information extraction targets. The design goal of mDTD is to overcome domain portability with minimum human intervention while maintaining a high extraction performance. The human experts compose an mDTD as seed rules with which our system automatically extracts the necessary training instances from structured documents on the Web. These extracted instances are used as inputs to SmL (Sequential mDTD Learner) which generates new mDTD rules based on the part-of-speech tags and lexical similarity features. Therefore, for learning, no hand-tagged corpus is required.

## Keywords

Web Data Mining, Information Extraction

## 1. INTRODUCTION

Recently, some researchers have addressed the problem of generating extraction patterns automatically, without human-annotated corpus, which is more desirable in practical-level applications. The DIPRE system [1] uses a bootstrapping method to find patterns and relations from Web documents without pre-annotated data. The process is initiated with small samples as seeds in some given relations, such as a relation of (author, title) pairs from the WWW. Similar to DIPRE, [9] uses an automatic bootstrapping method to find patterns for name classification. However, this method requires a named-entity tagger to mark all the instances of people's names, companies, and locations, and a parser to extract all the clauses from each document.

Our research improves these automatic bootstrapping ideas without pre-annotated corpus. We focus more on declarative-style knowledge, which can be extended with human interaction for practical-level performance in a real deployed commercial system. We present a new extraction method to combine declarative DTD-style extraction patterns and a machine learning algorithm without annotated corpus to generate the extraction patterns. Our approach differs from previous studies in several respects: First, we propose DTD-style declarative rules, called mDTD, as extraction patterns. The mDTD is able to express the page structure on Web sites organized with structured HTML documents. If necessary, mDTD can be edited by an application manager with a knowledge of DTD programming, because mDTD is encoded in normal texts after machine learning is completed. This provides a flexible model which can be easily modified to include additional sources of evidence for commercial-level performance. Second, structured (tabular form) documents automatically acquired from the Web are directly used as the input of machine learning, instead of named-entity tagged or parsed corpus.

## 2. mDTD PATTERN REPRESENTATION

The DTD concept has generally been used for markup languages, such as SGML, XML, and HTML. In these documents, DTD is usually located in one or more external files, and defines what elements belong to this document type [5]. Using DTD, SGML documents can encode the elements included in the documents, and also parse those elements that appear in the document. mDTD is used to encode and decode the textual elements of the extraction target. In the learning phase, mDTD rules are learned and added to the set of seed mDTD's for the extraction task. In the extracting phase, a learned mDTD rule set is used as extraction patterns to identify the elements in HTML documents from Web sites. Human experts write “Seed mDTD” for a given domain only once, and remainder extraction procedures are executed all automatically without human intervention.

A complete mDTD rule is made up of five components as in the following syntax:

*rule => <! keyword name opt (content) occurence_op action >*

, where *<!* and > are the rule declaration open and close symbols. In the above mDTD rule syntax, *keyword* specifies the types of rules.

The *name* designates a rule identifier, and *opt* means an option which describes the optional conditions of the rule. Some rules can be rewritten with *content* which complements the *name* object. The *occurrence_op* is an occurrence operator about *content* and *action*, which specifies the parameters and references of the rule.

Seed mDTD rules are composed by persons who have only slight knowledge about domain or shopping items.

*<!TARGET tItem - - ((startHtag)*, Tv, (endHtag)*)>*

The above rule means *tItem*, which is an extraction target item and the content has three components: *startHtag*, *Tv*, and *endHtag*. The first and last component may occur zero or more times, but the *Tv* component must occur exactly once to satisfy the *tItem* target. The seed mDTD rules are used to extract instances from structured documents, which are automatically gathered by using a Web robot. The extracted instances are directly used as input of the mDTD rule learner. If the instances are extracted correctly, they become the positive examples of the learner, otherwise, become the negative examples.

## 3. SmL: SEQUENTIAL mDTD LEARNER

The key idea of SmL learning is to prepare input examples for machine learner without human intervention. The examples come from structured documents by an automatic extraction process using seed mDTD which is the only part prepared by human experts. In our extraction learning system, SmL, we consider a family of algorithms for inductive learning, based on the sequential strategy; that is, they learn one rule, remove the data covered, and then iterate the one-rule-learning-discarding process. The SmL learner for mDTD is based on the previous CN2 algorithm [4] except for the measure of performance evaluation. Figure 1 shows the SmL algorithm which uses a general to specific beam search. In CN2, the performance measure for the generated rule is an information gain, similar to FOIL [7], while, in SmL, lexical similarity and coverage rate for examples are newly introduced as a measure in order to efficiently process the heterogeneous textual data in Web sites. The set of positive examples for learning is simply a set of field instances automatically extracted from structured documents. If the generated classes cover all the positive examples, the SmL algorithm is terminated. Each element of the classes makes two mDTD rules: one for connection between the mDTD nodes, and the other for representation of lexical similarity. In addition, SmL finds a common POS (part-of-speech) tag sequences from the generated classes using a POS tagging system [3]. These sequences are transformed into the mDTD rules with the same level of lexical similarity rules.

```
SmLearning( all_examples )
        let P = part_of_speech_tagger( all_examples )
        let rule_set = {}

        until P is empty do
                generate class by SmLForOneClass( P )
                find common part of speech tag sequence, postag, from class
                transform class and postag into mDTD rules
                add rules to rule_set
                remove from P all examples covered by class
        return rule_set

SmLForOneClass( P )
        let max_class = {}
        let n = | P |
        for ( n by n )
                class = FindMaxSimilarity(P), to find max similarity count
                max_class = MAX(max_class, class)
        return max_class
```

**[Figure 1] SmL sequential learning algorithm**

## 4. EXPERIMENTAL RESULTS

The Web documents are collected from about 80 Web shopping sites in Korean and English, with total of 410 documents, including 160 structured and 250 semi-structured ones. The performance of our system is measured, slot by slot, on the test set of 130 Korean and 120 English semi-structured documents. We also compared the previous state-of-the-art extraction systems, BWI [6], RAPIER [2], and WHISK

[8] with our SmLWeb. We choose slots with similar extraction tasks in each system for balanced comparison. Table 2 shows the comparison results, where SmLWeb shows reasonable performance compared with some other extraction systems in many different slots even if SmLWeb is an unsupervised training system with no human-annotated.

| Document | Slot name | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Item | | | Manufacturer | | | Model | | |
| | R | P | F1 | R | P | F1 | R | P | F1 |
| Korean | 83.2 | 82.1 | 82.6 | 88.6 | 88.3 | 88.5 | 93.7 | 96.9 | 95.3 |
| English | 72.3 | 73.5 | 72.9 | 83.1 | 77.9 | 80.4 | 89.5 | 86.2 | 87.8 |
| Average | 77.7 | 77.8 | 77.8 | 85.9 | 83.1 | 84.5 | 91.6 | 91.6 | 91.6 |

| Slot name | | | | | | | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Price | | | Specification | | | Size | | | | | |
| R | P | F1 | R | P | F1 | R | P | F1 | R | P | F1 |
| 98.3 | 99.5 | 98.9 | 70.2 | 81.7 | 75.5 | 75.6 | 99.7 | 85.9 | 84.9 | 91.3 | 87.9 |
| 89.8 | 93.1 | 91.4 | 43.6 | 69.5 | 53.6 | 72.1 | 90.9 | 80.4 | 75.1 | 81.9 | 78.4 |
| 94.1 | 96.3 | 95.2 | 56.9 | 75.6 | 64.6 | 73.9 | 95.3 | 83.2 | 80.0 | 86.6 | 83.2 |

**[Table 1] Performance of the AV domain information extraction for each slot**

| | SA(speaker)/AV(item) | | | SA(stime)/AV(model) | | | SJ(company)/AV(manufacturer) | | |
|---|---|---|---|---|---|---|---|---|---|
| | R | P | F | R | P | F | R | P | F |
| WHISK | 55.0 | 85.0 | 66.8 | 89.2 | 77.2 | 82.8 | - | - | - |
| BWI | 79.1 | 59.2 | 67.7 | 99.6 | 99.6 | 99.6 | 88.4 | 70.1 | 78.2 |
| RAPIER | 39.4 | 80.9 | 53.0 | 92.9 | 93.9 | 93.4 | 60.0 | 86.0 | 70.7 |
| **SmL** | **77.7** | **77.8** | **77.8** | **91.6** | **91.6** | **91.6** | **85.9** | **83.1** | **84.5** |

**[Table 2] Comparison with three previous extractors on the similar tasks (WHISK at 0.35 post-pruning threshold level)**

## 5. CONCLUSION

We have introduced the concept of mDTD, whose sequential learning helps to solve the domain-portability problem in Web IE systems. Unlike the previous approaches, we combined two new stable approaches: the representation of declarative rule type (called mDTD) and an automatic method to learn the mDTD rules from structured documents using a sequential SmL learner. The learned mDTD's are efficiently used as Web IE extraction patterns. The user of the SmLWeb system must write the domain-dependent part of the seed mDTD rules for the specified domain, which is similar to standard DTD programming. In the learning phase, we obtained training examples completely and automatically from structured documents, using the seed mDTD rules, without hand-tagged corpus. To adapt to new domain, users only need to rewrite a part of the seed mDTD rules to fit to the domain. The results of the SmLWeb system suggest that our combined methods are appropriate for web information extraction in E-commerce.

## REFERENCES


1. S. Brin, Extracting Patterns and Relations from the World Wide Web, *Proc. of the International Workshop on the Web and Databases*, 1998.
2. M. Califf and R. Mooney, Relational Learning of Pattern-Match Rules for Information Extraction, *Proc. of the 16th National Conference on Artificial Intelligence*, 1999.
3. J. Cha, G. Lee, and J. Lee, Generalized unknown morpheme guessing for hybrid POS tagging of Korean, Proc. *of the 6th Workshop on Very Large Corpora in Coling-Acl 98*, 1998.
4. P. Clark and T. Niblett, *The CN2 Induction Algorithm*, Machine Learning, Vol.3, 1989.
5. P. Flynn, *Understanding SGML and XML Tools - Practical programs for handling structured text*, Kluwer Academic Publishers, 1998.
6. D. Freitag and N. Kushmerick, Boosted Wrapper Induction, *Proc. of the 14th European Conference on Artificial Intelligence*, 2000.
7. J. Quinlan, *Learning Logical Definitions from Relations*, Machine Learning, Vol. 5, 1990.
8. S. Soderland, *Learning Information Extraction Rules for Semi-structured and Free Text*, Machine Learning, Vol. 34, 1999.
9. R. Yangarber and R. Grishman, Extraction Pattern Discovery through Corpus Analysis, *Proc. of the Conference on Applied Natural Language Processing ANLP-NAACL*, 2000.