BOOTSTRAP: Mining of Comparable Things

Kanchan Kundgir, Poonam Dere, Surabhi Mohite, Priti Mithari

Dr. D.Y. Patil Institute of Engineering and Technology, mohitesurabhi3@gmail.com, 7387841278

Abstract— Making Comparisons between things is a typical part of human decision making process. But however, it is difficult to know what are to be compared and what can be the alternatives. For e.g., if someone is interested in certain products such as digital cameras, then he /she would want to know what the alternatives are and compare different cameras before making any purchase. This type of comparison activity is very common in our daily life but requires high knowledge skill in order to make much better choice. Therefore, to address this difficulty, we are presenting a system to automatically mine comparable entities from comparative questions that users posted online. In this paper, we focus on finding a set of comparable entities provided a user's input entity. For example, provided an entity like Nokia N95 (a mobile phone), we want to find comparable entities such as Nokia N82, Blackberry and so on. To ensure high precision and high recall, we are developing a system that uses weakly-supervised bootstrapping method for comparative question identification and comparable entity extraction by leveraging a large online question archive. Our system calculates the precision and recall for a particular session depending on the correct comparators suggested. The result varies depending on the sessions. The results will prove to be very useful in helping users' exploration of alternative choices by suggesting comparable entities based on other users' prior requests.

Keywords— Bootstrapping method, Comparable entity mining, Information extraction, Part Of Speech Tags, sequential pattern mining, comparators, Weakly supervised mining, Precision, Recall.

INTRODUCTION

In decision-making process, comparing alternative options is one of the necessary steps that we carry out daily. But this activity requires high knowledge expertise to make better choice. For instance, while doing shopping of a laptop one must have detailed knowledge of its specifications like Processor, Storage, Graphics, Memory, Display, etc. In such cases, it becomes difficult for an individual with insufficient knowledge to make a good decision on which laptop to buy and also comparing the alternative options for the same.

Magazines such as PC Magazine, Consumer Reports and online media like CNet.com which makes efforts in providing editorial comparison content and surveys. The comparison activity in the World Wide Web normally involves- search for applicable web pages enclosing information regarding the targeted products, discovering competing products, and recognizing their pros and cons. Our focus, in this paper, is on finding a set of comparable entities provided a user's input entity. For e.g., provided an entity like Nokia N95 (mobile phone), we would want to find entities that are comparable like iPhone, Blackberry, Nokia N82, HTC and etc.In order to extract comparable entities from relative matter, we first need to find out whether the question is relative or not.

Our effort on comparable entity mining is related to the study on entity and relation removal in information extraction. According to our definition, a comparative question has to be a query with intention to contrast at least two entities. We exploit this insight and develop a weakly supervised bootstrapping means to identify comparative questions and extract comparable entities at the same time.

- *Comparative questions*: A question whose purpose is to compare two or more entities and these entities are explicitly mentioned in the question.
- *Comparator*: An entity in a comparative question which is to be compared [3].

According to the definitions, Q1 & Q2 below are not comparative questions whereas Q3 is. "Mumbai" and "Pune" are comparators.

Q1. "Which one is better?"
Q2. "Is Pune the best city?"
Q3. "Which city is better Mumbai or Pune?"

The results will be very useful in helping users' exploration of alternative choices by suggesting them comparable entities based on other previous users' requests.

SYSTEM OBJECTIVES

Project planning involves project scope which includes the determination and the documentation of a list of specific project objectives, tasks, deliverables, costs and deadlines. The following is the scope of our project:

1. To develop a system that will automatically mine comparable entities from comparative questionsthat users posted online so as to help them making better choices by suggesting alternatives.
2. The goal of this system is mining comparators from comparative questions and furthermore,provides and rank comparable entities for a user's input entity appropriately.
3. The objective of this system is helping users exploration of alternative choices by suggestingcomparable entities based on other users prior requests.

RELATED WORK

1. Overview

In terms of discovering related items for an entity, their work is similar to the research on recommender systems, to recommend items to a user. Recommender systems are similar between items and/or their statistical correlations in user log data [4]. For example, Amazon recommends products to its customers based on their own previous purchase; similar customers' previous purchase, and similarity between products. Comparable item is not equivalent to Recommending an item for finding customer item. In Amazon, the purpose of recommendation is to entice their customers to add more items to their shopping carts by suggesting similar or related items.

In the case of comparison, they help users explore alternatives, i.e., helping them make a decision among comparable items. For example, it is reasonable to recommend "iPod speaker" or "iPod batteries" if a user is interested in "iPod," but they are not comparing them with "iPod." However, items that are comparable with "iPod" such as "iPhone" or "PSP" which were found in comparative questions posted by users are difficult to be predicted simply based on item similarity between them. Although they are all music players, "iPhone" is mainly a mobile phone, and "PSP" is mainly a portable game device. They are similar but also different therefore beg comparison with each other. It is clear that comparator mining and item recommendation are related but not the same. Their comparator mining is related to then research on entity and relation extraction in information extraction [4], [6],[7].

2. Supervised Comparative Mining Method

Major contribution of authors Jindal and Liu (J&L) on mining comparative sentences and relations, in their system used class sequential rules (CSR) and label sequential rules (LSR). CSR maps a sequence pattern $S(s_1, s_2,.....s_n)$ to class C. Class C is either comparative or non-comparative question .and LSR maps an input sequence pattern $S(s_1s_2...s_i...s_n)$ to a labeled sequence $S'(s_1s_2...s_i...s_n)$by replacing one token s_i in the input sequence with a designated label (l_i). This token is referred as the anchor [7], [8].

J&L work on this method and treated comparative sentence identification as a classification problem and comparative relation extraction is called as an information extraction problem. They first manually created a set of 83 keywords is similar to the indicators of comparative sentences. These keywords were then used as pivots to create part-of-speech (POS) sequence data. The Table 1 below shows brief view of the Literature Survey [1].

The following were the drawbacks of this method:

i. The performance of J&L's method depends mainly on a set of keywords which are an indicative of comparative sentence.
ii. Because users can express comparative sentences or questions in many different ways and to have a high recall, a large annotated training corpus is required which makes this an expensive process
iii. CSRs and LSRs introduced by J&L was mostly a combination of POS tags and keywords. In spite of all these, it is a surprise that their rules achieved high precision but low recall.[11]

TABLE I

LITERATURE SURVEY[1]

PROPOSED SYSTEM

Sr. no	Paper Name	Conference	Approaches	Advantages	Disadvantages
1	Identifying Comparative Sentences in Text Documents.	ACM SIGIR Conf. Research and Development in Information Retrieval, 2006.	Combination of class sequential rule (CSR) mining and machine learning [8].	Extract comparative sentences from text is useful for many applications	It can achieve high precision but suffer from low recall.
2	Mining Comparative Sentences and Relations	Artificial Intelligence (AAAI ’06), 2006.	Identify comparative sentences from the texts and to extract comparative relations to its identified comparative sentences [9].	Evaluating an entity or event is to directly compare it with a similar entity or event.	It can achieve high precision but gives low recall.
3	Comparable Entity Mining from Comparative Questions	Proc. 48th Ann. Meeting of the Assoc. for Computational Linguistics (ACL ’10), 2010.	Mining the comparators from given entities of comparative questions [5].	Identifies comparative questions and extracts that comparators simultaneously using one single pattern	Their rules achieved high precision but low recall.
4	Relational Learning of Pattern Match Rules for Information Extraction	Proc. 16th Nat’l Conf. Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence (AAAI ’99/IAAI ’99), 1999.	Desired information can be extracted from natural language texts [6].	It can be research on relation and entity extraction in information extraction	The learned patterns employ limited syntactic and semantic information to identify potential slot fillers and their surrounding context.
5	Mining Knowledge from Text Using Information Extraction.	ACM SIGKDD Exploration Newsletter, vol. 7, no. 1, pp. 3-10, 2005.	Information extraction extracts structured data or knowledge from unstructured text [10].	Information Extraction is extracting structured data from unstructured or semi-structured web pages.	Cannot reduce demanding corpus-building requirements of information system.
6	Learning Surface Text Patterns for a Question Answering System	Proc. 40th Ann. Meeting on Assoc. for Computational Linguistics (ACL ’02), pp. 41-47, 2002	Automatically learning such regular expressions from the web, for given types of questions [11].	Their system assumes each sentence to be a simple sequence of words & searches for repeated word orderings as evidence for useful answer phrases.	The system does not classify or make any distinction between upper and lower case letters.

In our proposed system, we perform two main activities:

1. Comparator mining

2. Comparator ranking

1. COMPARATOR MINING

For mining Comparators from comparative questions, we've used **weakly supervised method**.

Weakly Supervised method for Comparator Mining: In our approach, a sequential pattern is defined as a sequence S ($s_1s_2...s_i...s_n$) where s_i can be a word, a POS tag, or a symbol denoting either a comparator ($C), or the beginning (#start) or the end of a question (#end). A sequential pattern is called an indicative extraction pattern (IEP) if it can be used to identify comparative questions and extract comparators in them with high reliability[1].

1.1 Mining Indicative Extraction Patterns

Our **weakly supervised method** is based on two assumptions:

1) If a sequential pattern can be used to extract many reliable comparator pairs then it is very likely to be an IEP.
2) The pair is capable to compare if a comparator pair can be extracted by an IEP.

Based on these two assumptions, we design our boot- strapping algorithm. There are two key steps in this method:

1) Pattern generation
2) Pattern evaluation

1.1.1 Pattern generation

The three kinds of sequential patterns are generated from sequences of questions are as follows [5]:

i) Lexical patterns- Lexical patterns indicate sequential patterns consisting of only words and symbols ($C, #start, and #end) .

ii) Generalized patterns- A lexical pattern can be too specific. So we generalize lexical patterns by replacing one or more words their POS tags.

iii) Specialized patterns- we perform pattern specialization by adding POS tags to all comparator slots. For example, from the lexical pattern '<$C or $C>' and the question 'ipod or zune?', '<$C=NN or $C=NN?>' will be produced as a specialized pattern [3].

1.1.2 Pattern evaluation

According to the following equation, the reliability score R^k(pi) for a candidate pattern p_i at iteration k is calculated:

$$\mathbf{R^k(p_i)} = \frac{\sum_{\forall cp_j \in CP^{k-1}} N_Q(p_i \to cp_j)}{N_Q(p_i \to *)}$$

Where,

$R^k(p_i)$ = Reliability score at iteration
P_i= Candidate pattern,
cp_j= known reliable comparator pairs,
CP_{k-1}= reliable comparator pair repository accumulated until the (k-1)th iteration,
$N_Q(x)$ = number of questions satisfying a condition x,
$p_i \to cp_j$= cp_j can be extracted from a question by applying pattern p_i,
$p_i \to *$= any question containing p_i

All the candidate patterns are evaluated and the pattern whose reliability score is greater than threshold γ is stored as IEPs in IEP database.[3]

1.2 Comparator extraction

By applying learned IEPs, we can easily identify comparative questions and collect comparator pairs from comparative questions existing in the question repository.Given a question and an IEP, the details of the process for Comparator extractions are as follows:

1. Generate sequence for the comparative question. If the IEP is a pattern without generalization, we just need to tokenize the questions and the sequence is the list of resulted token

2. If IEP is a specialized pattern, the POStag of extracted comparators should follow the constraints specified by the pattern.

According to above observation, we examined the following strategies:

1. Random strategy

2. Maximum length strategy
3. Maximum reliability strategy[3]

2. COMPARATOR RANKING

The remaining issue is to rank possible comparators for a user input. The ranking method that we have used is **Comparability-Based Ranking Method.**

A comparator would be more interesting if it is compared with the entity more frequently.
Based on this , we define a simple ranking function Rfreq (c;e) which ranks comparators according to the number of times that a comparator c is compared to the user's input e in comparative question archive Q:

$$R_{freq}\ (c;e) = N(Q_{c,e})$$

Where $Q_{c;q}$ is a set of questions from which c and e can be extracted as a comparator pair.
This ranking function can also be called as Frequency-based Method. [3]

DESIGN AND IMPLEMENTATION

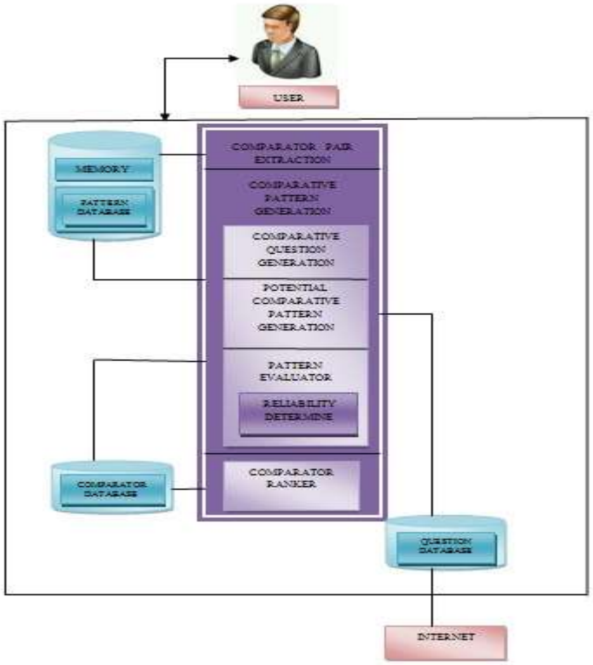

USER
MEMORY
PATTERN DATABASE
COMPARATOR PAIR EXTRACTION
COMPARATIVE PATTERN GENERATION
COMPARATIVE QUESTION GENERATION
POTENTIAL COMPARATIVE PATTERN GENERATION
PATTERN EVALUATOR
RELIABILITY DETERMINE
COMPARATOR DATABASE
COMPARATOR RANKER
QUESTION DATABASE
INTERNET

FIGURE: PROPOSED SYSTEM CONSISTING OF SEVERAL SUBSYSTEMS

OUR SYSTEM CONSISTS OF THE FOLLOWING MODULES:

1. **Algorithm**: This algorithm accepts question queue and comparator queue which is extracted from the question dataset. This algorithm check in the comparator queue if it contains any existing comparator i.e. which is already present in database. If is already present, it is removed from the queue. Same thing is followed for question queue.
   Then comparator queue and question queue are stored into the database.

2. **Database** contains DatabaseManager.
   DatabaseManager is responsible for all the database operations. DatabaseManager is interface between the project and the database (MySQL in our case)
   DatabaseManager has methods to
   - add comparator in the database.
   - add question into repository
   - check if a word is comparator
   - extract POS tag for a word

- extract pattern depending on POS for a word
- get count for a comparator
- save comparator for a comparator
- save comparator for a question
- extract comparator from a comparator pair with one of the comparator of pair as input (i.e. recommendation)

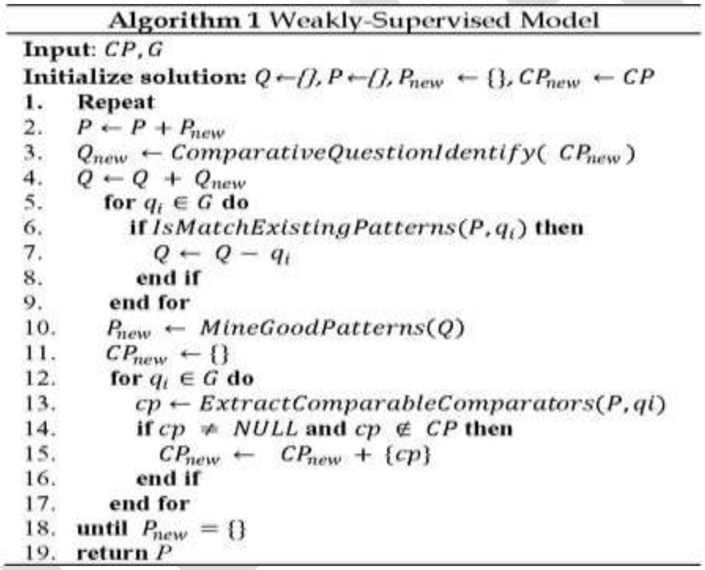
3. **Pattern Generator** which generates the 3 required pattern for Bootstrap algorithm:
   - Lexical Pattern
   - Specialised Pattern
   - Generalised Pattern

4. **Model** contains model class for comparator pair. Model class hold the information of comparator extracted from database. Model class helps us to have operations on the objects get easily executed.

5. **GUI** takes care of the UI part of the system. Has responsibility to show the UI and accept the user interaction as input. The UI components helps to load dataset into question, execute Bootstrap, show recommended suggestion, show pattern for question.

6. **Recommendation** is responsible to provide UI for recommendation system. It takes input as comparator and displays recommendation for comparator in the descending order of frequency count.

ALGORITHM



```
Algorithm 1 Weakly-Supervised Model
Input: $CP, G$
Initialize solution: $Q \leftarrow \{\}, P \leftarrow \{\}, P_{new} \leftarrow \{\}, CP_{new} \leftarrow CP$
Repeat
  $P \leftarrow P + P_{new}$
  $Q_{new} \leftarrow ComparativeQuestionIdentify(CP_{new})$
  $Q \leftarrow Q + Q_{new}$
    for $q_i \in G$ do
      if $IsMatchExistingPatterns(P, q_i)$ then
        $Q \leftarrow Q - q_i$
      end if
    end for
    $P_{new} \leftarrow MineGoodPatterns(Q)$
    $CP_{new} \leftarrow \{\}$
    for $q_i \in G$ do
      $cp \leftarrow ExtractComparableComparators(P, qi)$
      if $cp \neq NULL$ and $cp \notin CP$ then
        $CP_{new} \leftarrow CP_{new} + \{cp\}$
      end if
    end for
until $P_{new} = \{\}$
return $P$
```

FIGURE: PSEUDOCODE OF THE BOOTSTRAPPING ALGORITHM[3].

STEPS FOR BOOTSTRAPPING ALGORITHM:

i. The bootstrapping process starts with a single IEP.
ii. Then extract a set of initial seed comparator pairs from it.
iii. For each comparator pair all questions containing the pair are retrieved from a question collection and regarded as comparative questions.
iv. From comparative questions and comparator pairs all possible sequential patterns are generated and evaluated by measuring their reliability score defined in the Pattern Evaluation.
v. Patterns evaluated as reliable are IEPs and are added into an IEP repository.

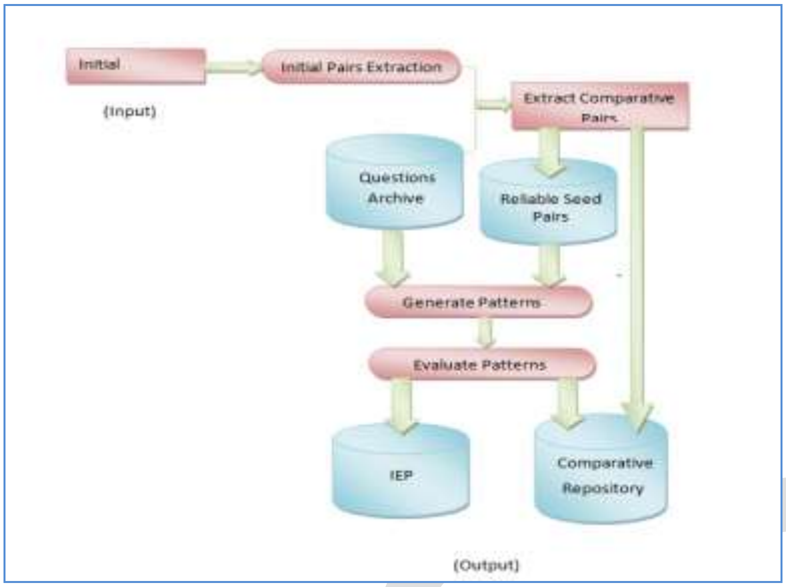


FIGURE: BOOTSTRAPPING ALGORITHM

EXPERIMENTAL RESULTS

Precision: It is the ratio of number of relevant comparable entities retrieved to the total number of irrelevant and relevant comparable entities retrieved. It has been expressed in percentage as:
Precision%=A/(A+C)*100% where,
A - number of relevant comparable entities retrieved,
C - number of irrelevantentities retrieved.

Recall: It is the ratio of number of relevant comparable entities retrievedto the total number of relevant comparable entities in the database.It has been expressed in percentage as:
Recall%=A/(A+B)*100 % where,
A - number of relevant comparable entities retrieved,
B - number of relevant entities not retrieved.

TABLE II
EXPERIMENTAL RESULTS

Sr. No.	Question(Input)	Comparators retrieved(Output)	Precision (%)	Recall (%)
1.	Which city is better, Pune or Baramati?	Sangali Satara iOS Paris Bhopal Nashik Pune	83.33	46.15

2.	Which mobile is better, samsung or iphone?	Nokia apple Apple nokia Samsung	100.0	80.0
3.	Which is better nokia or iphone?	Samsung samsung nokia	100.0	50.0
4.	Which is better Pune or Satara?	Sangali Baramati iOS Paris Bhopal	80.0	41.67
5.	Which city is better NYC or Pune?	Sangali Satara Baramati iOS Paris Bhopal NYC	83.33	50.0

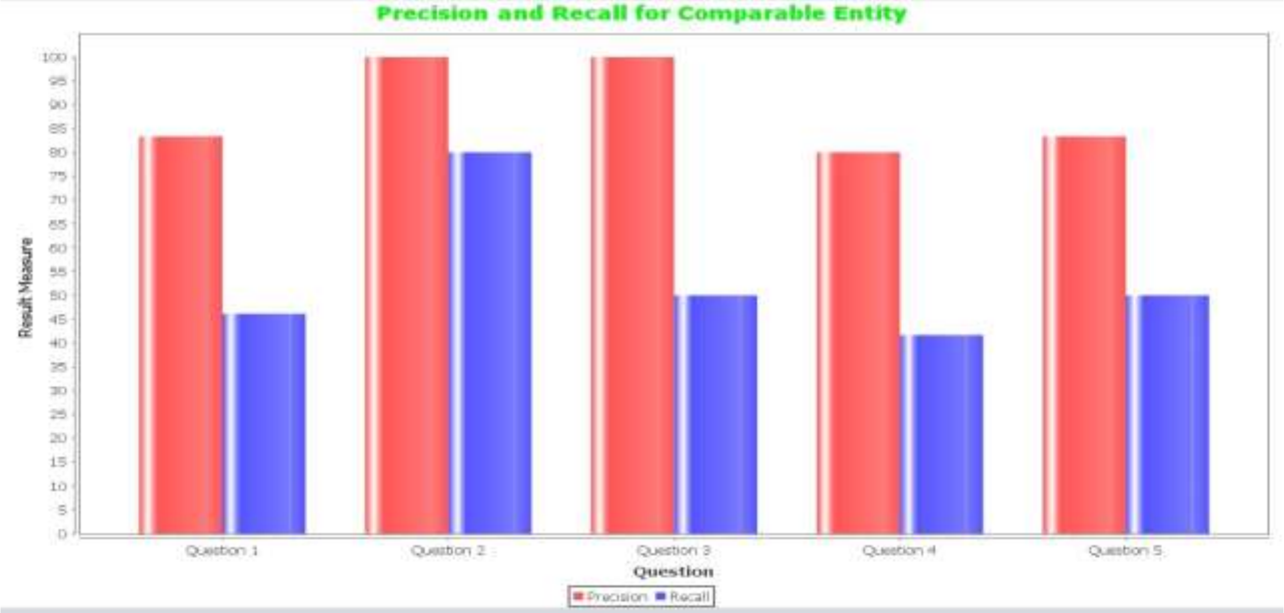


FIGURE: BAR GRAPH OF EXPERIMENTAL RESULTS FOR PRECISION AND RECALL

ACKNOWLEDGMENT

The authors wish to thank the researchers, publishers for making their resources available and teachers who provided them valuable assistance to the writing of the research summarized. Finally, they extend their heartfelt gratitude to their friends for assisting in the collection of the topics and their family for the help and inspiration extended. This paper was presented in part at International Journal of Advancement in Engineering Technology, Management And Applied Science,September 14 with paper title “A Study on Comparable Entity Mining from Comparative Questions”.

CONCLUSION

In this paper, we present a new supervised method for identifying comparative questions and extraction of comparator pairs at the same time along with providing entities in rank. Our proposed system’s key insight is that a good comparative question identification pattern should extract good comparator pairs, and a good comparator pair should occur in good comparative questions to bootstrap the extraction and identification process. This method considerably improves recall in together tasks whilst maintain elevated precision. Our system has user-friendly GUI.

Comparator mining outcome can be useful for commerce exploration or product recommendation organization. For instance, automatic proposition of comparable entities can help out users in their assessment activities earlier than building their acquire decision. In addition, the outcome can make available helpful information to companies which would like to recognize their competitors.

REFERENCES:

[1] Kanchan Kundgir, Poonam Dere, Surabhi Mohite, Priti Mithari, “A Study on Comparable Entity Mining from Comparative Questions”, International Journal of Advancement in Engineering Technology, Management And Applied Science, vol. 1 Issue 4, ISSN-number 2349-3224, September 14.

[2] Kanchan Kundgir, Poonam Dere, Surabhi Mohite, Priti Mithari, “Research on Comparable Entity Mining from Comparative Questions”, National Conference on "Advancements in Computer and Information Technology” (NCACIT-15), 2015

[3] Shasha Li, Chin-Yew Lin, Young-In Song, and Zhoujun Li, “Comparable Entity Mining from Comparative Questions,” IEEE Transactions On Knowledge And Data Engineering, vol. 25, no. 7, 1498-1509, 2013.

[4] Shasha Li, Chin-Yew Lin, Young-In Song, and Zhoujun Li, “Comparable Entity Mining from Comparative Questions,” Proc. 48th Ann. Meeting of the Assoc. for Computational Linguistics (ACL ’10), 2010.

[5] M.E. Califf and R.J. Mooney, “Relational Learning of Pattern- Match Rules for Information Extraction,” Proc. 16th Nat’l Conf. Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence (AAAI ’99/IAAI ’99), 1999.

[6] C. Cardie, “Empirical Methods in Information Extraction,” Artificial Intelligence Magazine, vol. 18, 1997, 65-79.

[7] N. Jindal and B. Liu, "Identifying Comparative Sentences in Text Documents," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '06), 2006, 244-251.

[8] N. Jindal and B. Liu, "Mining Comparative Sentences and Relations," Proc. 21st Nat'l Conf. Artificial Intelligence (AAAI '06), 2006.

[9] R.J. Mooney and R. Bunescu, "Mining Knowledge from Text Using Information Extraction," ACM SIGKDD Exploration Newsletter, vol. 7, no. 1, 2005, 3-10.

[10] D. Ravichandran and E. Hovy, "Learning Surface Text Patterns for a Question Answering System," Proc. 40th Ann. Meeting on Assoc. for Computational Linguistics (ACL '02), pp. 41-47, 2002.

[11] Shrutika Narayane, Sudipta Giri, "A Review on Comparable Entity Mining", Vol. 2, Issue 12, International Journal of Innovative Research in Computer and Communication Engineering, December 2014.

[12] Lanka Sri Naga Deepthi, Y.L.Sandhya Rani, "The Use of Comparative Questions in Comparable Entity Mining", Vol. 5, Issue 4, International Journal of Computer Science And Technology, Oct - Dec 2014