Lexical Pattern Generalization for Ontology Learning and Population: A Survey

Y. Rastegari, M. Sayadiharikandeh, and B. Zibanezhad

***Abstract*—Information extraction and refinement systems rely on a set of extraction patterns in pattern based approaches in order to construct taxonomies and finding instances of concepts in a corpus. Pattern based methods have high precision and suffer from low recall. Pattern generalization and modeling techniques can increase matching power and decrease GAP between training and test data.**

***Index Terms*—pattern generalization, lexical pattern modeling, semantic web, ontology learning and population**

I. Introduction

Since semantic web has been introduced as a part of the next generation of web in the context of information retrieval and refinement by Tim Berners-Lee in 1999, there have been many efforts in order to define and develop its related concepts. In semantic web different concepts of information and services are defined which makes it comprehensive for its consumers like machines. Over time, as descriptive languages have been designed like XML, XML Scheme, RDF and OWL, web sites became more structured and more understandable for machines. Considering the trend of webs toward to semantic web issues, it is expected that template of the web pages should change but if we take a look at the number of static web pages (known as surface webs) which is more than 4*10^9 [12], we see that it is kind of impossible. According to web developer's point of view, it is not reasonable to change the structure of web toward to semantic web but some techniques should be used to make web conceptual to be understood by machines and search engines. Ongoing techniques and technologies should follow a structured template. Thus, there has been so much research on this issue that led to semantic web levels depicted in fig.1.

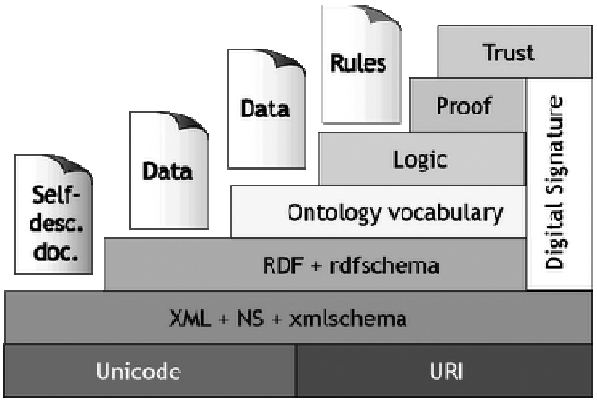

Figure 1. Semantic Web Stack

Yousef Rastegari is with the Islamic Azad University-South Tehran Branch, Tehran, Iran (email: rastegari.yousef@gmail.com)

Mohsen Sayadiharikandeh is with the Sharif University of Technology, Tehran, Iran (email: m_sayyadi@ce.sharif.edu)

Bahareh Zibanezhad is with the Islamic Azad University, Najafabad Branch, Esfahan, Iran (email: b.zibanezhad@gmail.com)

One of the most important layers of this stack is ontology layer which includes Ontology learning and population. Learning refers to (semi) automatic construction of taxonomy of concepts (ontology tree) and population refers to instantiating ontology concepts. In [2] ontology is defined lexically as *science of existence* and proposed a mathematical definition as shown in (1).

$$O := (C, \leq_C, R, \sigma_R, \leq_R, A, \sigma_A, T) \quad (1)$$

Taxonomy extraction techniques are generally classified in the following groups [9]:

- Systems based on pattern extraction and matching
- Systems based on distributional properties of words
- Systems based on dictionary definitions analysis
- Web-based combinational methods [1]

In this paper we will focus on pattern based methods and especially on different methods of modeling and generalization of patterns as the major component of these methods.

The rest of this paper is organized as follows: In section 2 we address the different definition of pattern. Next in section 3, we describe the evaluation parameters, precision and recall related to pattern based approaches. Section 4 described general process for pattern generalization and current methods. Finally, in section 5 conclude the paper.

II. What's a Pattern

There is no unified definition for pattern in pattern based methods. Each method proposed an arbitrary definition for pattern based on context and its idea. As an example, in [8] some lexico-syntactic patterns are used for acquisition of hyponyms. Hearst used six predefined patterns that were collected from different documents. In [3][4][5] dependency paths are used as patterns. In [1][8][9] consider a sequence of tokens as a pattern which could be retrieved after applying some preprocessing as described in detail in section 4.B. Some others like [6][7] use Hidden Markov Model (HMM) to explain patterns. However [10] evaluates all patterns retrieved by machine learning methods and shows how to convert a pattern to an entity which is understandable and annotatable by machines. We will discuss about pattern modeling and generalization in more details in section 4.

III. Precision in Contrast with Recall

Precision is defined as number of correct instances divided by the total number of instances retrieved. On the other hand, Recall is number of correct retrieved instances divided by all correct instances which should be retrieved. Pattern based methods use pattern matching on an input sentence and check if it matches or not. Patterns are usually built based on positive instances and have high precision but suffer from

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

low rate of recall. There could be multitude of factors which contribute in lowness of recall but among them two following factors are the most important: 1) Corpus sparseness and 2) Low flexibility of patterns in matching new instances. Corpus sparseness means low level of occurrence of instances in predefined patterns. For example consider Iran as an instance of concept Country which occurs before or after of its concept with more than patterns maximum length. But as patterns are commonly short, pairs of (instance, concept) can't be matched. On the other hand, long patterns are unusable due to low rate of occurrence in corpus and recall. To solve this problem, using (semi) automatic techniques are advised for generating patterns. Six predefined patterns of Hearst could not retrieve many pairs of (instance, concept) but in [11][1][9] some techniques have been proposed that respectively use whole WWW pages, search engine results (snippets), Wikipedia documents as a massive corpus and find more patterns than Hearst predefined patterns for different binary relationship. Low flexibility of pattern means, low power of matching with different new input sentences. As flexibility increases, GAP between training and test sets decreases.

IV. Pattern Generalization

In this section, firstly we address the different patterns representations and then different methods of pattern generalization are explained.

A. Patterns Generalization: Modeling

There have been used different methods of patterns modeling. Followings are some advantages of patterns modeling: 1) Increasing the machine readability property of patterns 2) Applying constraints on input strings 3) Defining physical concepts for different parts of pattern 4) Increasing the power of acceptance or in other word patterns generalization. Generally we could classify patterns modeling into the following categories.

1) Patterns Representation in Dependency Paths: In this case patterns are modeled in dependency paths which are extracted from dependency tree of syntactic parser. In [4] dependency path is used between two positive instances, and then X and Y are replaced by them as tow placeholders. After that, the final dependency path between these placeholders is called 'Bridge' and the containing sentence is named as a 'Pattern'. In [5] Chain and sub-tree models are addressed based on dependency paths between tokens. In [3] dependency paths are used as representing model for patterns. Dependency paths could be used in order to decrease corpus sparseness problem and also used due to its non-dependency to input tokens. In [4] two bridges are equal when the sequence of nodes and labels (type of dependency between a pair of tokens) are the same and there is no constraint on tokens to be the same. Most of pattern generating methods use pairs of positive instances and find them in the corpus and cut its containing sentence. Then Windowing operation is applied on extracted sentence with a desired length of l. If some positive pairs have been appeared in window it is used for the target relation. Thus, many sentences with length more than l which explain our target relation will be ignored due to length limitation but this problem is decreased in patterns based on dependency path.

2) Patterns Representation in Hidden Markov Model: Hidden Markov Model has a powerful statistical and mathematical base and is used in different areas such as Gene Prediction, Speech Recognition, Cryptanalysis, POS Tagging and etc. Recently HMM has been used in structure extraction [7] and patterns modeling [6] areas. In [7] nested HMM is used to structure extraction and segmenting input sentences into meaning parts. For instance, the proposed model is used for address or bibliography segmentation into meaning parts. The proposed model can be generalized and used in the field of pattern modeling. The proposed model includes N hidden states and M observable states. Hidden states are predefined segments and emits to observable states that are dictionary of symbols [7]. Matrixes M and A are indicators for transmission and emission matrixes respectively. Model is designed according to training data. HMM is used in [6] to decrease the gap between training and test instances. Other advantages with the HMM approach is that it can handle new data robustly, is computationally efficient and is easy for humans to interpret and tweak [7].

3) Representing Patterns by Sequence of Tokens: Patterns could be presented by sequence of tokens in a string format without any specified model. Regular expressions exist in this category. [1][8][9] Use this type of patterns. Most of operations (to be mentioned in 4.B) are applied on input strings in order to convert them to patterns.

4) Generated Patterns from Machine Learning Methods: This section includes different patterns resulted from machine learning algorithms. Each pattern has its own benefits and drawbacks. Each pattern of this category represents an entity with some attributes and constraints. In [5] different patterns have been evaluated.

B. Patterns Generalization: General Processes

After collecting some sentences which contain pair of (instance, concept) of our target relation, it is advised to do some operations in order to convert a special-purpose string to a general-purpose pattern. Among this operations are the followings:

- Part-Of-Speech (POS) Tagging: in this step the words of a sentence are marked due to its corresponding particular part of speech (e.g.: verb, noun, adjective and etc.)
- Stemming: Process of converting a word to its root. As an example the conversion of words *fished*, *fish*, *fishing*, *fisher* to the root *fish*.
- Chunking: in this process noun phrases are retrieved and replaced by NP. This process should be done when simple nouns of a noun phrase are not needed separately.
- Selective Substitution: Process of replacing one word with another one which has more general meaning.
- Windowing or Cropping: Cutting a part of a string from the first word to second one is called Windowing and Cropping refers to cutting part of a string around target word. In both, the length of cutting *l* is important. As an example Windowing with *l*=4 results a string with length 4 (considering both first and second words) and Cropping results a string with length 2*l*+1 (considering

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

target word). If positive instances haven't been captured by Windowing, related string should be removed because it can't represent a pattern with the desired length.

We should mention that it is not needed to do all the above processes. Words replacement could be done in small scale as [4] in which two words are replaced by their placeholders or in [9] two entries of Wikipedia are replaced by Target and Entry tokens. Otherwise word replacement could be done on a large enough scale level by mapping tables such as [6].

C. Pattern Generalization: Current Methods

1) Customized Edit Distance Algorithm: Reference [9] used a customized version of Edit-Distance (ED) algorithm to generalize the patterns in a positive way. ED algorithm takes two strings A and B as inputs and evaluates the number of operations (Insertion, Deletion, and Replacement) required transforming A into B. Another matrix called D is defined and initialized in parallel with matrix M (the main matrix of the ED algorithm). This matrix shows in each step which operation is used to be done in order to transform A to B (it could be referred as Log Matrix). The proposed algorithm in fig. 2 benefits from this matrix D to make a generalization of patterns. Eventually to enhance the precision of patterns and to prevent from inordinate generalization that causes irrelevant relations, the condition of equivalence of two tokens changed to equivalence of their POS tags and replacement operation or in another word using OR condition (e.g. nice | a) is allowed only if two tokens have the same POS tags. As a final step, threshold value is defined and used. When the algorithm meets this threshold value, it stops going forward which means stop generalization.

(1) Initialise the generalised pattern G as the empty string.
(2) Start at the last cell of the matrix $\mathcal{D}(i,j)$. In the example, it would be $\mathcal{D}(5,4)$.
(3) While we have not arrived to $\mathcal{D}(0,0)$,
  (a) If $(\mathcal{D}(i,j) = \texttt{E})$, then the two patterns contained the same token A[i]=B[j].
    - Set $G = A[i]\ G$
    - Decrement both i and j.
  (b) If $(\mathcal{D}(i,j) = \texttt{U})$, then the two patterns contained a different token.
    - $G = A[i]|B[j]\ G$, where | represents a disjunction of both terms.
    - Decrement both i and j.
  (c) If $(\mathcal{D}(i,j) = \texttt{R})$, then the first pattern contained tokens not present in the other.
    - Set $G = *\ G$, where $*$ represents any sequence of terms.
    - Decrement i.
  (d) If $(\mathcal{D}(i,j) = \texttt{I})$, then the second pattern contained tokens not present in the other.
    - Set $G = *\ G$
    - Decrement j

I: Insert
R: Remove
E: Equal
U: Replace

```
It is a kind   of
It is nice     of
-----------------
It is a|nice * of
```

Figure 2. Patterns generalization algorithm based on matrix D elements

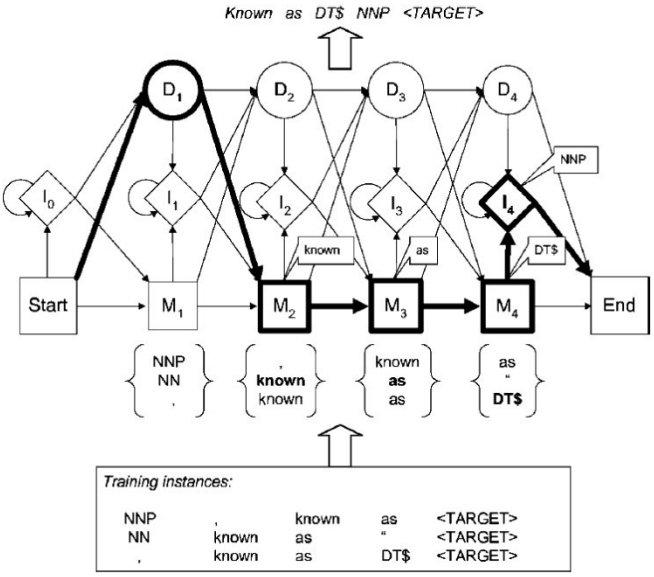

Figure 3. Using HMM in Patterns representation and generalization

2) Profile Hidden Markov Model: HMM is used for increasing the acceptance rate of patterns and decreasing the gap the between training and test instances. As patterns are collected based on training data, they have weakness to handle test data and find instances of target relation. In [6] this mathematical model is used to generalize patterns in QA systems. Let's use the example of mentioned paper to explain the usefulness of this model by increasing acceptance rate. As shown in fig. 3 HMM model is constructed based on training data. In fig. 3, M_i (i=1 ... l) refers to matching states and D_i and I_i in parallel with M_i refer to Deletion and Insertion states respectively. Matching states are related to tokens of patterns instances. l refers to length of the model which is constructed based on length of patterns instances that relates to Cropping or Windowing operations. From each matching state M_i, token t will be emitted with probability of $P(t|M_i)$. D_i state is a deletion state which is only assumed because of passing M_i without any emission. I_i is an insertion state capable of emitting to each token with probability of $P(t|I_i)$. As shown in fig. 3 I_i states have self loop and are capable of producing any number of tokens.

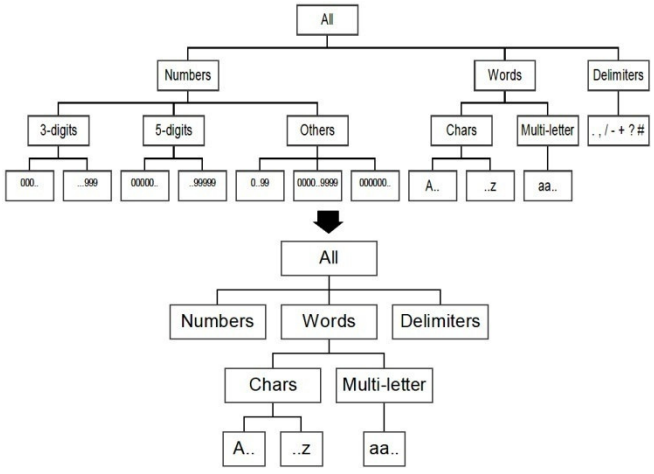

Figure 4. Symbols dictionary (observed states of HMM) in hierarchical view

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

3) Hierarchical Feature Selection: Reference [7] used nested HMM for pattern modeling and dividing sentences into predefined segments. Here one issue is how to convert the dictionary of symbols (Observing states) to a level of generality which leads in highest level of accuracy. Dictionary includes letters, numbers and delimiters. Dictionary is designed hierarchically as depicted in fig. 4. The Higher level has less clarity and lower level has less generality. Low clarity decreases the power of model in recognizing of the best transition path between matching states and accessing the desired output. In contrast with clarity the level of generality is also important. The lower level needs more training data to consider emission probability of all lower level symbols and prevents from decrease of matching power in handling new instances. Reference [7] uses pruning method to put a trade-off between two mentioned factors (Clarity and Generality). Set of segmented data is divided into two parts: Validation and Training (training should be twice big as validation). Starting from the lowest level of tree, assuming each symbol of training data as a token, in each level evaluate the model on validation part. This type of pruning is done to reach the highest precision. The highest precision leads to best hierarchy level of dictionary. Thus, patterns modeled based on HMM reach their highest precision of generalization.

V. Conclusion

Patterns based methods have high precision but suffer from low recall. Different approaches have their own point of view in patterns and used different techniques to reach higher recall like pattern generalization or pattern modeling based on dependency paths, Hidden Markov Model or some machine learning generated pattern models. This paper is important in a way that in generalization and modeling point of view, there hasn't been any survey to evaluate patterns as the most important component of these pattern based methods for information extraction and refinement.

References

[1] M. Neshati, "Ontology Learning and Taxonomy Construction from Text Corpus," M.S. thesis, Sharif University of Technology, Tehran, Iran, Nov. 2007.

[2] P. Cimiano, Ontology learning and Population from Text: Algorithms, Evaluation and Applications, 1rd ed., Springer, 2006.

[3] R. Snow, S. Jurafsky, A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery," In: Advances in Neural Information Processing Systems (NIPS 2004), Vancouver, British Columbia, December 13-18, 2004.

[4] F.M. Suchanek, G. Ifrim, G. Weikum, "LEILA: Learning to Extract Information by Linguistic Analysis," Proceedings of the 2nd Workshop on Ontology Learning and Population, pages 18–25, Sydney, July 2006.

[5] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition," Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, Sapporo, Japan , 2003, pp. 224-231..

[6] H. CUI, M. KAN, and T. CHUA, "Soft Pattern Matching Models for Definitional Question Answering ," ACM Transactions on Information Systems, Vol. 25, No. 2, Article 8, April 2007.

[7] V. Borkar, K. Deshmukh, S. Sarawagi, "Automatic segmentation of text into structured records," ACM SIGMOD Record, vol. 30, Issue 2, 2001, pp. 175-186, doi: 10.1145/376284.375682.

[8] M.A. Hearst, "Automatic Acquisition of Hyponyms from Large Text Corpora," Proceedings of the 14th conference on Computational linguistics-Volume 2, France, 1992, pp. 539-545.

[9] M. Ruiz-Casado, E. Alfonseca, P. Castells, "Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia," Elsevier Science Publishers B. V., Data & Knowledge Engineering, Vol. 61, No. 3., June 2007, pp. 484-499.

[10] I. Muslea, "Extraction Pattern for Information Extraction Tasks: A Survey," In AAAI-99 Workshop on Machine Learning for Information Extraction, 1999, pp. 1-6.

[11] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D.S. Weld, A. Yates, "Web-Scale Information Extraction in KnowItAll (Preliminary Results)," Proceedings of the Thirteenth International World Wide Web Conference, ACM Press, New York, 2004, pp. 100-110.

[12] R. Baeza-Yates, Excavando la web, El profesional de la informaci´on 13 (1) (2004) 4–10.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)