A Method for Comparable Mining from Comparative Questions

VENKATA REDDY¹, K.MAHESH BABU²

¹PG Scholar, Dept of CSE, Kottam Karunakar Reddy Institute of Technology, Kurnool, AP, India, E-mail: venkat.reddy494@gmail.com.
²Associate Professor, Dept of CSE, Kottam Karunakar Reddy Institute of Technology, Kurnool, AP, India.

Abstract: The process of comparing one thing with another thing is a typical part in human decision making process. Whenever we plan to buy a product, we usually check all the products in the market then compare the products and purchase the best one. But it is a typical task to select a single product by comparing multiple products and address a best product. In order to reduce this difficulty, we address a novel way to automatically mine comparable entities from comparative questions that users request. To ensure high precision and high recall, we develop a bootstrapping method for comparative question identification and comparable key word entity extraction. This method significantly determines the best comparative question and comparable entity to select a best product.

Keywords: Indicative Extraction Patterns, CSRs (Class Sequential Rule) And LSRs (Label Sequential Rule).

I. INTRODUCTION

The process of comparing one thing with another thing is a typical part in human decision making process. In decision-making process, comparing alternative options is one of the necessary steps that we carry out on a daily basis. However, this requires high knowledge expertise. For e.g., during online shopping of a laptop one must have detailed knowledge of its specifications like Processor, Memory, Storage, graphics, Display, etc. In such case, it becomes difficult for a person 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 Consumer Reports, PC Magazine and online media such as CNet.com make efforts in providing editorial comparison content and surveys to satisfy this need. A comparison activity, in World Wide Web era, normally involves: search for applicable web pages enclosing information regarding the targeted products, discovering challenging products, and recognize pros and cons. In this paper, our focus is on finding a set of comparable entities provided a user’s input entity. For example, provided an entity, Nokia N95 (cell phone), we want to find comparable entities such as Nokia N82, iPhone, blackberry and so on. To extract comparable entities from relative matter, we should first know whether a question is relative or not.

Bootstrapping method to identify comparative questions and extract comparators simultaneously. The comparative Question and the comparator can be defined as:

Comparative Question: A Question that makes to compare two or more keywords and it has to mention these keywords compulsorily in the question.

Comparator: A keyword which is a target of comparison in a comparative question.

According to the definitions, Q1 & Q2 below are not comparative questions whereas Q3 is. “Secunderabad” and “Hyderabad” are comparators.

Q1. “Which one is better?”
Q2. “Is Hyderabad the best city?”
Q3. “Which city is better Secunderabad or Hyderabad?”

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

II. INFORMATION EXTRACTION

In terms of discovering related items for an entity, our work is similar to the research on recommender systems, which recommend items to a user. Recommender systems mainly rely on similarities between items and/or their statistical correlations in user log data [12]. For example, Amazon recommends products to its customers based on their own purchase histories, similar customer’s purchase histories, and similarity between products. However, recommending an item is not equivalent to finding a comparable item. In the case of Amazon, the purpose of recommendation is to entice their customers to add more items to their shopping carts by suggesting similar or related items. Bootstrapping methods have been shown to be very effective in previous information extraction research [3,4,5]. Our work is similar to them in terms of methodology using bootstrapping technique to extract entities with a specific relation. However, our task is different from theirs in that it requires not only extracting entities (comparator extraction) but also ensuring that the entities are extracted from comparative questions.
III. WEAKLY SUPERVISED METHOD FOR COMPARATOR MINING

Our weakly supervised method is a pattern-based approach similar to J&L_1 method, but it is different in many aspects: Instead of using separate CSRs(Class sequential rule) and LSRs(Label sequential rule), our method aims to learn sequential patterns which can be used to identify comparative question and extract comparators simultaneously. In our approach, a sequential pattern is defined as a sequence $S(s_1, s_2, ..., s_n)$ 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. Once a question matches an IEP, it is classified as a comparative question and the token sequence $s$ corresponding to the comparator slots in the IEP are extracted as comparators. When a question can match multiple IEPs, the longest IEP is used.[6] Therefore, instead of manually creating a list of indicative keywords, we create a set of IEPs. We will show how to acquire IEPs automatically using a bootstrapping procedure with minimum supervision by taking advantage of a large unlabeled question collection in the following sub sections.

![Fig.1. Architecture Diagram.](image)

A. Mining Indicative Extraction Patterns

The weakly supervised IEP mining approach is based on two key assumptions: If a sequential pattern can be used to extract many reliable comparator pairs, it is very likely to be an IEP. 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 Fig.1. 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. Patterns evaluated as reliable on es 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. The overview of bootstrapping algorithm is shown below, where the databases store seed pairs and question archive and from them relevant data is extracted.

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. There exist two steps in our approach:

- **Pattern Generation**
- **Patter Evaluation**

**Pattern Generation:** To generate sequential patterns, we adapt the surface text pattern mining method introduced in [11]. For any given comparative question and its comparator pairs, comparators in the question are replaced with symbol $C$s. Two symbols, #start and #end, are attached to the beginning and the end of a sentence in the question. Then, the following three kinds of sequential patterns are generated from sequences of questions:

- **Lexical patterns:** Lexical patterns indicate sequential patterns consisting of only words and symbols ($C$, #start, and #end). They are generated by suffix tree algorithm [3] with two constraints: A pattern should contain more than one $C$, and its frequency in collection should be more than an empirically determined number.

- **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.

**IV. IMPLEMENTATION AND RESULTS**

At first we just create a comparative question and comparator keyword as shown in Fig.2:

---

VENKATA REDDY, K.MAHESH BABU

**International Journal of Scientific Engineering and Technology Research**

*Volume.03, IssueNo.30, October-2014, Pages: 5942-5945*
A Method for Comparable Mining from Comparative Questions

The comparable entity in lexical pattern is:

Fig.2. Create a comparative question and comparator keyword.

Then the Bootstrapping algorithm gives three types of comparative pattern results as shown in Figs.3, 4 and 5. The Lexical pattern comparable entity result for the given example is:

Fig.3. The Lexical pattern comparable entity.

Fig.4. Comparable entity in lexical pattern.

The comparable entity for specialized pattern is:

Fig.5. Comparable entity in lexical pattern.
At last by applying our bootstrapping method to the entire source data (60M questions), 328,364 unique comparator pairs were extracted from 679,909 automatically identified comparative questions. The result is:

<table>
<thead>
<tr>
<th>channel</th>
<th>Gap</th>
<th>iPod</th>
<th>kobe</th>
<th>canon</th>
</tr>
</thead>
<tbody>
<tr>
<td>handbag</td>
<td>Gap coupons</td>
<td>ipod nano</td>
<td>kobe Bryant status</td>
<td>canon t3i</td>
</tr>
<tr>
<td>sunglass</td>
<td>Gap outlet</td>
<td>ipod touch</td>
<td>lakers kobe</td>
<td>canon printers</td>
</tr>
<tr>
<td>earrings</td>
<td>Gap card</td>
<td>ipod best buy</td>
<td>kobe eason</td>
<td>canon printer drivers</td>
</tr>
<tr>
<td>watches</td>
<td>Gap card hole</td>
<td>iTunes</td>
<td>kobe dallas mavericks</td>
<td>canon downloads</td>
</tr>
<tr>
<td>watches 2i</td>
<td>Gap careers</td>
<td>Apple</td>
<td>kobe NEA</td>
<td>canon copiers</td>
</tr>
<tr>
<td>shoes</td>
<td>Gap casting call</td>
<td>ipod shuffle</td>
<td>kobe 2009</td>
<td>canon scanner</td>
</tr>
<tr>
<td>Dor</td>
<td>Old navy</td>
<td>ipod support</td>
<td>kobe san Antonio</td>
<td>canon lenses</td>
</tr>
<tr>
<td>shoes t2i</td>
<td>Banana republic</td>
<td>ipod classic</td>
<td>kobe Bryant 24</td>
<td>Nikon</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we present a new supervised method for identifying comparative questions and extraction of comparator pairs at the same time. We rely on the key insight 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. 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.

VI. REFERENCES


