A XML-BASED BOOTSTRAPPING METHOD FOR PATTERN ACQUISITION

Xingjie Zeng, Fang Li, Dongmo Zhang
Department of Computer Science, Shanghai Jiaotong University, Shanghai China, 200030

Athena I.Vakali
Department of Informatics Aristotle University of Thessaloniki Greece

Keywords: XML based applications, Bootstrapping method, Pattern Acquisition, Information Extraction

Abstract: Extensible Markup Language (XML) has been widely used as a middleware because of its flexibility. Fixed domain is one of the bottlenecks of Information Extraction (IE) technologies. In this paper we present a XML-based domain-adaptable bootstrapping method of pattern acquisition, which focuses on minimizing the cost of domain migration. The approach starts from a seed corpus with some seed patterns; extends the corpus based on the seed corpus through the Internet and acquires the new patterns from extended corpus. Positive and negative examples classified from training corpus are used to evaluate the patterns acquired. The result shows our method is a practical way in pattern acquisitions.

1 INTRODUCTION

XML plays a very important role in many Internet applications. There are three main advantages in XML based applications. First, XML can represent both structured and semi-structured information due to its flexible format. Second, it is very easy to convert other kinds of data into XML, because XML is regarded as a common standard for exchanging information. Third, all XML formats share a basic common grammar; any XML parser can parse any XML document. Many available tools for manipulating (parsing, reading, writing, translating) XML data (such as [A. Deutsch, etc. 1999]) are beneficial to different applications, thus XML-based information extraction system can deal with different data sources with different types of information expressed in different languages.

Information Extraction (IE) is a text understanding task, which involves finding facts in natural language texts, and transforming them into a logic or structured representation (e.g., a database table) according to predefined templates and patterns, such as Snowball [E. Agichtein etc. 2000], which extracts the company headquarters' location, the other extracts information about Infectious Disease Outbreaks from the web page content [R. Grishman, etc. 2002]. Even if IE seems to be now a relatively mature technology, it suffers from a number of yet unsolved problems that limit its dissemination through industrial applications, such as systems are not really portable from one domain to another. Domain migration means re-developing some resources, which is boring and time-consuming task (for example [Riloff 1995] mentions a 1500 hours development). In order to decrease the time spent on the elaboration of resources for the IE system, we use a seed corpus of domain dependent that helps defining associated resources.

In this paper, we present a domain-portable bootstrapping method to acquire domain patterns from web pages. "Domain-portable" means that the method can be conveniently used in different domains with relatively low cost of domain adaptation. The cost includes a seed corpus with some seed patterns, "key slots" for the specific domain and domain-specific Named Entities (NE) recognition system. The approach bootstraps two sets of data-points in parallel---in the dual spaces of patterns and corpus---which are statistically correlated with each other and with the topic of

Zeng X., Li F., Zhang D. and I.Vakali A. (2004).
A XML-BASED BOOTSTRAPPING METHOD FOR PATTERN ACQUISITION.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 303-308
DOI: 10.5220/0002607803030308

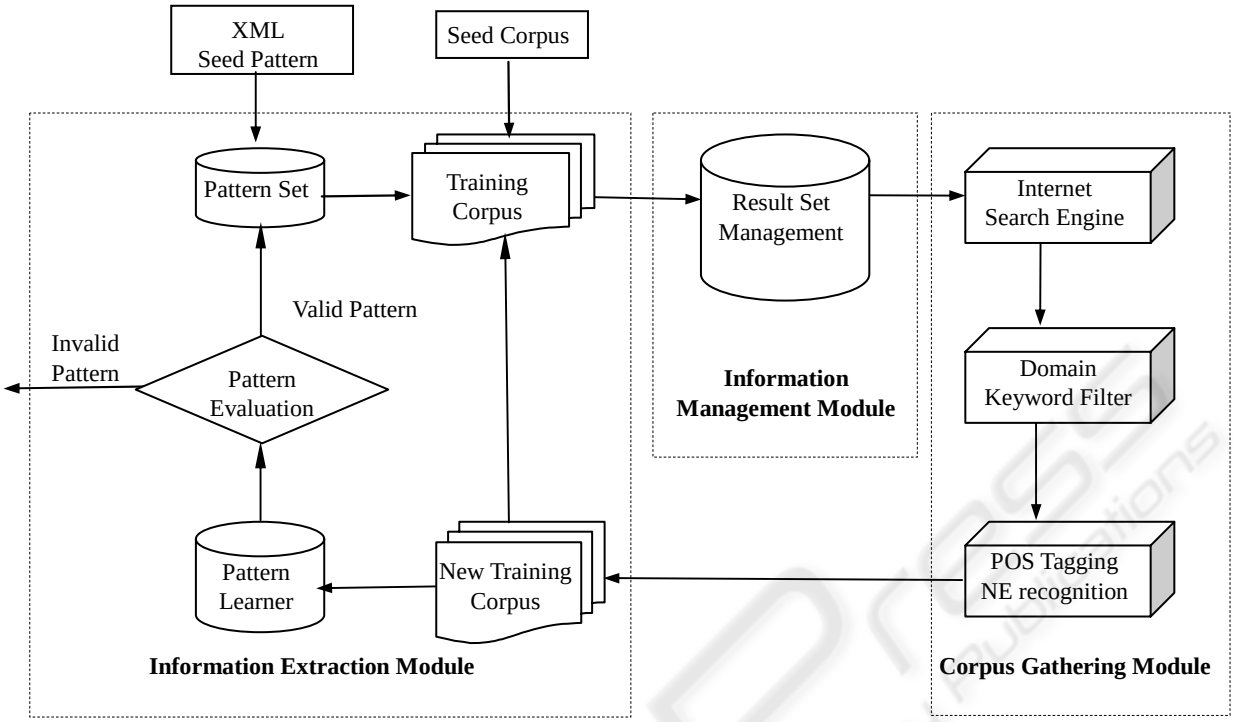

Figure 1: System framework

interest. Started from a tiny seed corpus, a huge corpus can be eventually collected and patterns can be extracted from it.

The rest of this paper is structured as follow: the system overview will be given first; then, the approach is described in section 3, 4 respectively; finally some experiments and evaluations are presented.

2 SYSTEM OVERVIEW

Figure 1 show the framework of our experimental system, which can be mainly divided into three modules: Corpus Gathering Module, Information Extraction Module, and Information Management Module.

- Corpus Gathering module:
  it searches for related web pages through the "google" search engine, extracted records provided by the Information Management Module are used as key words. Then it filters web page contents based on a simple domain keyword list, and new training corpus can be formed. After being tagged with part-of-speech (POS) and domain specific name entities (NE), the new corpus is sent to the Information Extraction Module.
- Information Extraction module:
  positive and negative examples are first identified by the classifier, then, the pattern learner draw some new patterns based on positive examples, the new patterns merge with the old patterns and evaluated by the negative examples. A special evaluation method has been developed in the system. Only those patterns, which are greater than the threshold, are regarded as the new patterns and used for next loop as old patterns.
- Information Management Module:
  based on the patterns, extracted information are stored as records in the module and managed according to key slots based on domain.

3 XML-BASED PATTERNS AND RECORDS

With XML there are already some parsers, if flat-text database is used, the whole parser have to write. When text format is changed the parser has to change as well, but the XML parser will not change, it's automatically forward-compatible. In addition, there is no need to preserve referential integrity in our application, and with no transactional support, obviously there is no obligation to use the Relational Database Management System (RDBMS).

```
<domain name=”investment”>                          // domain name
    <slot name=”investor”>                          // the pattern can extract which slot
        <pattern language="Chinese">                // pattern fits to which language
            <stuff type="FIRM-NP" />                // slot filler with proper matching NE type "FIRM-NP"
            <notext />                              // the upper and lower elements must be adjacent closely
            <substitutable>投资#v</substitutable>    // the word/phrase “invest” is in the thesaurus set
            <phrase type="MONEY" />                 // NE with "MONEY" type but not slot filler
            <notext />
            的#u                                    // a word/phrase “to” not in thesaurus set
            <notext />
            <phrase type="FIRM-NP" />
        </pattern>                                  // pattern end
        <pattern>……</pattern>                        // another pattern matching the same slot
    </slot>
    <slot><sulslot>……</subslot></slot>               //other slot/subslot
</domain>
```

The pattern can be showed as:[investor FIRM-NP]投资+[MONEY]的[FIRM-NP]

```
; “investor” means the NE [FIRM-NP] is slot filler for investor slot
; “+” means some undeclared elements can exist here
```

Figure 2: An example of pattern

Figure 2 illustrates an example of single-slot surface pattern in our test domain: investment. “Single-slot” means that the pattern can extract only one slot filler at a time. “Surface pattern” means that the pattern contains no semantic information, thus there is no requirement of semantic analysis for sentences in the corpus. In this way it is more realistic to implement the domain-adaptable method.

4 BOOTSTRAPPING METHOD FOR PATTERN ACQUISITION

We have developed a bootstrapping process for pattern acquisition that features the following four points:

1. A classifier is developed to identify positive examples and negative examples from training corpus, positive examples are made of those sentences that contain the related events, such as investment event in our system and the negative examples are made of the sentences that don’t contain the interested events.
2. Positive examples can be used to learn the extraction patterns associated with domain event. Negative examples are used to evaluate the new patterns acquired.
3. New acquired patterns with their records can be used to extract new corpus and extend the training corpus gradually.
4. Starting with a seed patterns and corpus, new acquired patterns merged with old patterns are used in the next iteration.

4.1 Document classifier

The training corpus is extracted from the Internet according to extracted records. Because of the complexity of natural languages, some sentences may not be regarded as a positive examples related to specific domain. Following two examples are searched from Internet by the same keyword set 英特尔(Intel), 投资(investment) :

Sen 1．英特尔放缓风险投资(Translation: Intel will slow down its venture investment)

Sen 2．这将使得英特尔在上海封装测试厂的投资总额由原先的1.98 亿美元增至5 亿美元。(Translation: It will increase Intel’s investment at Shanghai Capsulation and Test factory from 198 millions dollars to 500 millions dollars.)

These two sentences both contain the NE: “*英特尔(Intel)*”, however, the first sentence has nothing to do with the investment event, thus it is a negative example, while the second sentence describes an investment event and it is a positive example. In order to reduce the noise of these negative examples, we should classifier the new training corpus to maintain the high precision of the new extracted patterns.

The input of the classifier are sentences which have been POS tagged and NE recognized, the output is positive or negative examples based on a statistical model.

At the beginning, the training corpus is divided into positive examples P, negative examples N and undefined examples U.

P contains those sentences which can be

extracted by the current pattern set; N contains those sentences which includes less than τ keywords which are supposed to be related with certain event. τ is a predefined threshold, can be different value for different domain; U contains those sentences which are unknown as positive or negative.

Then, create the statistical model M_{statis} based on the P and N using the following procedure:

```
Proc
   Deleting the stop-word in each sentence S
   Dividing S into following segments using
   each [NE] as delimiter
      pre(NE1) = {P11, P12,……}
      follow(NE1) = {P21, P22,……}
      ……
      preced(NEi)     =     {Pi1,     Pi2,……}
      follow(NEi) = {P(i+1)1, P(i+1)2,……}
      ; where [NEi] is the recognized i^th NE
      ; in sentence S, Pij is the j^th term
      ; between [NEi-1] and [NEi]
   if([NEm]==[NEn])
   {
      pre(NEm)=pre(NEm)  pre (NEn);
      follow(NEm)=follow(NEm)  follow(NEn);
      pre(NEn)=follow(NEn)=null;
   }
   Score(ph) = a* (pos+neg)/sum

        a={ 0                 pos = neg = 0
          { ln(1 / (2 * neg))  pos = 0, neg ≠ 0
          { ln(2 * pos)        pos ≠ 0, neg = 0
          { ln(pos / neg)      other

      ; "ph" is a term in pre(NE) or follow(NE)
      ; "sum" is the appearance count of "ph"
      ; in the training corpus C
      ; "pos" is the count of "ph" in P
      ; "neg" is the count of "ph" in N
   Mstatis = {MNE1UMNE2U……}
      MNEi = {{Score(hPi1), Score(hPi2)……},
   {Score(tPi1), Score(tPi2)……}}:
   ; where:"hPij" is a term in pre(NEi)
   ;       "Score(hPij)" is the score of hPij
   ;       "tPij" is a term in following(NEi)
   ;       "Score(tPij)" is the score of tPij
endProc
```

Algorithm 1. Tern Position Weight Statistical Model Mstatis Generation

Now, we can calculate the Score(S) for each sentence S by simply sums each word's score in S, and then to sort the sentences based on Score(S). We define here two parameters Min(pos) and Max(neg), then reclassify the corpus and form new positive examples P_{new} and new negative examples N_{new} based on this two paramenters:

```
Min(pos) = min Score(s) in the P
Max(neg) = max Score(s) in the N
Pnew={sentences whose score is higher than
      or equal to Min(pos)}
Nnew={sentences whose score is lower than
      or equal to Max(neg)}
```

If P_{new} is same as P and N_{new} is same as N, the process of classifying is over, otherwise create new statistical model M_{statis} based on P_{new} and N_{new}, and to reclassify the corpus again and again.

4.2 Pattern Generation

From those sentences in positive examples P, new patterns are created. Then they are added to current pattern set. Based on the Crystal algorithm [S. Soderland, etc. 95] new pattern set are generalized and evaluated as the description in algorithm2.

In the Crystal algorithm, the criterion to evaluate a pattern's validation is the error rate of patterns' extraction result. But we think that precision is much more preferred to recall, for users are much willing to see a few hundred of records with comparably high precision than to see hundreds of records with much redundancy. So in our system, we restrict the evaluation criteria like this: if a pattern can extract result from the negative examples N, then it is deleted from the pattern set.

```
P=an initial pattern removed from the
pattern set
   Loop
      If (P can extract result from N)
         exit loop
      P'=the most similar pattern to P
      If (P'==NULL), exit loop
      U=the unification of P and P'
      If(U can extract result from N)
         exit loop
      Delete all patterns covered by U
      Set P=U
   Add P to the pattern set
Return the pattern set
```

Algorithm 2. Pattern Generalizing and Evaluating

The new pattern set will extract more record from the training corpus, and these new record can be used to search for new instances from Internet.

Such as from the Sen 2 of chapter 4.1, can generate new pattern as:

New Pattern: [FIRM-NP] 以[MONEY] 拿得[FIRM-NP] [PERCENTAGE]的股权
Translation: [investor FIRM-NP] has spent [MONEY] in buying [FIRM-NP]'s [PERCENTAGE] stocks

After generated, it may match the instances as:
Sen 3. 如[FIRM-NP 新桥投资]以[MONEY 5 亿美

金]拿得[FIRM-NP 韩国第一银行][PERCENTAGE 51%]的股权 (Translation: such as [FIRM-NP Newbridge Capital] has spent [MONEY 500 millions dollars] in buying [FIRM-NP Korea First Bank]'s 51% stock)

With some other pattern's extraction result, the new record {"investor: Newbridge Capital", "invested-party: Korea First Bank", "invest-sum: 500 billions dollars"} will be used for the next iteration.

4.3 Information Management

Let's look at two sentences first:

Sen 4. 这将使得英特尔在上海封装/测试厂的投资总额由原先的 1.98 亿美元增至 5 亿美元。
Translation: It will increase Intel's investment at Shanghai Capsulation and Test factory from 198 millions dollars to 500 millions dollars.

Sen 5. 英特尔(中国)有限公司9月20日在北京宣布：向位于上海的英特尔生产制造工厂新增投资 3.02 亿美元，这使得英特尔在上海封装/测试厂的投资总额达到 5 亿美元。
Translation: Sep. 20th, Intel Co., Ltd.(China) announced in Beijing that it will invest another 302 millions dollars to its factory at shanghai, this increases Intel's investment at Shanghai Capsulation and Test factory to 500 millions dollars.

Obviously, these two sentences present the same event, we can extract "investor"(Intel), "invested party"(Shanghai Capsulation and Test factory) and "total investment"(500 millions dollars) from both of them, while from the first sentence we can additionally extract "completed investment"(198 millions dollars) and from the second sentence we can additionally extract "investment date"(Sep. 20th) and "additional investment" (302 millions dollars). In order to manage information conveniently we should merge them into one record.

How can recognize two sentences address the same event or two records are overlap partly? Here we adopt an idea of "key slots". In another word, we think that in a specific domain there must be some slots that are much more important than other slots. If both "key slots" are same, their extracted records are about the same event and should be merged into one record. In different domain, there are different "key slots" (such as "investor" and "invested party" as the key slots in our investment domain). According to experimental result, the merge precision reaches 98.9%.

5 EXPERIMENT AND EVALUATIONS

In the experimental system, we select the "investment" as the test domain. We build up the seed corpus consisting of 7 investment events with 55 single-slot surface patterns. ICTCLAS system (provided by Institute of Computing Technology, China Science Academy) is used in our pattern acquisition to tokenize and POS tag.

5.1 Evaluation of the Positive and Negative Examples

At last, the training corpus consists of 8706 sentences extracted from 3443 web pages. After classifying the training corpus, we get the positive examples containing 1035 sentences with the precision of 87.74%, and the negative examples including 2682 sentences (thresholdτ=2) with the precision of 89.68%.

In positive examples, errors are mainly occurred with those sentences, which say a future or cancelled investment event. In negative examples, errors are caused by the POS tagging and NE recognition in the sentence. Especially many abbreviations of company name cannot be recognized properly, thus some sentences are misjudged as negative examples.

5.2 Evaluation of the Bootstrapping Pattern Acquisition

The training corpus comes from the Internet, where an event may be appeared several times in different web sites with different sentences, as long as we can extract one time from any one of these heterochromatic sentences we think we succeed in extracting this event.

After about 20 iterations the size of corpus and the result of information are stable, as illustrated in the figure 3. As we can see, new pattern set has high precision.

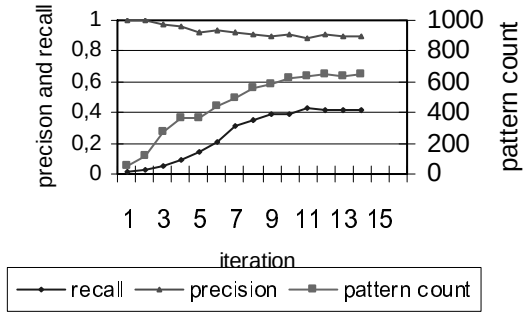

Figure 3: Evaluating the Learned Patterns

The recall is relatively low which is mainly caused by the following reasons:

1. In the pattern acquisition, we use only the positive examples, which is of only 11.89% in the total training corpus. Although recall on the training corpus can only reach 42.4%, the recall on the positive examples will be much better.
2. Many events have few instances even one instance; they are often of an individual or infrequent structure. Obviously it is much difficult to extract an event from few or even one instance.

Following shows two patterns in investment domain with its precision in pattern acquisition.

Pattern1 for invested-party with the precision of 88.9%:

Pattern1. 购买+[stuff FIRM-NP]+股份

Translation: buy stocks of [invested-party FIRM-NP]

Pattern2 for investment amount with precision of 94.1%:

Pattern2. 投资总额达到[invest-sum MONEY]

Translation: total investment reaches [invest-sum MONEY]

6 CONCLUSION

In this paper, we describe a bootstrapping method to acquire patterns in Information Extraction. Starting with a tiny seed corpus and patterns, the bootstrapping process collects new documents from the Internet and extracts new domain patterns. This approach overcomes the shortcoming in the scale of training corpus of traditional method. In order to improve the precision of acquisition, a classifier is used to identify new positive and negative examples for pattern acquisition and evaluation. A statistical model is used in the classification in our prototype. At last, we present a "key slots" idea in the information management module in order to merge multiple extracted records. Experiments show that the precision of pattern acquisition of our method is high.

However there are some points to be improved in our future work:

1. In classifying the new corpus, the recall is less than 50%. With the improvement on the recall, many new patterns can be acquired from the positive examples; also more negative examples can be used to evaluate these new patterns.
2. Different forms of a NE can not been identified, especially between the abbreviation of a company's name and its full name. For example, "上海汽车集团", "上汽集团", these tow phrases are the same company name but in different forms.

ACKNOWLEDGMENTS

This research work was supported by the grant No. 60083003 from National Natural Science Foundation of China.

REFERENCES

S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. Wiener, 1997. The Lorel query language for semi-structured data. *International Journal on Digital Libraries, 1(1):68-88, April*

S.Cluet, C.Delobel, J.Siemon and K. Smaga, 1998. Your Mediators Need Data Conversion! *in Processing of ACM-SIGMOD International Conference on Management of Data, 177-188.*

S.Cluet, S. Jacqmin, and J.Siemon, 1999. The New YTAL: Design and Specifications. *Technical Report, INRIA.*

A. Deutsch, M. Fernandex, D. Florescu, A.Levy and D. Suciu. A query language for XML. in International World Wide Web Conference, 1999

M. Fernandez, J. Siemon, P. Wadler, 1999. XML Query Languages: Experiences and Examples. http://wwwdb.research.bell-labs.com/user/simeon/xquery.html

E. Agichtein & L. Granvno, 2000. Snowball: Extracting Relations from Large Plain-Text Collections. *in Proceedings of the 5th ACM International Conference on Digital Libraries.*

R. Grishman, S. Huttunen & R. Yangarber, 2002, Real-Time Event Extraction for Infectious Disease Outbreaks. *in Proceedings of Human Language Technology Conference (HLT)*

S. Soderland, D. Fisher, J. Aseltine & Wendy Lhenert, 1995. CRYSTAL: Inducing a Conceptual Dictionary. *in proceedings of the 14th IJCAI' 95.*

Ellen Riloff & Janyee Wiebe, 2003. Learning Extraction Patterns for Subjective Extractions. *University of Utah, The Association for Computational Linguistics (ACL)*

Roman Yangarber, 2003. Counter_Training in Discovery of Semantic Patterns. *New York University, the Association for Computational Linguistics (ACL)*

Ellen Riloff, 1995. Little Words Can Make a Big Difference for Text Classification. *University of Utah, in proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval (SIGIR'95)*