Volume 5, Issue 2, February 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Hypernym-Hyponym Acquisition-A Juxtapose Study

Sheetal Saini, Prof. JayaSurya Venugopalan
CSA Department, Christ University,
Bangalore, India

Abstract— Relationship acquisition has always contributed as an important enhancement to Natural Language Processing. Using "definition extraction" methods proved to be a very useful approach, not only for relationship extraction but also for enhancing the limited coverage of words in WordNet. It also played a major role in developing question answer systems and ontology learning for unrestricted data. Various epistemologies developed for the later but it was noticed that many among the developed approaches suffered from low precision and low recall, especially the ones which were based on lexico syntactic patterns. This paper presents a comparative study of two different approaches used for Hypernym-Hyponym discovery. We start with analyzing the Pattern matching approach based on the Hearst Pattern and then compare it with LEILA which uses link grammar parser and deep syntactic analysis for relationship acquisition.

Keywords— Hypernym, LEILA, Pattern Matching, Hearst Pattern, lexico-syntactic pattern

I. INTRODUCTION & PREVIOUS WORK

This paper revolves around the comparative study of the concept which is acquisition of hypernym-hyponym relations for the given unrestricted text. By unrestricted text we mean text which could include any type of information and which is not pre-processed. In both the techniques the data/text is pre-processed in order to bring it into a form suitable for relationship extraction. Before drawing the comparisons between the two approaches of hypernym-hyponym retrieval , let define what a hypenym-hyponym pair is.

A noun N^1 is a "Hyponym" of noun N^2 if N^1 is a subtype of N^2, for instance in the sentence "Jaguar is a car". "Jaguar (N^1)" is a hyponym of "car (N^2)" and conversely "car" is a hypernym of Jaguar. Hypernym-Hyponym relation is also referred to as "is-a" relationship . Marti.A.Hearst in 1992 proposed a very simple yet powerful approach for acquisition of is-a relationship, she proposed six handcrafted patterns which well defined the hypernym-hyponym relationship between two nouns. Following were the six patterns.

- X such as Y
- such X as {Y,}* (or|and) Z
- X {,Y}* {,} or other Z
- X {,Y}* {,} and other Z
- X including{,Y}*{,}(or|and) Z
- X especially{,Y}*{,}(or|and) Z

Figure 1: The handcrafted six Hearst patterns[1]

However the Pattern matching approach also has its share of shortcomings which will be discussed further in the paper along with the solutions that the LEILA approach provides . LEILA relies on deep syntactic analysis for text as well as other advanced NLP methods like anaphora resolution, and combines them with machine learning techniques for robust and high-yield information extraction.

II. PATTERN MATCHING APPROACH

We implement the pattern matching approach as per the algorithm given below

Phase 1

- Input: The six handcrafted patterns [1] .
- Output: Obtain (hyponym,hpernym) pairs connected by any of the patterns given as input.

Phase2

- Input: The obtained (hyponym,hypernym) pairs in phase 1.
- Output: Sentences containing (hypernym,hyponym) given as input in phase 2 step 3.
- Extract the patterns from the sentences after parsing them.
- Iterate the steps of Phase1&2 unless no new hypernym-hyponym pairs or patterns are obtained.

Before we implement the algorithm the data is pre processed and Part Of Speech-Tagged (POS-Tagged) using any available POS tagger in this paper we have made use of Stanford POS tagger. However Part Of Speech tagging (POS-tagging) is just one of the steps used for data pre-processing. Below is the step wise pre-processing of data

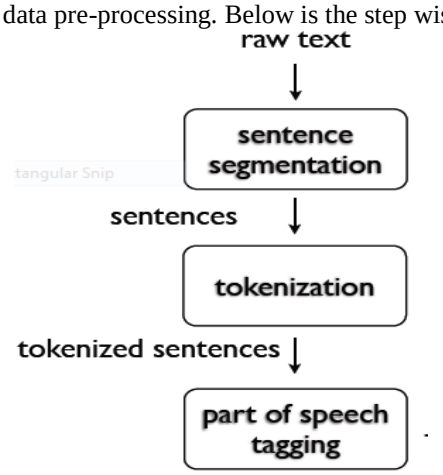

Figure2. The above figure shows the pre-processing steps of text

We iterate the pattern based algorithm on the Wikipedia data set and noticed that not only the algorithm suffers from limited recall but it also turns erroneous even if the data gets slightly complex .The algorithm shows errors owing to the local nature of the syntactic patterns . For instance in the sentences

(i) .....flowers such as roses..
(ii) .....local train in city such as Shramshakti...

The above given sentences match the pattern "X such as Y" (fig 1). For (i) where X is bound to flowers and Y is bound to roses the algorithm works fine and gives out the result (flowers, roses) as hypernym hyponym pairs but in case of (ii) where "X" is bound to "city" and "Y" is bound to "Shramshakti" we observe that the algorithm gives out (city, Shramshakti) as result which is incorrect.

III. LEILA

LEILA uses link grammar parser [3] for parsing/pre-processing the data and the linkages produced by the link grammar parser is used by the algorithm for deep syntactic analysis. LEILA was specially developed for relationship acquisition from web documents. Prior to stating the algorithm we define the terms linkage and pattern with respect to the algorithm.

- Linkage is the connected graph formed by the link parser as a result of parsing a sentence, a linkage can be complex or simple. In the example in Figure 4 [2] we establish a linkage where the relation of subject "Chopin" is well defined with the word "composer" followed by "n" which means the word is a noun.

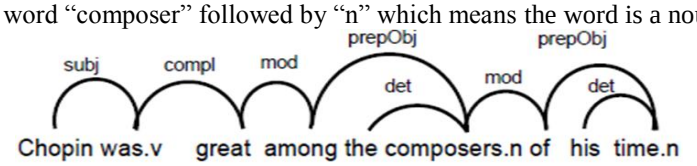

Figure4: Example of a linkage produced by link parser

- Pattern is defined as the linkage in which the two nouns are replaced using place holders, place holder could be any word i.e. X,Y,Z etc, in the example in Figure 5 [2]

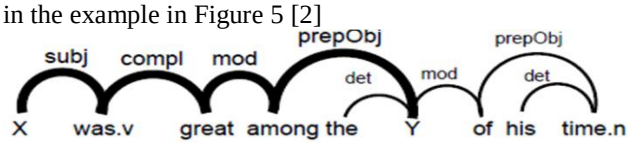

Figure5: Example of a pattern produced by LEILA algorithm

The core algorithm [2] process in three phases

Document Pre-processing

- Convert the document into linkages using link grammar
- Decide on a relationship to be discovered i.e. birthdates and persons

Discovery Phase

- Seeks linkages in which contains seed pairs
- The seed pairs in the linkage are replaced using placeholders, thus producing a pattern.
- The patterns are collected as positive patterns.
- The algorithm iterates again through the sentences and locates all linkages matching the positive pattern, but produces a counterexample.
- The corresponding patterns are collected as negative patterns

Testing Phase

- The algorithm considers again all sentences in the corpus.
- For each linkage, it generates all possible patterns by replacing two words by placeholders.
- If the two words form a candidate and the pattern is classified as positive, the produced pair is proposed as a new element of the target relation (an output pair).

IV. EXPERIMENTAL MEASURES, RESULTS AND COMPARISONS

We run both the algorithms on Wikipedia data set and Snow et al.(2004) and calculate the efficiency of the algorithms as per the following measures:

- Precision: defines the number of correct output pairs retrieved by the system over the number of pairs that were marked correct pairs.
- Recall: defines the number of output pairs correctly retrieved by the system over the number of correct output pairs in the dataset.
- F1-measure: is the harmonic mean of precision and recall values given by 2(PR)/P+R , where P denotes the precision and R denotes the Recall

Table I Performance of Algorithm on Wikipedia Dataset

Algorithm	**Measures**		
	Precision	**Recall**	**F1**
Pattern Matching	63.68%	10%	17.28%
LEILA	76.82%	33.59%	46.74%

Table III Performance Of Algorithm On Snow et al.(2004) Dataset

Algorithm	**Measures**		
	Precision	**Recall**	**F1**
Pattern Matching	86.68%	30%	44.57%
LEILA	90.15%	49.80%	32.07%

The above tables show that the LEILA which is based on link grammar parsing beats the pattern matching approach by huge margins in precision and recall measures. Also LEILA solves the shortcoming of Hearst Pattern caused due to the local nature of the patterns which we discussed in section II. Since LEILA uses deep syntactic analysis it is able to overcome the problem faced by Hearst Patterns.

V. CONCLUSION

This paper presents an overall comparison of Hearst Pattern approach and LEILA. The comparison was supported by the Wikipedia data set and Snow et al(2004) data set. Results suggest that LEILA is more efficient than the Hearst Pattern approach.

REFERENCES

[1] Automatic Acquisition of Hyponyms from Large Text Corpora, Fourteenth International Conference on Computational Linguistics, Nantes France, July 1992
[2] LEILA: Learning to Extract Information by Linguistic Analysis, Max-Planc University, Germany,2006
[3] Parsing English With A Link Grammar, Carnegie Mellon University, 2004
[4] Hudson R . Word Grammar,Basil Black-well ,1984
[5] Hearst .M.A. , Noun Homograph Disambiguation Using Local Context in Large Text Corpora,1991
[6] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/

