Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

(Special Issue for ITECE 2016)

TRIPLE EXTRACTION

Vishal Kansagra[1], Prof. Sourish Dasgupta[2], Darshan Thoria[3]

[1]Computer Science & Engineering, SLTIET

[2]ICT, DA-IICT

[3]Computer Science & Engineering, SLTIET

Abstract — Characterization of "Non-ISA" factual sentences can be used in Ontology learning. Characterization identifies subjects, objects and relations between them. It also identifies subject modifiers and object modifiers. Ontology Learning (OL) as a research field has been motivated by the possibility of automated generation of formal knowledge based on top of Natural Language (NL) document content so as to support reasoning based knowledge discovery. Most of the work done in this field has been made in Light-Weight OL, not much attempt has been made in Heavy-Weight OL. Ontology Learning is automated generation of ontologies from documents that contain natural language text. Characterize of "Non-ISA" factual sentences in English involve different stages like Triple Extraction, Normalize, Singularize and Characterization. In this paper we are going to focus on Triple-Extraction part which helps in characterization of sentences. In this module it converts Complex and compound "Non-Isa" into simple sentences. Characterization of Simple sentences are far easier than compound and complex sentences. This characterization can be useful to further convert a "Non-ISA" factual sentence in English into its equivalent Description Logic (DL), which is a part of Heavy weight OL, which makes the information retrieval very effective and reliable.

Keywords: Triple-Extraction, Characterization, Description Logic, Factual Sentences

I. INTRODUCTION

As mentioned Ontology Learning is broadly classified into two categories: (i) Light-Weight ontology and (ii) Heavy-Weight ontology [1]. Light-Weight ontology makes little or no use of axioms while Heavy-Weight ontology makes intensive use of axioms to define the domain knowledge. Figure given below gives brief overview of different types of ontology.

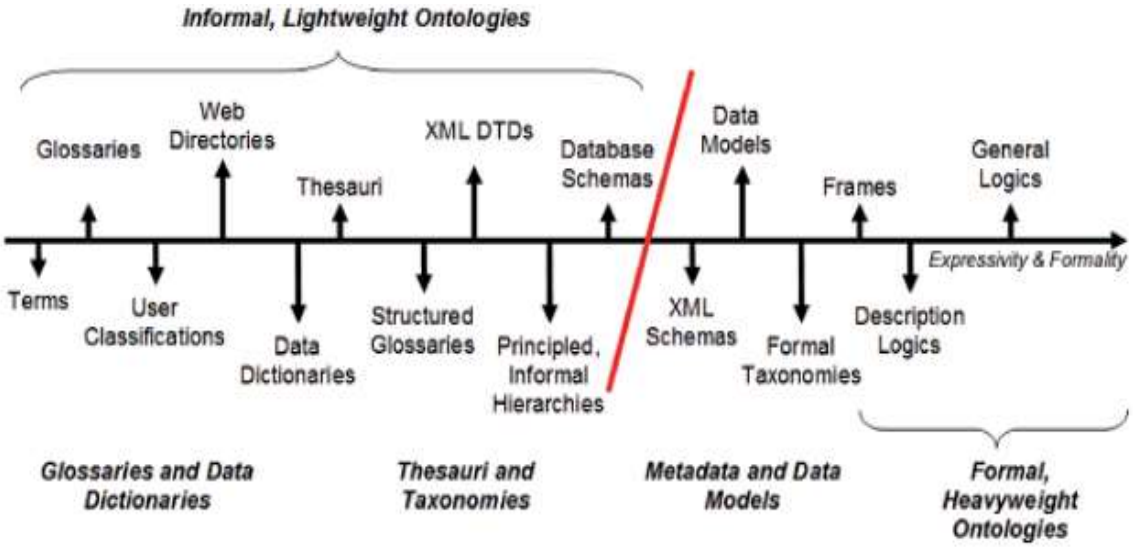

Figure 1: Ontology Spectrum: Wong, W., Liu, W., and Bennamoun, M. 2012. Ontology learning from text: A look back and into the future. ACM Compute. Surv. 44, 4 (Aug.2012), 20:1−20:36

II. BACKGROUND

2.1 Natural Language Processing Tools

During the characterization of "NON-ISA" english factual sentences few NLP tools were used. All the tools or the packages, namely NER, POS Tagger and lemmatizer, were off-the-shelf. Named Entity Recognition (NER) was used to identify the superclass of the given instance. For an example "Steve Job is an innovator" when served as an input to NER will tag "Steve" and "Job" as "Person". The tag so obtained are used for inducing the sentence. However categories used by NER are limited making it difficult to classify all possible instances. Part-Of-Speech Tagger is used to give part of speech to a word. Sentences serve as input to POS Tagger and tagged sentences are used to determine the templateto be selected. Example if we give Smoked as an input to the lemmatizer which will give an output as smoke. Stanford POS Tagger[2, 3] and Stanford NER[4] are used in characterization of Non-ISA factual sentence.

III. TRIPLE-EXTRACTION

3.1 Triple-Extraction

In this section we are going to study the algorithm for Triple-Extraction, analysis of Triple-Extraction algorithm, Example of Triple-Extraction and challenges of Triple-Extraction. After Extracting NonISA sentences from datasets, sentences are tagged using Stanford POS tagger. After tagging these sentences, tagged sentences are fed into Triple-Extraction module which converts complex and compound "Non-ISA" sentences into simple "Non-ISA" sentences. Triple-Extraction module is very useful because handling a simple "Non-ISA" sentences is very easy compared to compound and complex sentences. Triple-Extraction module first finds relation and it's index in the given sentence. By using relation and it's index Triple-Extraction module finds left halve of sentence (Subject part before relation) and right halve (Object part after relation) of sentence. After that it checks for "and" in left and right halves. If left halve and right halve(s) contain "and" then it breaks the string into individuals and stores these individuals into subjectlist and objectlist respectively. After that Triple-Extraction module combines both subjectlist and objectlist using relation and at last it gives simple sentences as an output.
Some Examples are given below in which complex and compound "Non-ISA sentences converted into simple "Non-ISA" sentences by Triple-Extraction module.
Examples - **Input** - Ram and Sam play football. **Output** - 1) Ram plays football. 2) Sam plays football. **Input** - Ram plays cricket and Sam plays football. **Output** - 1) Ram plays cricket. 2) Sam plays football. **Input** - Rohit and Kohli play good cricket and bad football. **Output** - 1) Rohit plays good cricket. 2) Rohit plays bad football. 3) Kohli plays good cricket. 4) Kohli plays bad football. **Input** - Rohit or Kohli play good cricket and bad football. **Output** - 1) Rohit or Kohli play good cricket. 2) Rohit or Kohli play bad football. **Input** - Rohit and Kohli play good cricket and bad football respective **Output** - 1) Rohit plays good cricket. 2) Kohli plays bad football. **Input** - John and Joe who, love Ice-Cream and chocolate, like cricket and basketball. **Output** - 1) John loves Ice-Cream. 2) John loves Chocolate. 3) John likes cricket. 4) John likes basketball. 5) Joe loves Ice-Cream. 6) Joe loves Chocolate. 7) Joe likes cricket. 8) Joe likes basketball.

3.2 Algorithm for Triple-Extraction

1: procedure GENERATETRIPLES
2: indexRel ← Index of relation
3: S ← leftHalve of sentence
4: if S contain and then splits string at and
5: SubjectList ← indivisuals
6: end if
7: O ← RightHalve of sentence
8: if O contain second Relation then
9: temp ← String from relation to second relation
10: temp1 ← remaing part of String
11: end if
12: if temp contain and then splits string at and
13: Objectlist ← indivisuals
14: end if
15: if temp1 contain and then splits string at and
16: Objectlist1 ← indivisuals

17: end if
18: ArrayList a
19: for i = SubjectList do
20: for J = ObjectList do
21: Stmt ← SubjectList(i) + Relation + ObjectList(j)
22: a ← Stmt
23: end for
24: end for
25: for i = SubjectList do
26: for J = ObjectList1 do
27: Stmt ← SubjectList(i) + SecondRelation + ObjectList1(j)
28: a ← Stmt
29: end for
30: end for
31: Return A

3.3 Analysis of Algorithm

Run time complexity of this Triple-Extraction algorithm depends on size of the subjectlist and objectlist. For a single sentence if sentence contains n subject and n object then the size of subjectlist and objectlist becomes n each. There are two for loops nested in algorithm two times hence it's runtime complexity is ($n^2 + n^2$). Hence runtime complexity of this algorithm is $O(n^2)$

3.4 Challenges in Triple-Extraction

There are many cases which won't work with Triple-Extraction module. Some of the cases are mentioned below. For Example John is student body and placement body member In which correct output should be 1. John is student body member 2. John is placement body member But it gives output as 1. John is student body 2. John is placement body member. For sentences like Ram and Sam play cricket and football and John and Joe play basketball and hockey, present Triple-Extraction module will not work.

IV. RESULT AND ANALYSIS.

4.1 Evalution on English Wikipedia Data Set

In English wikipedia Data set there are 255 complex sentences from which our algorithm converts 245 sentences into simple sentences and form 328 compound sentences it convetrs 315 sucssesfully.
In Complex sentences, some sentences containing a clause beginning with "which" give wrong output. Example is giving below. **Input**:- Every cat, which is wild, drinks milk. **Output** (System generated):- Every cat is Wild. Every cat drinks milk. **Output** (Expected):- Every wild cat drinks milk. Right Now we are working on these type of sentences.For Compound sentences, sentences combining more than two independent sentences, for example one given below, give wrong output. Ram and Sam play cricket and football and John and Joe play basketball and hockey. Right Now we are also working on improving the algorithm to work successfully on sentences like the one given above.

V. CONCLUSION

In this paper we have analyzed by using Triple-Extraction module we can make the characteriztion process easier, which converts the complex and compound "Non-ISA" factual english sentences into simple sentences. By using charactrization we can automatically converts the english factual sentences into Ontology. This automatically generated ontology will help in machine intelligence field, information retrival field and Artificial Inteligence filed.

REFRENCES

[1] W. Wong, W. Liu, and M. Bennamoun, "Ontology learning from text: A look back and into the future," ACM Computing Surveys (CSUR), vol. 44, no. 4, p. 20, 2012.
[2] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich partof-speech tagging with a cyclic dependency network," in Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, 2003, pp. 173–180.

[3] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13. Association for Computational Linguistics, 2000, pp. 63–70
[4] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005, pp. 363–370
[5] "ENGLISH WIKIPEDIA DATA SET," http://homepages.inf.ed.ac.uk/kwoodsen/data/wiki-rev-data.tar.gz.