iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** I **Month of publication:** January 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Data Mining Information System

Divya C D[1], Bharath T S[2]
[1]Assistant Professor ,Dept of Computer Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, India,Affiliated to VTU Belgaum
[2]P G Student at SS College of Engineering, Udaipur,India

Abstract— Decision making between couple of things gone turn out critical for human thinking ability. Though, it is not always simple to recognize what to evaluate and what are the alternatives. To tackle this complexity, we explored a novel way to robotically mine equivalent entities from comparative questions that Users posted online. To make sure high accuracy, we created a weakly-Supervised bootstrapping method for Comparable question detection and comparable entity mining by leveraging a huge online questions archive. The investigational consequences explain our method gets eighty plus percentage of both F1-measures in comparative question detection and in comparable entity extraction. Both considerably do better than an previous state-of-the-art method.

Keywords — Bootstrapping; comparable entity mining; Information extraction; Class sequential rules(CSR); Label Sequential rules(LSR); Indicative extraction pattern(IEP).

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 mobiles, he or she would want to know what the alternatives are and compare different mobiles before making a purchase. This type of comparison activity is very common in our daily life but requires high knowledge skill. Magazines such as Consumer Reports and PC Magazine and online media such as CNet.com strive in providing editorial comparison content and surveys to satisfy this need. 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 a users input entity. For example, given an entity, Nokia N95 (a cellphone), we want to find comparable entities such as Nokia N82, iPhone and so on. In general, it is difficult to decide if two entities are comparable or not since people do compare apples and oranges for various reasons. For example, “Ford” and “BMW” might be comparable as “car manufacturers”, but we rarely see people comparing “Ford Focus” (car model) and “BMW 328i”. Things also get more complicated when an entity has several functionalities. For example, one might compare “iPhone” and “PSP” as “portable game player” while compare “iPhone” and “Nokia N95” as “mobile phone”. Fortunately, plenty of comparative questions are posted online, which provide evidences for what people want to compare, e.g. “Which to buy, iPod or iPhone?”. We call “iPod” and “iPhone” in this example as comparators. In this paper, we define comparative questions and comparators 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. According to these definitions, Q1 and Q2 below are not comparative questions while Q3 is “iPod Touch” and “Zune HD” are comparators.

Q1: “Which one is better?”

Q2: “Is Lumix GH-1 the best camera?”

Q3: “What’s the difference between iPod Touch and Zune

The goal of this work is mining comparators from comparative questions. The results would be very useful in helping users ‟exploration of alternative choices by suggesting comparable entities based on other users” prior requests. 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 Apriori-TID algorithm to extract comparable products.

A. Comparator Extraction

It is a manual process of comparing the products and based on the user's mindset which may not be accurate, which leads to confusions, less efficient and lack of customer satisfaction. The similar techniques can be functional to comparative question detection and comparator extractions from questions. This method normally can get high accuracy but faces low recall.

B. Comparable Entity Mining

We explore a novel weakly authenticated method to recognize comparative questions and extract comparator pairs simultaneously. we created a weakly-supervised bootstrapping method for comparative question detection and comparable entity mining by leveraging a huge online questions archive. By leveraging huge amount of bootstrapping process and the unlabeled information with slight supervision to conclude four parameters. Pros: To make certain max precision and high recall.

C. Evaluation Of Patterns (Comparable Questions)

1) Lexical Patterns: Lexical patterns point out sequential patterns containing of only symbols and words ($C, #start, and #end). They are developed by algorithm named as suffix tree with couple of conditions: A pattern should hold more than one $C, and its regularity in set should be more than an empirically resolute number.

2) Generalized Patterns: A lexical pattern nature will be too exact or specific. Therefore, we simplify lexical patterns by restoring one or more words respective of their POS tags. $2n - 1$ generalized patterns can be formed from a lexical pattern consisting N words not including $Cs.

3) Specialized Patterns: In few situations, a pattern can in the format of more general. Let us take a case, although a query "zune or ipod?" is proportional, the pattern "<$C or $C>" is more general, and there can be a lot of non-comparative queries matching the pattern, for example, "false or true?" For this cause, we do pattern specialty by addition POS tags to all comparator slots. From the lexical pattern let us take a simple case or situation, "<$C or $C>"and the query "zune or ipod?","<$C/NNor $C/NN?>" will be created as a specialized pattern. Pattern Evaluation (comparable questions): Incomplete awareness about consistent comparator pairs. Let us take a simple e.g., very little reliable pairs are normally exposed in the early hours stage of bootstrapping. In this situation, the importance of might be miscalculated which could impact the efficiency from non-reliable patterns of on individual IEPs. We moderate this crisis by a look ahead process. Let us signify the deposit of applicant patterns at the k iteration. We describe the maintain C for comparator pair S which can be take out by pk and won't exist in the current dependable collections.

II. RELATED WORKS

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 customers" 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. While in the case of comparison, we would like to help users explore alternatives, i.e. helping them make a decision among comparable items. For example, it is reasonable to recommend "iPod speaker" or "iPod batteries" if a user is interested in "iPod", but we would not compare them with "iPod". However, items that are comparable with "iPod" such as "iPhone" or "**PSP**" which were found in comparative questions posted by users are difficult to be predicted simply based on item similarity between them. Although they are all music players, "iPhone" is mainly a mobile phone, and "PSP" is mainly a portable game device. They are similar but also different therefore beg comparison with each other. It is clear that comparator mining and item recommendation are related but not the same. Our work on comparator mining is related to the research on entity and relation extraction in information extraction [2] [1] [14] [10] [5]. Specifically, the most relevant work is by Jindal and Liu [6] [7] on mining comparative sentences and relations. Their methods applied class sequential rules (CSR) [6] and label sequential rules (LSR) [6] learned from annotated corpora 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. However, their methods typically can achieve high precision but suffer from low recall [7] . However, ensuring high recall is crucial in our intended application scenario where users can issue arbitrary queries. To address this problem, we develop a weakly-

supervised bootstrapping pattern learning method by effectively leveraging unlabeled questions. Bootstrapping methods have been shown to be very effective in previous information extraction research [13] [12] [11] [9] [7]. 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 (comparative question identification), which is generally not required in IE task.

A. Mining Indicative Extraction Patterns

Our 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. 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 later 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.

B. Supervised Comparative Mining Method

In J&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 non-comparative, was used to create sequences and CSRs were mined. A Naïve Bayes classifier was trained using the CSRs as features. The classifier was then used to identify comparative sentences. J&L‟s method have been proved effective in their experimental setups. However, it has the following weaknesses:

The performance of J&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. It is also difficult to ensure the completeness of the keyword list.

Users can express comparative sentences or questions in many different ways. To have high recall, a large annotated training corpus is necessary. This is an expensive process.

Example CSRs and LSRs given in [7] are mostly a combination of POS tags and keywords. It is a surprise that their rules achieved high precision but low recall. They attributed most errors to POS tagging errors. However, we suspect that their rules might be too specific and over fit their small training set (about 2,600 sentences). We would like to increase recall, avoid over fitting, and allow rules to include discriminative lexical tokens to retain precision.

C. Weakly Supervised Method For Comparator Mining

Our weakly supervised method is a pattern-based approach similar to J&L‟s method, but it is different in many aspects: Instead of using separate CSRs and LSRs, our method aims to learn sequential patterns which can be used to identify comparative question and extract comparators simultaneously. 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 because the longest IEP is likely to be the most Specific and relevant pattern for the given question. 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 subsections. The evaluations confirm that our weakly supervised method can achieve high recall while retain high precision.

D. Association Mining Rules

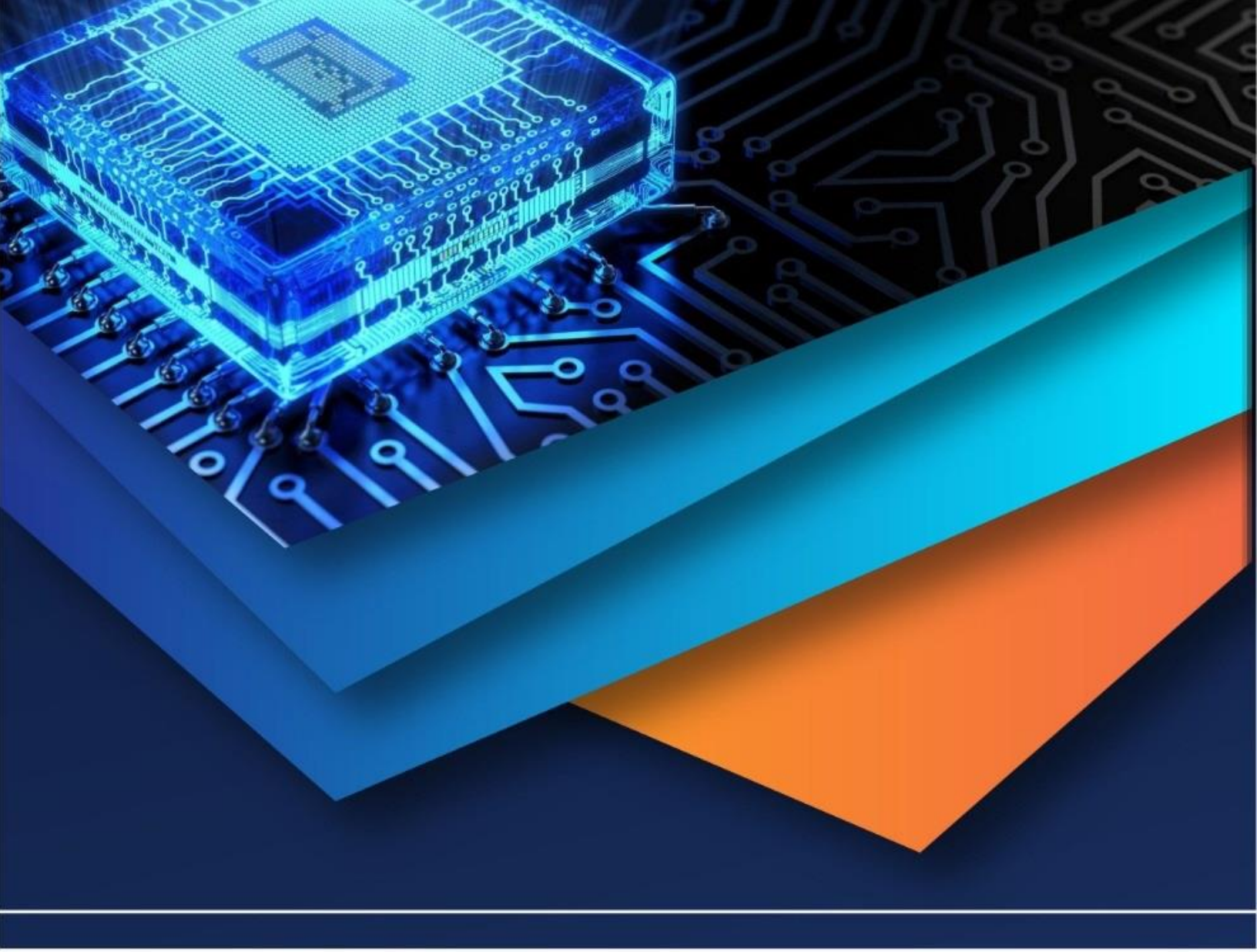
Because of the rapid growth in worldwide information, efficiency of association rules mining (ARM) has been concerned for several years. In this paper, an Apriori TID algorithm is proposed. This adopts a new count-based method to prune candidate item sets and uses generation record to reduce total data scan amount. Experiments demonstrate that our algorithm outperforms the other existing ARM methods.

III. CONCLUSION AND FUTURE WORK

In this paper, we present a novel weakly supervised method to identify comparative questions and extract comparator pairs simultaneously. 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. By leveraging large amount of unlabeled data and the bootstrapping process with slight supervision to determine four parameters, we found 328,364 unique comparator pairs and 6,869 extraction patterns without the need of creating a set of comparative question indicator keywords. The experimental results show that our method is effective in both comparative question identification and comparator extraction. It significantly improves recall in both tasks while maintains high precision. Our examples show that these comparator pairs reflect what users are really interested in comparing. 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 identify comparator aliases such as „LV' and „Louis Vuitton" and how to separate ambiguous entities such "Paris vs. London" as location and "Paris vs. 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] Mary Elaine Califf and Raymond J. Mooney. 1999. 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. Scaling personalized 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] Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In Proceedings of AAAI '06. 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, 7(1):3–10.
[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. In Proceedings of ACL '02, pages 41–47.
[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. Machine Learning, 34(1-3):233–272.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY