International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Humming Bird (01st March 2014)

RESEARCH ARTICLE OPEN ACCESS

Comparable Entity Mining From Comparative Question Archive

Veena.G.R.Kumar1, Jameemah Simon 2
1Veena G.R. Kumar. Author is currently pursuing M.E (CSE) in VINS Christian college of Engineering,Nagercoil. e-mail:veena.grkumar@gmail.com.
2Jemeema , Assistant Lecturer in VINS Christian college of Engineering.

Abstract:
Comparing one thing with another is a typical part of human decision making process, especially during an online purchase scheme. Without comparing it is not fair to purchase a product, since it won't give an ideal performance. To get rid of this difficulty, my paper presents an ideal way for automatically mine comparable entities from comparative questions that users posted online. It gives an opportunity to improve the search experience by automatically offering comparisons to user. A weekly supervised bootstrapping algorithm is employed here for comparative question identification and comparable entity extraction by collecting a large online question archive. It also provides users to add new attributes of their interest to the annotation form, so that the next search retrieves the provided new attribute information. This technique would outperform the existing system of online shopping.
Keywords— Information extraction, Natural language processing, part of speech, Robust automated production of IER

I. INTRODUCTION

Comparing alternative options is one essential step in decision making that we carry out every day. For example, if someone is interested in certain products such as digital cameras, he or she would want to know what the alternatives are and compare different cameras before making a purchase. This type of comparison activity is very common in our daily life but requires high knowledge skill. 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 pros and cons.

In this paper, we focus on finding a set of comparable entities given user's input entity. For example, given an entity, Nokia N95 (a cell 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. According to our definition, a comparative question has to be a question with intent to 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 intent. However, we observe that a question is very likely to be a comparative question if it contains at least two entities. We leverage this insight and develop a weakly supervised

bootstrapping method to identify comparative questions and extract comparators simultaneously.

The comparative questions and comparators can be thus defined as:

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.

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 [8] . 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 [9,11,12]. Our work

is similar to them in terms of methodology using bootstrapping techniqu e to extract entities with a specific relation. However, our task is different from theirs in that it requires not o nly extracting entities (comparator extraction) but also ensuring that the entities are extracted from comparative questions (comparative question identification), which is generally not required in IE task.

III. WEAKLY SUPERVISED METHO D FOR COMPARATOR MINING

Our weakly supervised method is a pattern-based approach similar to J&L_s[6] method, but it is different in many aspects: Instead of using separat e 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(s1s2 … 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). A sequential pattern is called an indicative ex traction pattern (IEP) if it can be used to identify comparativ e questions and extract comparators in them with high reli ability. Once a question matches an IEP, it is classified as a comparative question and the token sequence s co rresponding 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 crea te a set of IEPs. We will show how to acquire IEPs automatically using a bootstrapping procedure with minimum s upervision by taking advantage of a large unlabeled questi on collection in the following sub sections.

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 assu mptions, we design our bootstrapping algorithm as shown in Figure 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 meas uring 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.

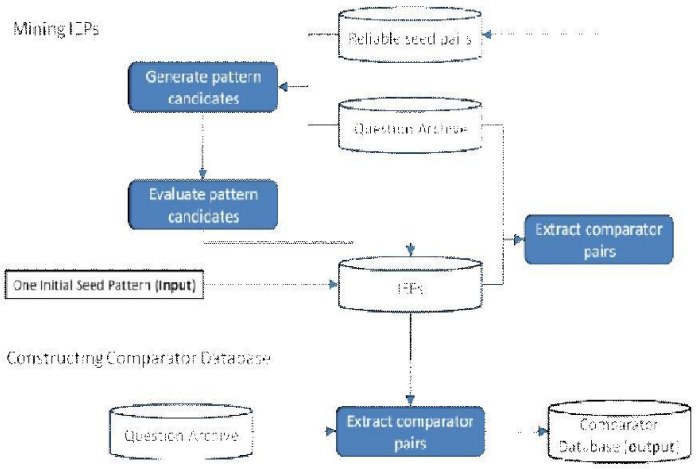

Figure 1: Overview of the boo tstrapping algorithm

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 are two key steps in o ur method:

(1) Pattern generation

(2) Pattern evaluation

(1) 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 $Cs. 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 .

Generalized patterns: A lexical pattern can be too specific.
words with their POS tags. 2 − 1 generalized patterns can be produced from a lexical pattern containing N words

Thus, we generalize lexical patterns by replacing one or more excluding $Cs.

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.

(2) Pattern Evaluation

According to our first assumption, a reliability score Rk(pi) for a candidate pattern

follows: at iteration k can be defined as

Rk pi = NQ(pi->cp) cp j € CP k-1

NQ (pi->*) ∀ (1)

where pi can extract known reliable comparator pairs cpj.CPK-1 indicates the reliable comparator pair repository accumulated until the (k-1)h iteration. (x) means the number of questions satisfying a condition x. The condition pi → cpj

denotes that cpj can be extracted from a question by applying pattern pi while the condition pi→_ denotes any question containing pattern.

However, Equation (1) can suffer from incomplete knowledge about reliable comparator pairs. For example, very few reliable pairs are generally discovered in early stage of bootstrapping. In this case, the value of Equation (1) might be underestimated which could affect the effectiveness of equation (1) on distinguishing IEPs from non-reliable patterns. We mitigate this problem by a look ahead procedure. Let us denote the set of candidate patterns at the iteration k by Pk.I define the support S for comparator pair c pi which can be extracted by P k and does not exist in the current reliable set:

S c pi = N(pk->c pi) (2)

where p k cp->i means that one of the patterns in p k can extract c pi in certain questions. Intuitively, if c pi can be extracted by many candidate patterns in , it is likely to be extracted as a reliable one in the next iteration. Based on this intuition, a pair c pi whose support S is more than a threshold α is regarded as a likely-reliable pair. Using likely-reliable pairs, look ahead reliability score R pi is defined:

R k pi= NQ(pi :pi) ∀ ∈ Rel k NQ(pi→_) (3)

where c p rel k indicates a set of likely-reliable pairs based on P K .By interpolating Equation (1) and (3), the final reliability score R (pi) final k for a pattern is defined as follows:

(pi)final k = λ. Rk pi +(1-λ). R (pi) (4)

Using Equation (4), I evaluate all candidate patterns and select patterns whose score is more than threshold γ as IEPs. All necessary parameter values are empirically determined and are diagnosed based on values.

IV. EXPERIMENTAL RESULT

A.Examples of Comparator Extraction

By applying our bootstrapping method to the entire source data (60M questions), 328,364 unique comparator p air s we r e extracted fr o m 679,909 automatically identified comparative questions.

	Chane	Gap	iPod	Kobe	Canon
1	Dior	Old Navy	Zune	Lebron	Nikon
2	Louis	American	mp3	Jordan	Sony
3	Coach	Banana	PSP	MJ	Kodak
4	Gucci	Guess by	cell	Shaq	Panasonic
5	Prada	ACP	iPhone	Wade	Casio
6	Lancom	Old Navy	Creative	T-mac	Olympus
7	Versace	Hollister	Zen	Lebron	Hp
8	LV	Aeropostal	iPod	Nash	Lexmark
9	Mac	American	iPod	KG	Pentax
1	Dooney	Guess	iRiver	Bonds	Xerox

Table 6: Examples of comparators for different entities

Table 6 lists top 10 frequently compared entities for a target item, such as Chanel, Gap, in our question archive. As shown in the table, our comparator mining method successfully discovers realistic comparators. For example, for Chanel, most results are high end fashion brands such as Dior or Louis Vuitton, while the ranking results for Gap usually contains similar apparel brands for young people, such as Old Navy or Banana Republic. For the basketball player Kobe_, most of the top ranked comparators are also famous basketball players. 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 Lexmark, or Xerox, 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. For example, our method recommends „Nikon d40i_ „ Canon rebel xti_, „Canon rebel xt_, „Nikon d3000_, „Pentax k100d_, Canon eos 1000d_ as kon 40d.

Chanel	Gap	iPod	Kobe	Canon
Chanel	Gap	***iPod***	Kobe	Canon t2i
Chanel	Gap outlet	***iPod***	Lakers	Canon
Chanel	Gap card	iPod best	Kobe espn	Canon
Chanel	Gap	iTunes	Kobe Dallas	Canon
Chanel	Gap	Apple	Kobe NBA	Canon
Chanel	Gap	iPod	Kobe 2009	Canon
Chanel	***Old navy***	iPod	Kobeesan	Canon
Dior	***Banana***	iPod	Kobe	***Nikon***

Table 7: Related queries returned by Google related searches for the same target entities in table6

Table 7 can show the difference between our comparator mining and query/item recommendation. As shown in the table, Google related searches generally suggest a mixed set of two kinds of related queries for a target entity: (1) queries specified with subtopics for an original query (e.g., Chanel handbag for Chanel) and (2) its comparable entities (e.g., Dior for Chanel). It confirms one of our claims that comparator mining and query/item recommendation are related but not the same.

V. CONCLUSION

In this paper, I present a novel weakly supervised method to identify comparative questions and extract comparator pairs simultaneously. I 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. By leveraging large amount of unlabeled data and the bootstrapping process with slight supervision to determine four parameters, it is found that 328,364 unique comparator pairs and 6,869 extraction patterns without the need of creating a set of comparative question indicator keywords

REFERENCES

[1] Mary Elain Califf and Raymond J. Mooney, Relational learning of pattern match rules for information extraction. In Proceedings of AAAI'99 /IAAI'99
[2] Claire Cardie. 1997. Empirical methods in information extraction. AI magazine, 18:65–79.
[3] Dan Gusfield. 1997. Algorithms on strings, trees, and sequences: computer science and computational biology. Cambridge University Press, New York, NY, USA

[4] Taher H. Haveliwala. 2002. Topic-sensitive pagerank.In Proceedings of WWW ’02, pages 517– 526.

[5] Glen Jeh and Jennifer Widom. 2003. Scalingpersonalized web search. In Proceedings of WWW ’03, pages 271–279.

[6] Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In Proceedings of SIGIR ’06, pages 244–251.

[7] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy.2008. Semantic class learning from the web with hyponym pattern linkage graphs. In Proceedings of ACL-08: HLT, pages 1048–1056.

[8] Greg Linden, Brent Smith and Jeremy York. 2003.Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, pages 76-80.

[9] Raymond J. Mooney and Razvan Bunescu. 2005.Mining knowledge from text using information extraction. ACM SIGKDD Exploration Newsletter.

[10] Dragomir Radev, Weiguo Fan, Hong Qi, and Harris Wu and Amardeep Grewal. 2002. Probabilistic question answering on the web. Journal of the American Society for Information Science and Technology, pages 408–419.

[11] Deepak Ravichandran and Eduard Hovy. 2002.Learning surface text patterns for a question answering system.

[12] Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi- level bootstrapping. In Proceedings of AAAI ’99/IAAI ’99, pages 474–479.

[13] Ellen Riloff. 1996. Automatically generating extraction patterns from untagged text. In Proceedings of the 13th National Conference on Artificial Intelligence, pages 1044–1049.

[14] Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. Ma- chine Learning, 34(1-3):233–272.

[15] Veena G R Kumar received the B.E degree from Ponjesly college of Engineering,Nagercoil in 2008 and currently doing M.E in VINS christian college of Engineering,Nagercoil.Her area of interest is in data mining and networking.