ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

Enhancing Comparable Entity Mining from Comparative Questions

Shrutika Narayane[1] and Sudipta Giri[2]

ME student, Department of Information Technology, MIT, College of Engineering, Kothrud, Pune, India[1]

Assistant Professor, Department of Information Technology, MIT, College of Engineering, Kothrud, Pune, India[2]

Abstract: It is a recognized fact that human beings usually tend to compare one thing with the other which is typical part of human decision making process. But, this is difficult to know; what are to be compared and what can be the alternatives. This alternatives foundation process is very much difficult to cipher out what user really wants. This comparison activity is very common in our daily life but requires high knowledge skill. We present a novel way of automatically mining comparable entities from comparative questions in order to address this difficulty overcoming some performance issues of existing system. The results will be very useful in helping user's exploration of alternative choices by suggesting comparable entities based on other users' prior requests. The proposed system overcomes these drawbacks and improves the efficiency of mining comparators. Therefore, in order to assure high precision, recall and ambiguity resolution, we propose a hybrid approach that combines the Bootstrapping, heuristic rules, pattern matching, and association rules combination. Association rules gives Support and Confidence count of each comparable entity pair aims to extract frequently compared entity pair, interesting correlations from archived questions. The primary goal of this project is to identify comparative question and then we pull out its comparable entities from the query by making use of an extensive question record. Initially, the user presents a query as an input; later the system will identify whether the fired question is comparable or not. Once the system verifies that the query is comparative; the required entities are extracted, and the output is presented to the user with the possible alternative options along with their ranking in repository. This approach provides better results compared to the existing approach.

Keywords: Information extraction, bootstrapping, sequential pattern mining, comparable entity (comparator) mining, POS tagger and association rule.

I. INTRODUCTION

In day to day life people come across a situation that they must decide upon one thing among the several, while purchasing. For better selection, user probably attempt to compare entities that they are interesting in. In assistance with decision making, comparing various entities having common utility but with distinguishing features plays crucial role to make better decision. When we fire a query to the search engine like Google for comparing the entities, it returns the results containing several web pages, web links for making comparisons. It needs to go through each web page separately to figure out what will be the best choice, which requires a lot of efforts and analysis resulting towards better decision. We can achieve the same objectives without going through each and every web page manually. Henceforth to make decision process simple and easy there is a need of novel recommendation system which can extract many relevant things related to the end user query from the database containing questions posted by online users. Comparative question and its comparable entities which are explicitly mentioned in question are two main components of decision making process [1].

Comparative questions: A question intended to compare entities with similar utility.

Comparators: Target entities in a comparative question which are to be compared are comparative entities or also called as comparators.

In the following example Q1 & Q2 are not comparative questions whereas Q3 is comparative question in which "Pune" and "Bangalore" are comparators.

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

The outcomes of these comparative questions will be very useful in helping user's exploration i.e. recommending various alternative choices by suggesting comparable entities on the basis of other previous online user's requests. Knowing popularity of comparator pairs among the users for market analysis is useful for leading companies to know their competitor in market.

II. RELATED WORK

Initially basic relevant work is done by Jindal & Liu on comparative sentences mining and their relations [2] that was Supervised learning tends to the machine learning task of containing a function from labelled training sets of data. The training data consist of a set of training examples and uses the class sequential rules (CSR) and label sequential rules (LSR) to identify comparative sentences and extract comparative relations [2].

CSR is a classification rule which maps a sequence pattern *S (s*1*, s*2 . . . *sn)* (a class *C. C* is either comparative or non-competitive).

DOI 10.17148/IJARCCE.2015.41030

LSR maps an input sequence pattern *S (s1, s2 . . . si . . . sn)* to a labelled sequence *S (s1, s2 . . . li . . . sn)* by replacing token *si* in the input sequence with a designated label *(li)* and this token is referred as the anchor.

But J & L's method have some drawbacks like limited domains and require large amount of keywords indicating comparative sentences. Firstly they manually created a set of 83 keywords like exceed, beat, outperform and better that are indicators of comparative questions [2]. Evaluation of an entity or event is directly comparing it with a similar entity or event. Identification of comparative sentences from texts and to mine comparative relations from its identified comparative sentences, it can achieve high precision but gives low recall.

However, supervised training for exact entity and relation extraction is expensive, requiring a substantial number of labelled training sets for each type of entity and relation to be extracted [3]. Because of this, researchers have explored semi-supervised learning methods that use small number of labelled examples of the predicate to be extracted, along with a large volume of unlabelled text [4], [5]. Whereas bootstrapping method is very significant one in previous information mining research [6], also referred as weakly supervised bootstrapping technique, significant to extract comparable entities with highly precise manner [1] i.e. with high recall as well as high precision preferred. Kai-Sheng, Chun-Cheng and Yuen-Hsien [7] proposed a system entity mining based on only part-of-speech tagging. This technique was applicable to cipher out the criminal acts, information litigation information and investigation clues by the law enforcement team. To achieve this, a network is built for entity related visualization and exploration.

III.IMPLEMENTATION DETAILS

We are proposing a novel way of pattern generation & pattern evaluation for identifying comparative questions and extract their comparator pairs simultaneously. Basically we rely on two key insights that,

- A good comparative question identification pattern should extract good comparators,
- A good comparator pair should occur in good comparative questions to bootstrap the extraction and identification process.

We are using various data mining techniques to achieve our objectives aggregated bootstrapping method, graph based ranking technique, association mining rules are used to obtain better results.

A. Bootstrapping Method

Bootstrapping is a technique for comparative question identification and comparable entity extraction from user's input sequential string which is depicted in Fig.1. Pattern generation is main step where three kinds of patterns get covered, namely Lexical, Generalized and Specialized these are explained in next section.

The working steps of boot-strapping method are shown below:

Step 1: Scan the Storage database.
Step 2: Process the user input query.
Step 3: Match the input query pattern with existing one (it can be used to identify comparative questions and extract comparators from them).
Step 4: Identify Comparative and Non Comparative Questions.
Step 5: Extract comparable entities from comparative questions.

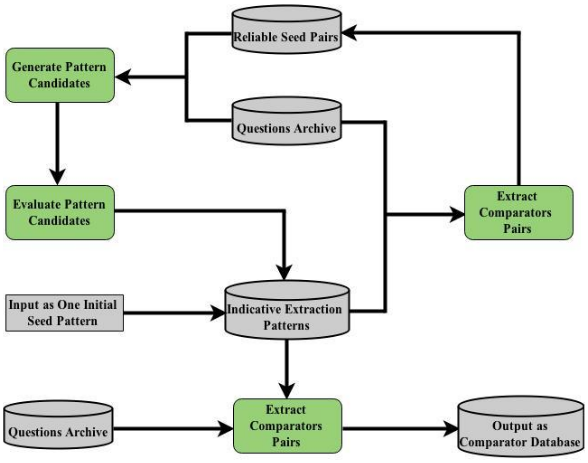
Reliable Seed Pairs
Generate Pattern Candidates
Questions Archive
Evaluate Pattern Candidates
Extract Comparators Pairs
Input as One Initial Seed Pattern
Indicative Extraction Patterns
Questions Archive
Extract Comparators Pairs
Output as Comparator Database

Fig. 1 Overview of Bootstrapping Method

After performing all the above steps, we will get comparative questions from user input queries and simultaneously comparators get extracted [1].

B. Covered Indicative Extraction Patterns

User's question is taken as a form of sequential pattern therefor Indicative Extraction Pattern (IEP) is a sequential pattern which we are using for identification of comparative questions along with its comparator extraction. A question is classified as a comparative question if it matches an IEP and the token sequences corresponding to the comparator slots in the IEP are extracted as comparators [1]. Comparative question is getting matched with following three types of pattern.

1. *Lexical patterns:*
   These patterns indicate sequential patterns consisting of only words and symbols ($C, #start, and #end).
   E.g. which is better, Pune or Bangalore?

2. *Generalized patterns:*
   A lexical pattern is too specific for matching. So lexical patterns are generalized by replacing one or more words their POS tags.
   E.g. which city is better, Pune or Bangalore?

3. *Specialized patterns:*
   Pattern specialization is done by adding POS tags to all comparator slots. For example, from the lexical pattern *'<$ or $C>'* and the question 'Paris or London?', *'<$C=NN or $C=NN?>'* will become specialized pattern.
   E.g. Pune or Bangalore?

DOI 10.17148/IJARCCE.2015.41030

C. Part-of-Speech Tagger

Part-of-speech (POS) taggers plays crucial role in our proposed technique to extract comparators [8]. In English grammar, part-of- speech of a word is a linguistic category defined by its syntactic behaviour. Common POS categories are noun, pronoun, verb, adjective, adverb, preposition, conjunction and interjection. Then there are so many categories and subcategories which rose from different forms of these categories. We use NLC POS taggers, Important POS tags to this work and their categories are:

NN: Noun, NNP: Proper Noun, PRP: Pronoun, VBZ: Verb, present tense, 3rd person singular, JJR: Comparative Adjective, JJS: Superlative Adjective, RBR: Comparative Adverb, RBS: Superlative Adverb. Although JJR, JJS, RBR, and RBS tags represent comparatives, many sentences containing such tags may or may not be comparisons [1]. Hence, we can't solely use or rely only on these tags for comparative sentences identification. A short table method called as phrase chunking where we change the POS tag of a word or merge words and mark them as one through a POS tag.

For instance, "this is better Nokia or Samsung". When the user enters this question into the system, the parser-chunking tagger first scans the question completely distinguish any portions of the speech related keywords [2] like Noun, Pronoun, Adjective, Verb and hence along. It is starting to classify "Apple, Samsung" as Nouns that are the comparators, "better" as the adjective, or as a coordinating conjunction and so on for whole question.

User submits a query to the system, in query comparative question identification phase; the query posted by the end user is treated as sequential pattern and is passed through the Parser-chunking tagger. Initially, system identifies whether the input query is a comparative question or not. In order to identify query is a comparative query, we make use of pattern for matching patterns with input query. If it get successful and identified as comparative; control successfully get transferred to the next phase of entity filtration where all the strings is removed except comparable entity pair of question.

Here, the comparable entity pairs are the user given input target entities of comparison which are explicitly mentioned in question. The filtration includes all the parts of speech related keywords in the input query such as better, good, which, who, he, this, that, than and so on. Once the entities are extracted from the input query the bootstrapping successfully approach to extract the relevant alternative queries for the end user. In order to achieve relevancy, initially it checks the existing data to identify any relevant matching comparators in database.

If it is unable to trace any results for that query from the Dataset, then the control is passed on to the database where the comparison is done to end the relevant alternatives and the outcome is displayed to the end user. The database contains comparable entity in the form of pairs which is recorded from all previous user's archived queries. Overall system suggest one preferable entity from user input entity pair, ranks that entity pair, recommend alternative comparators, support & confidence count of entity pair are given as shown in Fig.2.

D. Ambiguity Resolution

In all previous existing system were facing the problem while recommending comparators for ambiguous entities i.e. entity stands for more than one thing and failed to give accurate output which concerns to only that relevant item. For resolving ambiguity we assembled the datasets in systematic categorization manner. For that we are making analysis of each entity to cipher out whether entity is ambiguous or not by complete scanning to find presence of entity in datasets. It requires whole data set examination; if user entered entities presence in more than one set it gets detected as ambiguous one. By ciphering category of both the entities our system extracts comparators only from same data set is extracted. As "Paris" is ambiguous entity which stands for 'celebrity' as well as 'location'. System efficiently identifies it and resulting concern comparators related to that category only.

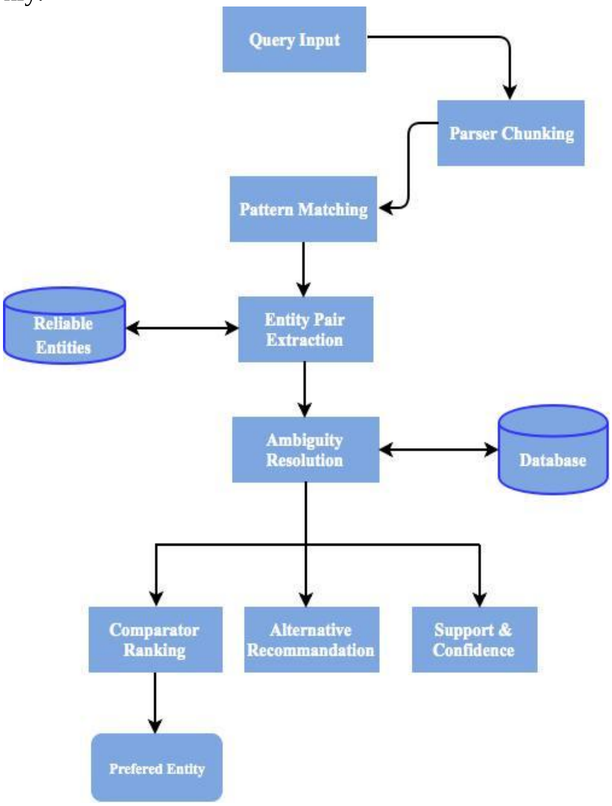

Fig. 2 Overview of System Workflow

E. Comparator Ranking

Ranking the comparators for a user's input depending upon their repeatability, frequency and represent ability in dataset is comparator ranking. A comparator can be considered as valuable comparator in ranking if it is compared to many other important comparators including the input entity. Ranking of comparators indicate the popularity in accordance of their frequency and represent ability [1].

DOI 10.17148/IJARCCE.2015.41030

We define a graph *G = (V, E)*. In this graph, *V* is the set of nodes *v*, which consists of comparable entities of the input. The edge e_{ij} between v_i and v_j means that the two Comparators which are compared in our comparator pair Repository.

Count of v_i, v_j is the frequency of comparator pairs v_i and v_j in our repository. A transition probability is defined as follows:

Comparability based and graph based ranking are mainly used ranking techniques. We used graph based ranking method for comparator ranking. Frequency is consider as efficient parameter for comparator ranking but the frequency-based ranking method can suffer when an user input occurs rarely in collection of questions; Hence in addition to it representing ability should also be considered. We regard a comparator representative if it is frequently used as a baseline while making comparison of interested entity. A comparator can be considered as valuable comparator in ranking if it is compared to too many other important comparators including the input entity.

F. Association Rule For Entity Mining

Association rules is well researched techniques of data mining. We are using association rules to extract interesting correlations, frequent entity pair comparison, in user's transaction databases. There are two basic measures for association rules, support (s) and confidence (c) [9]. Since the database of archived questions is large and admin concern only about those frequently compared by users. Thresholds of support and confidence are predefined by admin called as minimal confidence.

Two basic parameters of Association Rule Mining (ARM) are;, support and confidence. Where; *X*, *Y* is user's comparable entities which are explicitly mentioned in the question. Support(s) is how many transactions contain entity *X*, *Y* to the total number of transactions in the database; where, *X, Y* is a comparable entity pair in user's comparative questions. And Confidence of entity pair is Support of *(X, Y)* to the support of entity *X*. Formulas for Support and Confidence are as:

$$\text{Support (S)} = \frac{\text{Count of } (X, Y)}{\text{Total number of transactions}} \qquad (2)$$

$$\text{Confidence}(\text{X}, \text{Y}) = \frac{\text{Support of } (X, Y)}{\text{Support of } X} \qquad (3)$$

The count for each entity increases by one every time when same pair of comparable entity is encountered in different transaction in database during the scanning process of dataset of previously compared entity. The admin is interested in those entity pair, which is bought together frequently; a high support of entity is desired for more interesting association rules. Predefined minimum support is considered as a threshold, if support exceeds threshold value, it will become entity of interest.

IV. EXPERIMENTAL RESULTS

In this section the overall outcome along with experimental results of our system and existing system is compared. When user enters query, our system For user input query our system gives details regarding its status as comparative question or non-comparative one, then if both the user entered entities are belongs to same domain then category as same. As none of the entity is ambiguous it shows no ambiguity. Preference is according to the high rank system gives the preference to that entity which has high rank. Rank of both the entities is calculated by graph based method. Alternatives or recommendation for users entities are suggested by analyzing previously archived questions. Afterward Support and confidence for user defined entity is provided which shows the popularity of the current comparable entity pair of user.

$$P(v_i|v_j) = \frac{\text{Count}(v_i, v_j)}{\text{Count}(v_i, *)} \qquad (1)$$

First step is log in to the system as a user or administrator. After successful login system goes to the input query. When user entered ambiguous entity; our system successfully identifies it and gives relevant comparators without any confusion. For resolving ambiguity we assembled the datasets in systematic categorized manner. And we make analysis of each entity to cipher out whether entity is ambiguous or not by scanning presence of entity in datasets. As here "Paris" is ambiguous entity which stands for 'celebrity' as well as 'location'. System efficiently identifies it and resulting concern comparators only.

To evaluate the performance of our system, we manually labelled a random 2500 questions from Yahoo question answers category. This gave me a total of 500 questions of which 185 questions where comparative. For the comparative questions I also tagged the comparative pairs for each sentence. The dataset used for testing contained 500 questions; 500 questions randomly selected and tagged. Set-A contains both comparative and non-comparative questions and Set-B contains only comparative questions. We measured the precision, recall and f-score of comparative question. When system tested to calculate performance only in case of identification of comparative and non-comparative questions on Set(A+B) and compared with existing system. For comparator extraction from Set-B containing only comparative questions we calculated the precision, recall and F-score. When our system is compared with existing system (basic bootstrapping) process; it is nearby equivalent in comparative question identification but in case of comparator extraction it is much effective and gives high F-score compared to existing one. F-score count for identification of comparative questions and comparator extraction is as shown in Table 1.

Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records in database. Recall is the ratio of the number of

DOI 10.17148/IJARCCE.2015.41030

relevant records retrieved to the total Number of relevant records in the database.

TABLE 1: Comparison with Existing System

	Identification (Set A+B)		**Comparator Extraction (Set-B)**	
	Existing System	Our System	Existing System	Our System
Precision	0.89	0.92	0.76	0.87
Recall	0.87	0.89	0.72	0.89
F-Score	0.879	0.904	0.739	0.879

Overall system performance of our model with the existing is compared by averaging recall, precision, f-score values then we get graphical results as shown in Fig.3.

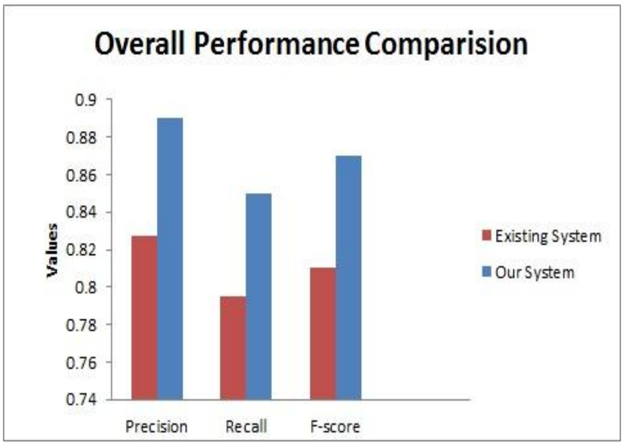

Fig. 3 Overall system performance comparison with existing system

V. CONCLUSION

Our System identifies comparative questions from users query and extracts its comparable entities simultaneously; which explicitly mentioned in question archived. Reliable alternative comparators are recommended for user's comparable entity pair. Comparator ranking is done from previous user's posted questions entities which make decision making more convenient. We used hybrid approach formed by aggregating bootstrapping, pattern matching and Association rules which significantly achieved high recall maintaining high precision while extracting the comparators. Ambiguous entity conflict is successfully resolved along with their relevant comparator extraction with high precision. Systematic way of data storage and analysis increased the overall performance and accuracy in comparator extraction. We rely on two key sights that good comparative question identification patterns extract good comparators, and a good comparator pair occurs in good comparative questions. Support and Confidence count indicate popular comparison entity pair in market. Comparator ranking is useful for manufacturing companies to know their competitors in the market. Also, comparator mining techniques can be used in many applications such as marketing intelligence, product benchmarking and e-commerce. In the future, we would like training datasets which will still improve the overall functioning of the data retrieval mechanism utilizing very less time. Mining of rare extraction patterns is still challenging task; such as, how to identify comparator aliases such as "HCL" and "Hindustan Computers Limited".

REFERENCES

[1] Li Shasha, Chin-Yew Lin, Young-In Song, and Zhoujun Li, "Comparable Entity Mining from Comparative Questions", Knowledge and Data Engineering, IEEE Transactions on 25, no. 7, pp. 1498-1509, 2013.

[2] Nitin Jindal and Bing Liu, "Identifying Comparative Sentences in Text Documents", Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 244-251, 2006.

[3] Mooney J. Raymond and Razvan Bunescu, "Mining Knowledge from Text Using Information Extraction", ACM SIGKDD explorations newsletter 7.1, pp. 3-10, 2005.

[4] Stephen Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text", Machine Learning, vol. 34, nos. 1-3, pp. 233-272, 1999.

[5] Carlson Andrew, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr, and Tom M. Mitchell, "Coupled Semi-supervised Learning for Information Extraction", Proceedings of the 3rd ACM international conference on Web search and data mining, pp. 101-110, 2010.

[6] Li Shasha, Chin-Yew Lin, Young-In Song, and Zhoujun Li, "Comparative Entity Mining", U.S. Patent no. 8, 484, 201, July 2013.

[7] Yang, Kai-Sheng, Chun-Cheng Chen, Yuen-Hsien Tseng, and Zih-Ping Ho, "Name entity extraction based on POS tagging for criminal information analysis and relation visualization", In Information Science and Service Science and Data Mining (ISSDM), 6th International Conference on New Trends in, pp. 785-789. IEEE, 2012.

[8] Li Shasha, Chin-Yew Lin, Young-In Song, and Zhoujun Li, "Comparable Entity Mining from Comparative Questions", Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL'10), pp. 650-658, 2010.

[9] Rashmi, A. G., "A Novel Method to Extract Comparison of Products Using Comparative Questions", International Journal of Computer Science & Engineering Technology (IJCSET), vol. 6 no. 5, pp. 311-316, 2015.

DOI 10.17148/IJARCCE.2015.41030