



INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A NOVEL APPROACH OF WEAKLY SUPERVISED MODEL FOR COMPARABLE ENTITY MINING

Tejaswini H. Patil*, Sonali Patil

*M.E.Student, Department of computer Engg. ACEM, Pune, India

* Asst. Prof., Department of computer Engg. ACEM, Pune, India

ABSTRACT

To compare things which are having same function but differ in their properties is part of human decision making process. Differentiating products having same job but differing in their capabilities or properties is difficult task. Consider someone want to purchase online mobile phone then he or she should know which mobile phone to compare with how many and which alternatives. This is common activity but requires high knowledge skills. So to deal with this difficulty we developed a novel way to automatically get comparable entities from comparative questions. The proposed system will consider comparators which are alises of each other. Given users input entity, we find comparable entities for that entity. This method achieves high precision and high recall. It is also going to rank results according to relevance to users requirements. This method achieves more than eighty percent F1-measure in both comparative question identification and comparable entity extraction.

KEYWORDS: Information extraction, bootstrapping, sequential pattern mining, comparable entity mining, Indicative Extraction Pattern, lexical pattern, generalized pattern, specialized pattern, Comparator alises.

INTRODUCTION

While making decision important step is to compare alternatives available and we carry out this task each day. But this require high knowledge skills. When using internet a comparison action involves searching for the web pages which are relevant i.e pages which contains information related to targeted product then searching details of product, comparing them to find out pros and cons. This is time consuming and not much efficient process. So we provide a weakly supervised method. [1] This method automatically mine comparable entities from comparative questions. The comparative questions posted online by user are stored in question collection. This method achieves high precision and also maintain high recall. A question is said to be comparative question if it compare at least two entities. Please note that a question containing at least two entities is not a comparative question if it does not have comparison intention. So two things are important those are 1. Minimum two entities 2. Comparison intention. However, we observe that a question is very likely to be a comparative question if it contains at least two entities. A weakly supervised method is used for this purpose. We define two terms here. 1. Comparative Question-A question comparing two or more entities. 2. Comparator-The entities which are target of comparison in comparative questions.

Example- Q.1 Which phone is better Nokia N85 or iPhone?

Q 2. Whether Nikon camera is best camera.

First question compare two entities so the question is comparative question. The comparators in question are Nokia N 85 and iPhone. But the second question is not comparative question because it does not compare two entities. The goal of work is mining comparators from comparative questions, to provide comparable entities. for users input entity and also rank comparable entities for users input entity. But the approach presented here is not capable of identifying comparator alises i.e if comparative question includes comparator alises e.g. What is difference between HCL and Hindustan Corporation Limited? So we improve this method so that it will identify comparator alises i.e when comparative question contains alises, our method will first check question for alises and then only proceed to next step.

RELATED WORK

Our work is related to research on recommender system [2]. In recommender systems when customer purchase a product, system will recommend other products by observing his/her trend in purchase. For example if customer

purchase laptop then system will recommend laptop batteries. But recommending an item is different concept than comparable entity mining.

Our work is also related to entity and relation extraction in information extraction[3][4][5][6][7]. The more relevant work is mining comparative sentences and relations[8][9]. In this work they used Class Sequential Rules(CSRs)[8] and Label Sequential Rules(LSRs)[8] to identify comparative sentences and extract relations. This method achieves high precision but it is having low recall[9].

CSR is a classification rule. It maps a sequence pattern $S(s_1s_2 \dots s_n)$ a class C . In our problem, C is either comparative or noncomparative. LSR is a labeling rule. It maps an input sequence pattern $S(s_1 s_2 \dots s_i \dots s_n)$ to a labeled sequence $S'(s_1 s_2 \dots l_i \dots s_n)$ by replacing one token s_i in the input sequence with a designated label (l_i).

Supervised Comparative Mining Method-J and L treated comparative sentence identification as a classification problem and comparative relation extraction as an information extraction problem. They first manually created a set of 83 keywords such as beat, exceed, and outperform that are likely indicators of comparative sentences. These keywords were then used as pivots to create part-of-speech (POS) sequence data. A manually annotated corpus with class information, i.e., comparative or noncomparative, was used to create sequences and CSRs were mined. J and L's method is having following weakness. 1. The performance of J and L's method relies heavily on a set of comparative sentence indicative keywords. These keywords were manually created and they offered no guidelines to select keywords for inclusion. 2. To ensure the completeness of the keyword list is difficult. 3. To have high recall, a large annotated training corpus is necessary. This is an expensive process. Weakly Supervised comparator mining method for comparator mining was proposed by Shasha Li, Chin- Yew Lin, Young-In Song, and Zhoujun Li[1]. In this method they used sequential patterns to identify comparative questions and extract comparators simultaneously. Sequential pattern is defined as sequence $S(s_1s_2 \dots s_i \dots s_n)$ 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 ($\#end$). Indicative Extraction Pattern-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. In mining indicative extraction patterns bootstrapping algorithm is used. In this algorithm two steps are performed 1. Pattern Generation. 2. Pattern Evaluation. Comparator extraction-To extract comparator from comparative questions three strategies are used. Random strategy. Given a question, randomly select a pattern among patterns which can be applied to the question. Maximum length strategy. Given a question, select the longest one among patterns which can be applied to the question. Maximum reliability strategy. Given a question, select the most reliable one among patterns which can be applied to the question. The weakness of this method is that it is not possible to identify comparator alises e.g. If question such as Which is better antivirus NP or Net Protector? arise then this method will not be able to identify that the comparators in this question are alises of each other. In next section we introduced method to avoid these difficulties.

IMPROVED WEAKLY SUPERVISED METHOD.

The Weakly supervised method is improved to verify comparator alises. This method is pattern based using sequential patterns. To simultaneously identify comparative questions and extract comparator in them, this method learn sequential patterns.

A. Verifying Comparator Alises

When a question found, the verifier first verify whether comparators are alises of each other by checking patterns available for them if they are not then only it proceed to next step.

B. Indicative Extraction Patterns Mining Two key assumption on which Weakly Supervised IEP mining approach is based on are.

1. If a sequential pattern can be used to extract many reliable comparator pairs, then the sequential pattern is IEP.
2. If a comparator pair can be extracted by an IEP, the pair is reliable.

The Bootstrapping algorithm is based on these two assumptions. Pattern Generation Surface text pattern mining method used to to generate sequential patterns. Three types of sequential patterns are generated from questions.

1. Lexical patterns -Lexical patterns indicate sequential patterns consisting of only words and symbols ($\$C$, $\#start$, and $\#end$). Suffix tree algorithm [10] used to generate lexical pattern.
2. Generalized patterns.- A lexical pattern is much

more specific than required hence it is generalized by replacing one or more words/phrases with their POS tags. From a lexical pattern containing N words 2^{n-1} generalized patterns can be produced.

3 Specialized patterns. In some cases, a pattern can be too general that there can be many noncomparative questions matching the pattern. So these patterns are specialized.

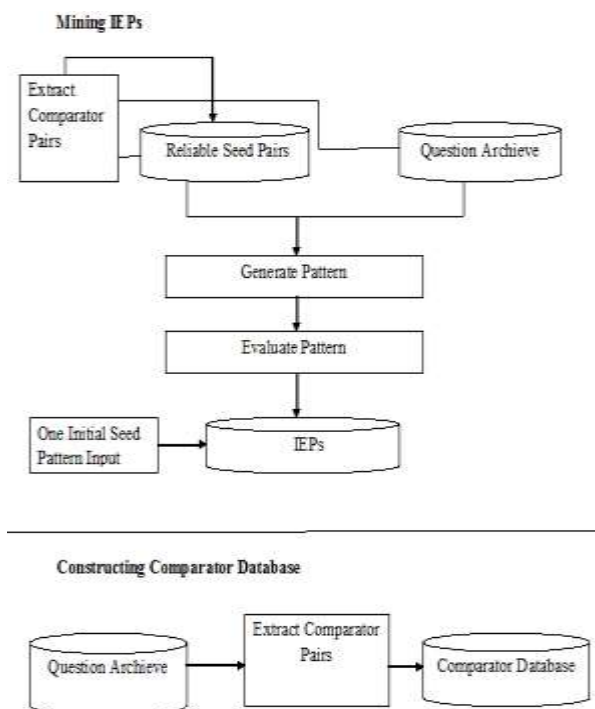
Note that generalized patterns are generated from lexical patterns and the specialized patterns are generated from the combined set of generalized patterns and lexical patterns. The final set of candidate patterns is a mixture of lexical patterns, generalized patterns and specialized patterns

IMPLEMENTATION DETAILS

SYSTEM FLOW

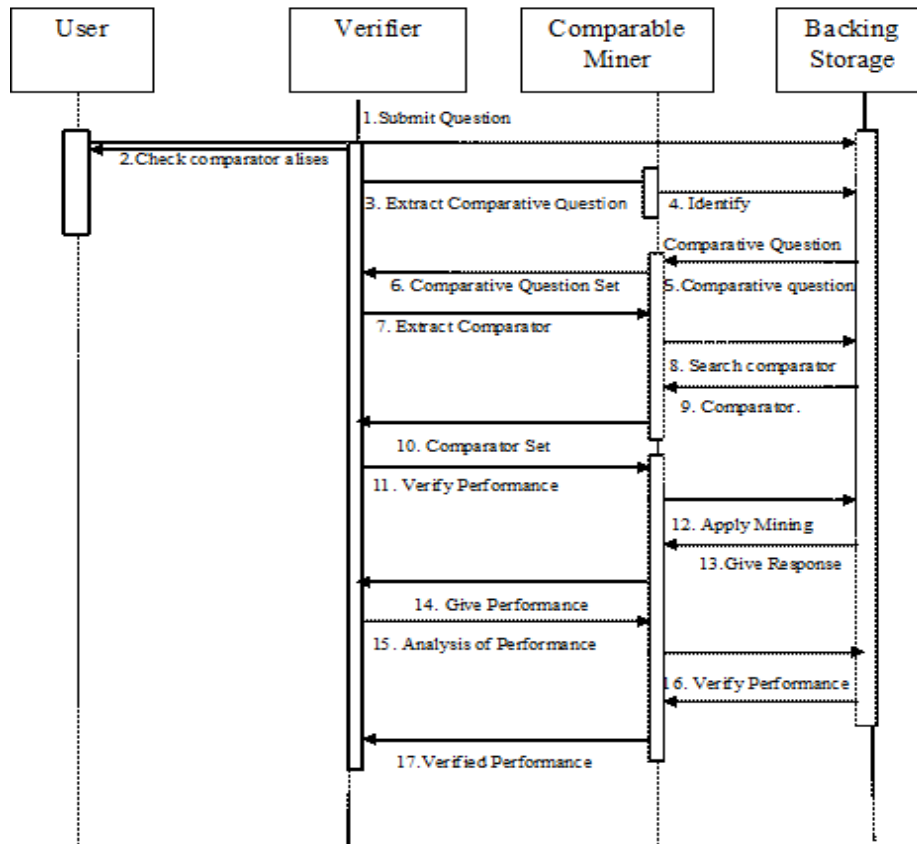
1. Start with single IEP.
2. From single IEP we extract a set of initial seed comparator pairs.
3. For each comparator pair all questions containing the pair are retrieved from question collection and regarded as comparative questions.

Figure 1. SYSTEM FLOW



4. From comparative questions and comparator pairs, all possible sequential patterns are generated and evaluated by measuring their reliability score.
5. When pattern evaluated as reliable then it is IEP and it is added into IEP repository.
6. New comparator pairs are extracted from question collection using latest IEPs.
7. The new comparators are added into reliable comparator repository and used as seeds for pattern learning in next iteration.
8. All question from which reliable comparators are extracted are removed from collection to allow finding new patterns efficiently in latter iteration.
9. The process iterates until no more patterns can be found from question collection.

Figure 3. System Model



COMPARATOR RANKING

The remaining issue is to rank possible comparators for a user’s input. The following ranking models are examined for this issue.

Comparability-Based Ranking Method

Intuitively, a comparator would be more interesting for an entity if it is compared with the entity more frequently. Based on this intuition, we define a simple ranking function $R_{freq}(c,e)$ 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 :

$$R_{freq}(c; e) = N(Qc;e)$$

where $Qc;e$ is a set of questions from which c and e can be extracted as a comparator pair. We will call this ranking function as Frequency-based Method.

EXPERIMENT SETUP AND RESULTS

All experiments were conducted on data collected from Yahoo! Answers question title field. The reason that we used only a title field is that they clearly express a main intention of an asker with a form of simple questions in general.

Evaluation Data for Comparator Extraction

We used data stored in file bigdata.txt. This file contains comparative as well as non comparative questions. The questions were then annotated as comparative and non comparative based on criteria for comparative question identification. Then the questions were classified into two files as comparative and non comparative. All experiments were conducted on these files.

Table 1 Examples of Comparators for Different Entities

	Nikon	Dell	Canon	Samsung
1.	Canon	Hp	Sony	LG
2.	Sony	Compaq	Hp	Nokia
3.	Panasonic	Lenovo	Samsung	Lenovo
4.	Hp	Asus	Epson	Asus
5.	Kodak	Acer	-----	Blackberry
6.	Casio	LG	-----	Motorolla

Table 1 is the list of frequently compared entities for a target item, such as Nikon, Dell, Canon, Samsung in our question archive. As shown in the table, our comparator mining method successfully discovers realistic comparators. For example, for Nikon, most results are frequently compared brands such as Canon, Sony, Panasonic, Hp, Kodak, Casio etc, while the ranking results for Dell usually contains similar brands such as Hp, Samsung, Epson etc. Some interesting comparators are shown for Canon (the company name). It is famous for different kinds of its products, for example, digital cameras and printers, so it can be compared to different kinds of companies. For example, it is compared to HP, or Panasonic, the printer manufacturers, and also compared to Nikon, Sony, or Kodak, the digital camera manufactures. Besides general entities such as a brand or company name, our method also found an interesting comparable entity for a specific item in the experiments.

Table 2 shows performance results. Precision is the positive predicate value indicating fraction of retrieved instances those are relevant is more than 80 percent. Recall also called sensitivity indicate fraction of relevant instances those are retrieved is also more than 80 percent. F-score is a measure that combines precision and recall. It is harmonic mean of precision and recall. F-score is also more than 80 percent.

We also analyzed the effect of pattern generalization and specialization. Table 3 shows the results. Despite of the simplicity of our methods, they significantly contribute to performance improvements. This result shows the importance of learning patterns flexibly to capture various comparative question expressions.

Table 2 Performance Results

	Identification Only (SET-A+SET B)	Extraction only (SET-B)	All (SET-B)
Recall :	0.839547547277	0.77513227	0.76302083
Precision:	0.809392265193	0.90993788	0.76701570
F-score :	0.824191279887	0.83714285	0.76501305

Table 3 Effect of pattern specialization and generalization on Performance

	Recall	Precision	F-score
Original:	0.699617956	0.425995929	0.549460853
Specialized	0.763219588	0.622609434	0.633993292
Generalized	0.763219588	0.735811150	0.749264799

Table 4 below shows the example entities and their alises

<i>Table 4 Example entities and their alises</i>	
Entity	Alises
LG	Life's Good
HTC	High Tech Computers
HCL	Hindustan Corporation Ltd.
NP	Net protector
MS	Microsoft
Skypee	Sky Peer

Table 4 shows comparators and their alises such as for the company name HCL alise is Hindustan Corporation Ltd. The alise for entity NP is Net protector etc.

Our weakly Supervised method using bootstrapping algorithm will first identify comparator alises from comparator database and if the comparable entities are not alises of each other then only it will proceed to next that is it will generate patterns for comparators otherwise it will neglect the alises.

CONCLUSION AND FUTURE SCOPE.

In this paper, we focus on a novel weakly supervised method to identify comparative questions and extract comparator pairs simultaneously. We improve bootstrapping algorithm to first verify whether comparators are alises of each other. If they are not then only it will further proceed. We rely on the key insight that a good comparative question identification pattern should extract good comparators, and a good comparator pair should occur in good comparative questions to bootstrap the extraction and identification process. The goal of this work is mining comparators from comparative questions and then furthermore, provide and rank comparable entities for a user's input entity appropriately. Results would be very useful in helping users' exploration of alternative choices by suggesting comparable entities based on other users' prior requests. This is the first attempt to specially address the problem on finding good comparators to support users' comparison activity. This is also the first to propose using comparative questions posted online that react what users truly care about as the medium from which we mine comparable entities. Once a question matches an IEP, it is classified as a comparative question and the token sequences corresponding to the comparator slots in the IEP are extracted as comparators. When a question can match multiple IEPs, the longest IEP is used. Therefore, instead of manually creating a list of indicative keywords, we create a set of IEPs. The evaluations shown confirm that our weakly supervised method can achieve high recall while retain high precision. Our comparator mining results can be used for a commerce search or product recommendation system. For example, automatic suggestion of comparable entities can assist users in their comparison activities before making their purchase decisions. Also, our results can provide useful information to companies which want to identify their competitors.

In the future, we would like to improve extraction pattern application and mine rare extraction patterns. How to separate ambiguous entities such "Paris versus London" as location and "Paris versus Nicole" as celebrity are all interesting research topics. We also plan to develop methods to summarize answers pooled by a given comparator pair.

ACKNOWLEDGMENT


I take this opportunity to thank all those involved directly or indirectly in this work. So it gives me great pleasure, on the completion of this paper, to acknowledge and appreciate all those who were there to help me. I express my sincere and profound thanks to all our teachers-HOD, PG Coordinator and Project Guide. I would like to say thanks to our college for the boost that it has provided. I cannot express in few words my gratitude towards guide for his 'studentlike' enthusiasm and his guidance from time to time. I heartily thank for all his help and valuable time. His invaluable advice has helped me bring this dissertation work at this stage.

I would like to thank all our internal guides for providing the resources and also like to thank for encouraging me to do it more effectively. I also acknowledge the research work done by all researchers in this field. And last but not least, all my friends, who have helped me directly or indirectly throughout the work. We are very much thankful to all authors; those are mentioned in the references and all the respected people who helped us for designing and development of our work.

REFERENCES

1. Shasha Li, Chin-Yew Lin, Member, Young-In Song, and Zhoujun Li, "Comparable Entity Mining from Comparative Questions" *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, July 2013
2. G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, Jan./Feb. 2003.
3. 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.
4. C. Cardie, "Empirical Methods in Information Extraction," *Artificial Intelligence Magazine*, vol. 18, pp. 65-79, 1997.
5. E. Riloff and R. Jones, "Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping," *Proc. 16th Nat'l Conf. Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conf. (AAAI '99/IAAI '99)*, pp. 474-479, 1999.
6. E. Riloff, "Automatically Generating Extraction Patterns from Untagged Text," *Proc. 13th Nat'l Conf. Artificial Intelligence*, pp. 1044-1049, 1996.
7. S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," *Machine Learning*, vol. 34, nos. 1-3, pp. 233-272, 1999.
8. 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)*, pp. 244-251, 2006.
9. N. Jindal and B. Liu, "Mining Comparative Sentences and Relations," *Proc. 21st Nat'l Conf. Artificial Intelligence (AAAI '06)*,
10. D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge Univ. Press, 1997.

AUTHOR BIBLIOGRAPHY

	<p>Tejaswini H. Patil Obtained the B.E, degree from TKIET, Warananagar, Maharashtra. At present pursuing the M.E. in Computer Engineering from Department of Computer Science and Engineering Alard College of Engineering and Management Pune. Maharashtra</p>
	<p>Sonali Patil Obtained M.E degree in computer Engineering and at present assistant professor at Alard College of Engineering and Management Pune. Maharashtra</p>