International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

A Survey on Comparable Entity Mining From Comparative Questions

Tejaswini H. Patil

ME Student, Department of Computer Science and Engineering, Alard College of Engineering and Management, Pune Maharashtra, India

Abstract: ***In day to day life it is necessary to compare number of options. It is one of the necessary processes in making decision while purchasing online. Though, it is not simple task to compare different alternatives for each and every comparison. In this paper, we performed a survey on a novel way to automatically mine comparable entities from comparative questions that users posted online. To perform this task, used a weakly supervised method to identify comparative question and mine comparable entities. After mining comparative entities, next step is to rank possible comparators for user's input and or that purpose comparability based ranking and graph based ranking methods employed.***

Keywords: Information extraction, bootstrapping, sequential pattern mining, comparable entity mining, Indicative Extraction Pattern, lexical pattern, generalized pattern, and specialized pattern.

1. Introduction

This work is survey on Comparable Entity Mining from Comparative Questions [1]. The essential step while making decision is comparing alternative options because without comparing number of options available, it is not possible to make best decision. For example, if someone is interested in certain products such as laptops, he or she would wants to know what the alternatives are and compare different laptop types before making a purchase. This type of comparison activity is very common in our daily life but requires high knowledge skills. In the World Wide Web era, a comparison activity typically involves: search for relevant web pages containing information about the targeted products, find competing products, read reviews, and identify best option. We focus on finding a set of comparable entities given user's input entity. For example, given an entity, Samsung Galaxy (a smart phone), we want to find comparable entities such as Nokia N82, iphone and so on. To mine comparators from comparative questions, we first have to detect whether a question is comparative or not.

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.

Comparative question. A question that intends to compare two or more entities and it has to mention these entities explicitly in the question.

Comparator. An entity which is a target of comparison in a comparative question.

Q 1. Which camera is better Canon or Nikon?
Q 2. Whether Nikon camera is best camera
First question compare two entities so the question is comparative question. But the second question is not comparative question because it does not compare two entities.

The rest of this paper is organized as follows: the next Section discusses previous works. Section 3 presents weakly supervised method for comparator mining. Section 4 Discussed how to rank comparable entities for a user's input Entity and build a comparator database. Section 5 reports the Evaluations of our techniques, and we conclude the paper And discuss future work in Section 6.

2. Related Work

The work is related to rrecommender system [2].The algorithms are best known for their use on e-commerce Web sites, where they use input about a customer's interests to generate a list of recommended items. Many applications use only the items that customers purchase to predict which item he or she will buy after this.

Comparator mining is related to the research on entity and relation extraction in information extraction specifically, the most relevant work is mining comparative sentences and relations. [1][3][4][5][6] Their methods applied Class Sequential Rules (CSRs) [7] and Label Sequential Rules (LSR) [7] to identify comparative sentences and extract comparative relations respectively in the news and review domains. The same techniques can be applied to comparative question identification and comparator mining from questions. But the method can typically achieve high precision but suffer from low recall. So in this work used Weakly Supervised Method for Comparator Mining which achieves high precision and high recall.

3. A Weakly Supervised Method for Comparator Mining

This system presents a novel weakly supervised method to identify comparative questions and extract comparator pairs simultaneously. The system 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. To ensure high precision and high recall used Weakly Supervised Bootstrapping method for comparative question identification and comparable entity extraction.

3.1 Mining Indicative Extraction Patterns

In our approach, a sequential pattern is defined as a sequence S (s1, s2 . . . si . . . sn) where si 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 (IEP) is a sequential pattern that can be used to identify comparative questions and extract comparators in them with high reliability. 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. 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.

3.1.1 Pattern Generation (Comparable Entity)

The weakly supervised IEP mining approach is based on two key assumptions.

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

Based on these two assumptions, we design our bootstrapping algorithm as shown in Figure 1.

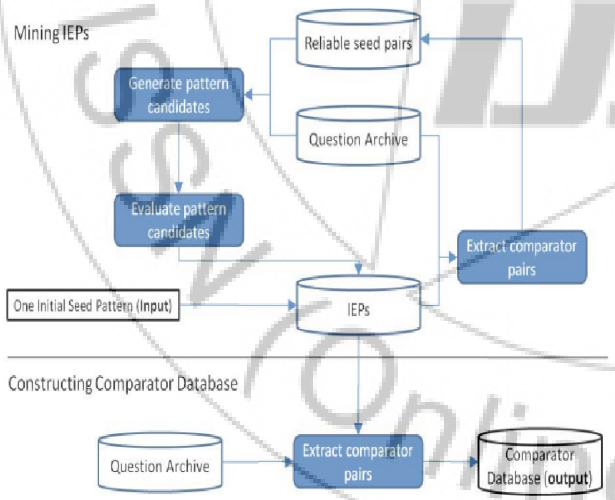

Figure 1: Overview of Bootstrapping Algorithm.

The bootstrapping process starts with a single IEP. From it, we extract a set of initial seed comparator pairs. For each comparator pair, all questions containing the pair are retrieved from a question collection and regarded as comparative questions. From the comparative questions and comparator pairs, all possible sequential patterns are generated and evaluated by measuring their reliability score defined in the Pattern Evaluation section. Patterns evaluated as reliable ones are IEPs and are added into an IEP repository. Then, new comparator pairs are extracted from the question collection using the latest IEPs. The new comparators are added to a reliable comparator repository and used as new seeds for pattern learning in the next iteration. All questions from which reliable comparators are extracted are removed from the collection to allow finding new patterns efficiently in later iterations. The process iterates until no more new patterns can be found from the question collection.

The pseudo code of the algorithm is shown in Figure 2.

Algorithm 1 Weakly-Supervised Model

Input: CP, G
Initialize solution: $Q \leftarrow \{\}, P \leftarrow \{\}, P_{new} \leftarrow \{\}, CP_{new} \leftarrow CP$

```
Repeat
P ← P + P_new
Q_new ← ComparativeQuestionIdentify( CP_new )
Q ← Q + Q_new
  for q_i ∈ G do
    if IsMatchExistingPatterns(P, q_i) then
      Q ← Q − q_i
    end if
  end for
  P_new ← MineGoodPatterns(Q)
  CP_new ← {}
  for q_i ∈ G do
    cp ← ExtractComparableComparators(P, qi)
    if cp ≠ NULL and cp ∉ CP then
      CP_new ← CP_new + {cp}
    end if
  end for
until P_new = {}
return P
```

Figure 2: Pseudo code of the Bootstrapping algorithm.

There are two key steps in our method:
1) Pattern generation and
2) Pattern evaluation.
In the following sections, we will explain them in details.

To generate sequential patterns, here used the surface text pattern mining method.[8] For any given comparative question and its comparator pairs, comparators in the question are replaced with symbol $Cs. Two symbols, #start and #end, are attached to the beginning and the end of each sentence in the question. To reduce diversity of sequence data and mine potential patterns, phrase chuncking is applied. Then, the following three kinds of sequential patterns are generated from sequences of questions:

1. Lexical patterns- Lexical patterns indicate sequential patterns consisting of only words and symbols ($C, #start, and #end). They are generated by suffix tree algorithm with two constraints: a pattern should contain more than one $C, and its frequency in collection should be more than an empirically determined number.
2. Generalized patterns-A lexical pattern can be too specific. Thus, we generalize lexical patterns by replacing one or more words/phrases with their POS tags. 2^n- 1 generalized patterns can be produced from a lexical pattern containing N words excluding $Cs.
3. Specialized patterns- In some cases, a pattern can be too general. For example, although a question “ipod or

zune?" is comparative, the pattern "<$C or $C>" is too general, and there can be many non comparative questions matching the pattern, for instance, "true or false?". For this reason, we perform pattern specialization by adding POS tags to all comparator slots. For example, from the lexical pattern "<$C or $C>" and the question "ipod or zune?", "<$C=NN or $C=NN?>" will be produced as a specialized pattern.

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.

3.1.2 Pattern Evaluation(Comparable Question)

Incomplete knowledge about reliable comparator pairs. e.g. Very few reliable pairs are generally discovered in early stage of bootstrapping. So to discover more and more reliable pair's pattern evolution operation is performed.

4. Comparator Ranking

The next step remain is to rank possible comparators for a user's input. There are two methods for ranking comparators.

4.1 Comparability-Based Ranking Method

A comparator would be more interesting for an entity if it is compared with the entity more frequently. Based on this intuition, 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. The ranking method is also called Frequency-based Method. The another ranking function is R_{rel} by combining reliability scores estimated in comparator mining phase

$$R_{rel}(c, e) = \sum_{q \in c,e} R(p_{q,c,e}),$$

Where $p_{q, c, e}$ means the pattern that is selected to extract comparator pair of c and e from question q in comparator mining phase. This ranking function will be denoted as Reliability-based Method.

4.2 Graph Based Ranking Method

Though frequency is efficient for comparator ranking, the frequency-based method can suffer when an input occurs rarely in question collection; for example, suppose the case that all possible comparators to the input are compared only once in questions. In this case, the Frequency-based method may fail to produce a meaningful ranking result. Then, represent ability should also be considered. We regard a comparator representative if it is frequently used as a baseline in the area the user is interested in. For example, when one wants to buy a smart phone and he/she is considering "Nokia N82," "Nokia N95" is the first one he/she wants to compare. That's because "Nokia N95" is a well-known smart phone and it's usually used as a baseline to help users know the performance of other smart phones better.

One possible solution to consider represent ability can be to use graph-based method such as Page Rank. If a comparator is compared to many other important comparators which can be also compared to the input entity, it would be considered as a valuable comparator in ranking. Based on this idea, we examine Page Rank algorithm to rank comparators for a given input entity which combine frequency and represent ability. We define a graph G (V, E). In the graph, V is the set of nodes v, which consists of comparable comparators of the input. The edge e_{ij} between v_i and v_j means that the two. Comparators are compared in our comparator pair repository.

5. Experimental Results

Table 1: Examples of comparators for different entities

iPod	*Canon*	*Lenova*
Cell	Nikon	Acer
iPhone	Sony	Dell
Zen	Panasonic	iBall
iPod nano	Kodak	LG
MP3	Casio	Toshiba
Zune	Hp	Sony

The table above illustrate comparator for different entities. Such as for iPod the comparable entities are cell phone, iPhone, Zen,etc. For Camera type Canon comparable entities are Nikon, Sony, panasonic etc and for laptop type Lenova the Comparator entities are Acer, Dell, iBall, LG etc.

Conclusion

This paper presents a novel weakly supervised method to identify comparative questions and extract comparator pairs simultaneously. It 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 experimental results show that bootstrapping method is effective in both comparative question identification and comparator extraction. It significantly improves recall in both tasks while maintains high precision. The 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, results can provide useful information to companies which want to identify their competitors.

The future work is to improve extraction pattern application and mine rare extraction patterns. How to identify comparator aliases such as "LV" and "Louis Vuitton" and 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.

References

[1] S.Li, C.-Y. Lin, Y.-I. Song, and Z. Li,"Comparable Entity Mining from Comparative Questions," Proc. 48th Ann. Meeting of the Assoc. for Computational Linguistics (ACL '10), 2010.
[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] C. Cardie, "Empirical Methods in Information Extraction,"Artificial Intelligence Magazine, vol. 18, pp. 65-79, 1997.
[4] 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.
[5] E. Riloff, "Automatically Generating Extraction Patterns from Untagged Text," Proc. 13th Nat'l Conf. Artificial Intelligence,pp. 1044-1049, 1996.
[6] S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," Machine Learning, vol. 34, nos. 1-3, pp. 233-272, 1999.
[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), pp. 244-251, 2006.
[8] 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.

Author Profile

Tejaswini H. Patil obtained the B.E, degree from TKIET, Warananagar, Maharashtra, India. At present she is pursuing M.E. in Computer Engineering from Department of Computer Science and Engineering Alard College of Engineering and Management Pune. Maharashtra, India.