ISSN:0975-9646

Web Intelligence: Analysis of Unstructured Database of Documents Using KnowItAll

Shweta Gupta, Kovid Agarwal, Shamla Mantri

Department of Computer Science, Pune University,MITCOE, Pune, Maharashtra, India

Abstract— **The web is a huge source of information. The ability to mine the web for the exact required information is a huge area for companies today. Many big companies are already working in this area and are searching for automation in this field because in the current situation this work has to be done manually. This paper presents a way to extract the exact required information from any database of documents like resume and present the relevant information to user in desired format with using the KnowItAll algorithm to increase the recall value. It is using the unsupervised learning concept and so it is faster than any manual method.**

Keywords— **KNOWITALL, Information Extraction, Text mining**

I. Introduction

The current scenario in today's times is that there is a rapid proliferation of information in digital format. People have less time to absorb more information. Currently there is a lack of tools to handle unstructured data. Text mining refers to taking a set of documents as input and extracting patterns[6], connections, profiles and trends from them.

Information Extraction is the task of automatically extracting knowledge from text. Unsupervised information extraction dispenses with hand-tagged training data. Because unsupervised extraction systems do not require human intervention, they can recursively discover new relations, attributes, and instances in a fully automated, scalable manner.

Web Intelligence[9] means analysing information from documents without having any prior knowledge on that topic.

II. Analysis of Unstructured Database of Documents

A. Text Mining

Text Mining[5] is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. A key element is the linking together of the extracted information together to form new facts or new hypothesis to be explored further by more conventional means of experimentation.

Text mining is different from what we're familiar with in web search. In search, the user is typically looking for something that is already known and has been written by someone else. The problem is pushing aside all the material that currently isn't relevant to your needs in order to find the relevant information. In text mining, the goal is to discover heretofore unknown information, something that no one yet knows and so could not have yet written down.

The working of text mining is shown below in Fig. 1. Initially unstructured text from various documents and databases is collected and tagging is done on that to yield output in the form of XML data which is then stored in databases. Then using all this data role-based interfaces are developed.

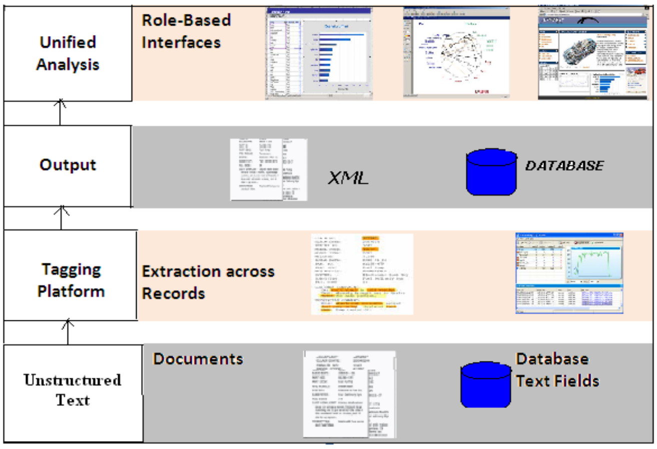


Fig. 1 Working in Text Mining

Text mining is unique because:

i. Feature extraction is possible.

ii. Very large number of features that represent each of the documents can be derived.

iii. The need for prior knowledge.

iv. Even patterns supported by small number of documents may be significant.

v. Huge number of patterns, hence need for visualization with Interactive exploration.

Given any unstructured text document tagging is done on it and data is classified or tagged further as shown in Fig. 2. Data is classified and analysed. Its type is determined before storing it in the Knowledge Database.

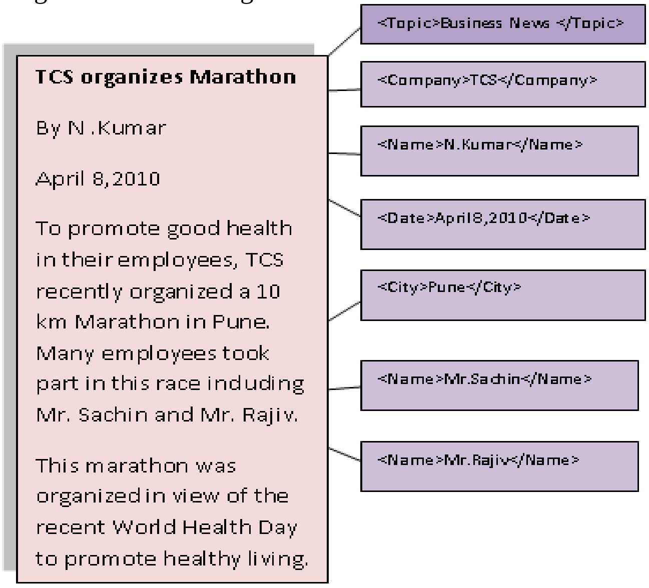

TCS organizes Marathon
By N .Kumar
April 8, 2010
To promote good health in their employees, TCS recently organized a 10 km Marathon in Pune. Many employees took part in this race including Mr. Sachin and Mr. Rajiv.
This marathon was organized in view of the recent World Health Day to promote healthy living.
<Topic>Business News </Topic>
<Company>TCS</Company>
<Name>N.Kumar</Name>
<Date>April8,2010</Date>
<City>Pune</City>
<Name>Mr.Sachin</Name>
<Name>Mr.Rajiv</Name>

Fig. 2 Data Tagging Example

B. Pattern Learning

The patterns are sequences of tokens, skips, and slots which have been found out from the documents. The tokens can match only themselves, the skips match zero or more arbitrary tokens, and slots match instance attributes. Regular expressions can be devised using these which further helps in extraction of new occurrences.

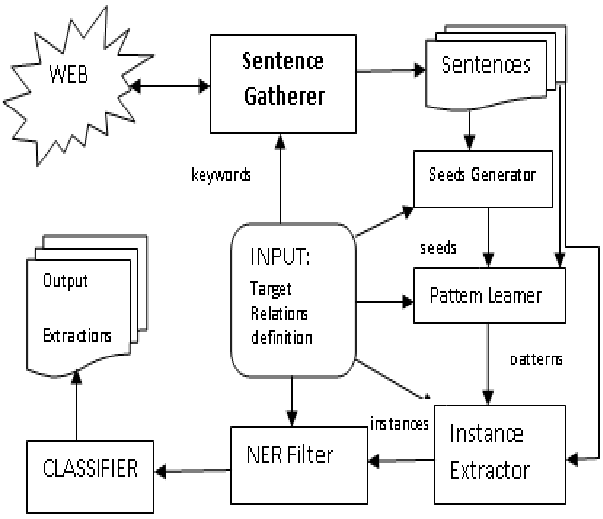


Fig. 3 Flowchart of Web Intelligence using KnowItAll

Major Steps in Pattern Learning:

i. The sentences containing the arguments of the seed instances are extracted from the large set of sentences returned by the Sentence Gatherer.

ii. Then, the patterns are learned from the seed sentences.

iii. We need to generate automatically
   a. Positive Instances
   b. Negative Instances

iv. Finally, the patterns are post-processed and filtered.

The final flow chart of Web Intelligence using KnowItAll is shown in Fig. 3.

C. Pattern Generation

The patterns for a predicate P are generalizations of pairs of sentences from the positive set of P. The function Generalize(S1, S2) is applied to each pair of sentences S1 and S2 from the positive set of the predicate. The function generates a pattern that is the best generalization of its two arguments.

The following pseudo code shows the process of generating the patterns:

```
For each predicate P
For each pair S1, S2 from PositiveSet(P)
Let Pattern = Generalize(S1, S2).
Add Pattern to PatternsSet(P).
```

III. **KNOWITALL**

KnowItAll[1][2] is a system developed at University of Washington by Oren Etzioni and colleagues (Etzioni, Cafarella et al. 2005).

KnowItAll is an autonomous, domain independent system that extracts facts from the Web. The primary focus of the system is on extracting entities (unary predicates), although KnowItAll is able to extract relations (N-ary predicates) too.

KnowItAll introduces a novel, generate-and-test architecture that extracts information in two stages. Inspired by Hearst [3], KnowItAll utilizes a set of eight domain independent extraction patterns to generate candidate facts.

Next, KnowItAll automatically tests the plausibility of the candidate facts it extracts using pointwise mutual information (PMI) statistics computed by treating the Web as a massive corpus of text. Extending Turney's PMI-IR algorithm [4], KnowItAll leverages existing Web search engines to compute these statistics efficiently. Based on these PMI statistics, KnowItAll associates a probability with every fact it extracts, enabling it to automatically manage the tradeoff between precision and recall. Since we cannot compute "true recall" on the Web, the paper uses the term "recall" to refer to the size of the set of facts extracted.

We describe and compare three distinct methods added to KnowItAll in order to improve its recall:

- *Pattern Learning* (*PL*): learns domain-specific patterns that serve both as extraction rules and as validation patterns to assess the accuracy of instances extracted by the rules.
- *Subclass Extraction* (*SE*): automatically identifies subclasses in order to facilitate extraction.

For example, in order to identify scientists, it is helpful to determine subclasses of scientists (e.g., physicists, geologists, etc.) and look for instances of these subclasses.

- *List Extraction* (*LE*): locates lists of class instances, learns a "wrapper" for each list, and uses the wrapper to extract list elements.

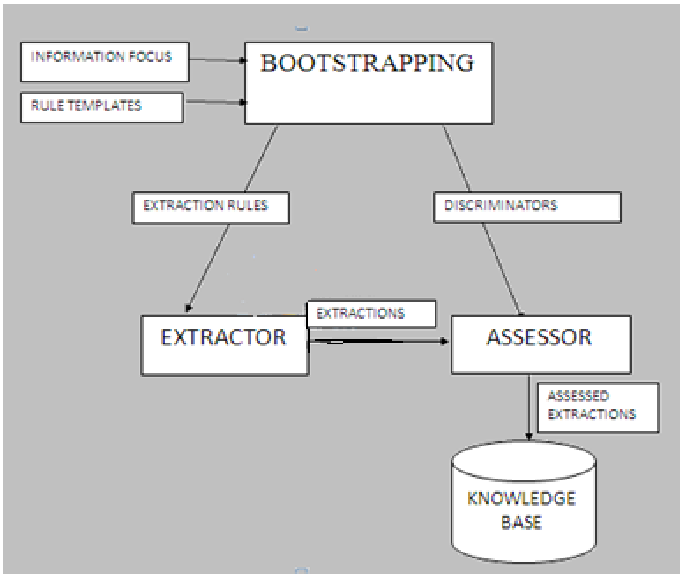
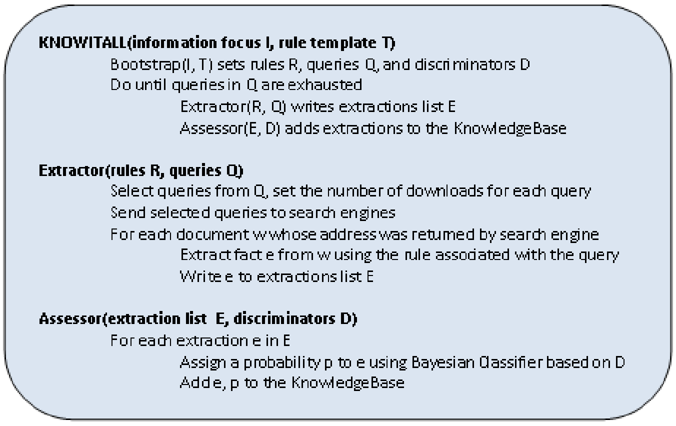


Fig. 4 Flowchart of the main components in KnowItAll

Each of the methods dispenses with hand-labelled training examples by bootstrapping from the information extracted by KnowItAll's domain-independent patterns. We evaluate each method experimentally, demonstrate their synergy, and compare with the baseline KnowItAll system described in [2]. A system flowchart is shown in Fig. 4 and pseudocode in Fig. 5 for the baseline KnowItAll system.



```
KNOWITALL(information focus I, rule template T)
        Bootstrap(I, T) sets rules R, queries Q, and discriminators D
        Do until queries in Q are exhausted
                Extractor(R, Q) writes extractions list E
                Assessor(E, D) adds extractions to the KnowledgeBase

Extractor(rules R, queries Q)
        Select queries from Q, set the number of downloads for each query
        Send selected queries to search engines
        For each document w whose address was returned by search engine
                Extract fact e from w using the rule associated with the query
                Write e to extractions list E

Assessor(extraction list E, discriminators D)
        For each extraction e in E
                Assign a probability p to e using Bayesian Classifier based on D
                Add e, p to the KnowledgeBase
```

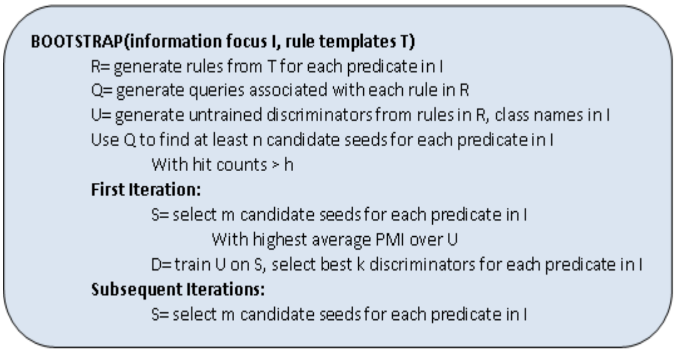
Fig. 5 High level pseudocode for KnowItAll

KnowItAll's Bootstrapping[7][8] step uses a set of domain-independent extraction patterns to create its set of extraction rules and discriminators for each predicate in its focus. The Bootstrapping is fully automatic, in contrast to other bootstrapping methods that require a set of manually created training seeds.

The two main KnowItAll modules are the Extractor and the Assessor. The Extractor creates a query from keywords in each rule, sends the query to a search engine, and applies the rule to extract information from the resulting Web pages. The Assessor computes a probability that each extraction is correct before adding the extraction to KnowItAll's knowledge base. The Assessor bases its probability computation on search engine hit counts used to compute the mutual information between the extracted instance of a class and a set of automatically generated discriminator phrases associated with that class. This assessment process is an extension of Turney's PMI-IR algorithm [4].

Bootstrapping is able to find its own set of seeds to train the discriminators, without requiring any hand-chosen examples. It does this by using the queries and extraction rules to find a set of candidate seeds for each predicate. Each of these candidate seeds must have a minimum number of hit counts for the instance itself; otherwise the PMI scores from this seed will be unreliable.

After assembling the set of candidate seeds, Bootstrapping computes PMI*(c, u)* for each candidate seed *c*, and each untrained discriminator phrase *u*. The candidate seeds are ranked by average PMI score and the best *m* become the first set of bootstrapped seeds. The pseudocode for Bootstrapping is shown in Fig. 6 below.



```
BOOTSTRAP(information focus I, rule templates T)
        R= generate rules from T for each predicate in I
        Q= generate queries associated with each rule in R
        U= generate untrained discriminators from rules in R, class names in I
        Use Q to find at least n candidate seeds for each predicate in I
                With hit counts > h
        First Iteration:
                S= select m candidate seeds for each predicate in I
                        With highest average PMI over U
                D= train U on S, select best k discriminators for each predicate in I
        Subsequent Iterations:
                S= select m candidate seeds for each predicate in I
```

Fig. 6 Pseudocode for Bootstrapping

Thus we can use untrained discriminator phrases to generate our first set of seeds, which we use to train the discriminators.

IV. Conclusions

Web Intelligence has been recognized as a new direction for scientific research and development to explore the fundamental roles as well as practical impacts of Artificial Intelligence (e.g., knowledge representation, planning, knowledge discovery and data mining, intelligent agents, and social network intelligence)

The bulk of previous work on Information Extraction has been carried out on small corpora using hand-labelled training examples. The use of hand-labelled training examples has enabled mechanisms such Hidden Markov Models or Conditional Random Fields to extract information from complex sentences. In contrast, KnowItAll's focus is on unsupervised information extraction from the Web. KnowItAll takes as input a set of predicate names, but no hand-labelled training examples of any kind, and bootstraps its extraction process from a small set of generic extraction patterns. To achieve high precision, KnowItAll utilizes a novel generate-and-test architecture, which relies on mutual-information statistics computed over the Web corpus.

Although KnowItAll is still "young", it suggests futuristic possibilities for systems that scale up information extraction, new kinds of search engines based on massive Web based information extraction, and the automatic accumulation of large collections of facts to support knowledge-based AI systems.

Acknowledgment

This research was supported and encouraged by TRDDC, Hadapsar, Pune. We are thankful to Mr. Rajiv Srivastava, Consultant at TCS and Mr. Girish K.Palshikar, Scientist at TCS and also many other people from TRDDC who helped us directly or indirectly to gain immense knowledge on this topic. We are also thankful to Prof. Shamla Mantri and Prof.R.K. Bedi and many other teachers from our college for their guidance, support and encouragement.

References

[1] Oren Etzioni , Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, Alexander Yates, Unsupervised named-entity extraction from the Web: An experimental study, 2005.

[2] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, A. Yates, Web-scale information extraction in KnowItAll, in: Proceedings of the 13th International World Wide Web

[3] M. Hearst, Automatic acquisition of hyponyms from large text corpora, in: Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992, pp. 539–545.

[4] P.D. Turney, Mining the Web for synonyms: PMI-IR versus LSA on TOEFL, in: Proceedings of the Twelfth European Conference on Machine Learning, Freiburg, Germany, 2001, pp. 491–502.

[5] Ronen Feldman, Text Mining and Link Analysis

[6] S. Soderland, Learning information extraction rules for semi-structured and free text, Machine Learning 34 (1–3) (1999) 233–272.

[7] E. Riloff, R. Jones, Learning dictionaries for information extraction by multi-level bootstrapping, in: Proceedings of the Sixteenth National Conference on Artificial Intelligence, Orlando, FL, 1999, pp. 474–479.

[8] W. Lin, R. Yangarber, R. Grishman, Bootstrapped learning of semantic classes from positive and negative examples, in: Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data, Washington, DC, 2003, pp. 103–111.

[9] http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1300364