IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 2, Ver. V (Mar – Apr. 2015), PP 28-31
www.iosrjournals.org

Extraction of Data Using Comparable Entity Mining

Pravin Biradar [1,] Himanshu Sharma[2,] Parag Kothari[3,]
Shrikant Koparkar[4,] Ms Vidya Nikam.
[1,2,3,4],*Dept. of Computer Engineering DYPCOE,Akurdi Pune-33, India*

Abstract- *An important approach to text mining involves the use of natural-language information extraction. Information extraction (IE) distils structured data or knowledge from unstructured text by identifying references to named entities as well as stated relationships between such entities. IE systems can be used to directly extricate abstract knowledge from a text corpus, or to extract concrete data from a set of documents which can then be further analyzed with traditional data-mining techniques to discover more general patterns. Here is the methods and implemented systems for both of these approaches and summarize results on mining real text corpora of biomedical abstracts, job announcements, and product descriptions. Challenges that arise when employing current information extraction technology to discover knowledge in text are considered. Additionally, latest IEP which is accumulated in database can be used for the offline working. The system fetches content of current web page, stores and updates data to database, so that user can browse data online as well as offline.*
Keywords-*Indicative Extraction Patterns (IEP), Information Extraction (IE)*

I. Introduction

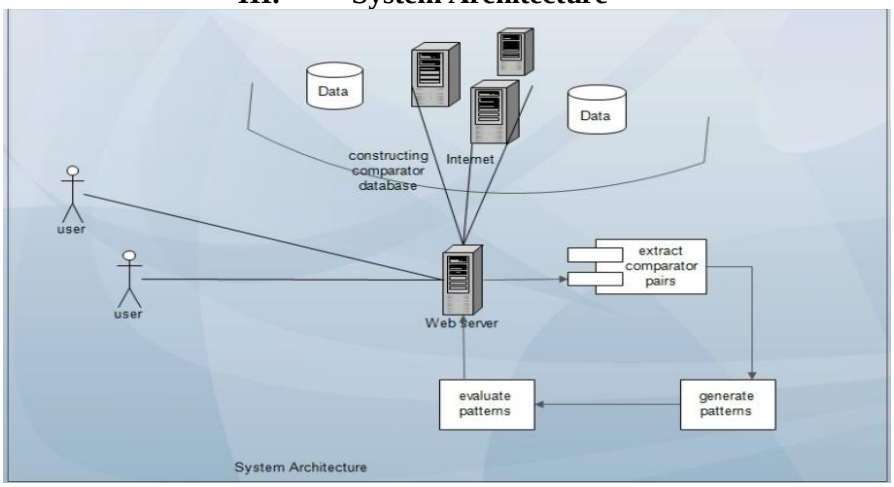
Database mining is motivated by the decision support problem faced by most large retail organizations. Progress in barcode technology has made it possible for retail organizations to collect and store massive amounts of sales data, referred to as the basket data. A record in such data typically consists of the transaction date and the items bought in the transaction. Very often, data records also contain customer-id, particularly when the purchase has been made using a credit card or a frequent-buyer card. Catalogue companies also collect such data using the orders they receive. It introduces the problem of mining sequential patterns over this data. An example of such a pattern is of such a pattern is that customers typically rent Star Wars, then Empire Strikes Back, and then Return of the Jedi. Note that these rentals need not be consecutive. Customers who rent some other videos in between also support this sequential pattern. Elements of a sequential pattern need not be simple items. Fitted Sheet and at sheet and pillow cases, followed by comforter, followed by drapes and ruffles is an example of a sequential pattern in which the elements are sets of items. One of the most important ways of evaluating an entity or event is to directly compare it with a similar entity or event. The objective of this work is to extract and to analyze comparative sentences in evaluative texts on the web, e.g., customer reviews, forum discussions, and blogs[1]. This task has many important applications. After a new product is launched, the manufacturer of the product wants to know consumer opinions on how the product compares with those of its competitors. Extracting such information can help businesses in its marketing and product benchmarking efforts. The main focus has been on sentiment classification and opinion extraction (positive or negative comments on an entity or event.

II. Related Work

It explores the use of Support Vector Machines (SVMs) for learning text classifier from examples. It analyzes the particular properties of learning with text data and identifies why SVMs are appropriate for this task.[6] It propose an approach by exploiting a large number of available pairs of question-answer documents in order to search the best similar question to user's question. [1] Text classification is the technique that increased in importance over the last period when the documents became digital. Clustering analysis is proposed in the current paper as an a priori step in the process of Bayesian classification, as a filter of the words used in the aggregation of the probabilities.[8] Automatic Text Categorization and Clustering are becoming more and more important as the amount of text in electronic format grows and the access to it becomes more necessary and widespread. [3]

In linguistics, comparatives are based on specialized morphemes, more/most, -er/-est, less/least and as, for the purpose of establishing orderings of superiority, inferiority and equality, and than and as for making a 'standard' against which an entity is compared. [7] The extraction patterns are generated from tagged text and untagged text. For tagged text, AutoSlog is a dictionary construction system which uses heuristic rules that creates extraction patterns automatically. AutoSlog-TS is the system to generate domain specific extraction pattern automatically without annotated training data. A user only needs to provide sample texts and spend some time to filtering and labelling the extraction pattern. [4]

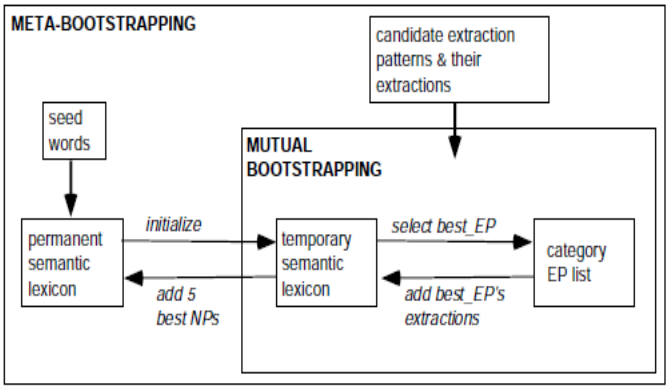
III. System Architecture



IV. Algorithm And Method

i) Pattern Generation

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, generalized patterns, specialized patterns.



Lexical patterns: Lexical patterns indicate sequential patterns consisting of only words and symbols ($C, #start, and #end). They are generated by suffix tree algorithm (Gusfield, 1997) with two constraints: A pattern should contain more than one $C, and its frequency in collection should be more than an empirically determined number β.

Generalized patterns: A lexical pattern can be too specific. Thus, we generalize lexical patterns by replacing one or more words with their POS tags. $2n - 1$ generalized patterns can be produced from a lexical pattern containing N words 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.

ii) Bootstrapping

The bootstrapping process starts with a single IEP. From it, 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. Patterns evaluated as reliable ones are IEPs and are added into an IEP repository.

iii) Mutual Bootstrapping

Some IE tasks, the set of possible extractions is finite. For example, extracting country names from text is straightforward because it is easy to define a list of all countries. However, most IE tasks require the extraction of potentially open ended set of phrases. Most IE system use both semantic lexicon of known phrases and a dictionary of extraction patterns to recognize relevant noun phrases. The semantic lexicon can also support the use of semantic constraints in the extraction pattern. The goal is to automate the construction of both the lexicon and extraction pattern for a semantic category using bootstrapping. The heart of this approach is based on the observation that the extraction pattern can generate new example of semantic category, which in turn can be used to identify new extraction pattern, this process is referred as mutual bootstrapping. Mutual bootstrapping process begins with a text corpus and handful of predefined seed words for a semantic category. Before bootstrapping begins, the text corpus is used to generate the candidate extraction pattern. And then applied the extraction patterns to corpus and recorded their extraction. This extraction pattern is then used to propose new lexicon that belong in the semantic lexicon. The fig.1 outlines the mutual bootstrapping algorithm. All of its iteration are assumed to be category members and are added to the semantic lexicon (SemLex). Then the new best extraction pattern is identified, based on both the original seed word and new words that are just added to the lexicon, and the process repeats. Since the semantic lexicon is constantly growing, extraction patterns need to be recorded after each iteration.[5]

Generate all candidate extraction patterns from the training corpus using AutoSlog.

Apply the candidate Extraction patterns to the training corpus and save the patterns with their extraction to EP data
SemLex={seed _words}
Cat_EPlist={}

Mutual Bootstrapping Loop
1. Score all extraction patterns in EPdata.
2. best_EP=the highest scoring extraction patterns not already in Cat_EPlist
3. Add best_EP to cat_EPlist
4. Add best_EP's extraction to SemLex.
5. Go to step 1

V. Experimental Setup

Operating system- Windows 7
Technology - .NET
IDE- Microsoft Visual studio 2010
Database- SQL Server 2008 R2 Express

VI. Result

To ensure high precision and high recall, proposed system develops a weakly-supervised bootstrapping method for comparative question identification and comparable entity extraction by leveraging a large amount of data. Performance is better than previous works in terms of sequential patterns generation. Comparative Question Identification and Comparator Extraction is efficient. It reduces time and cost for searching the same result sets.

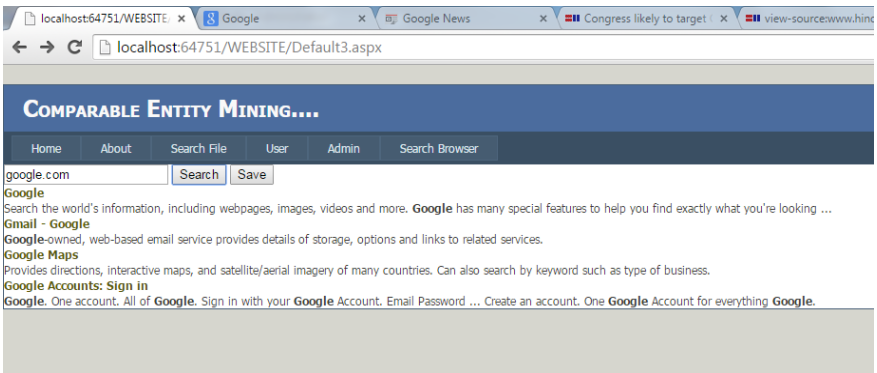


Fig: **Online search**

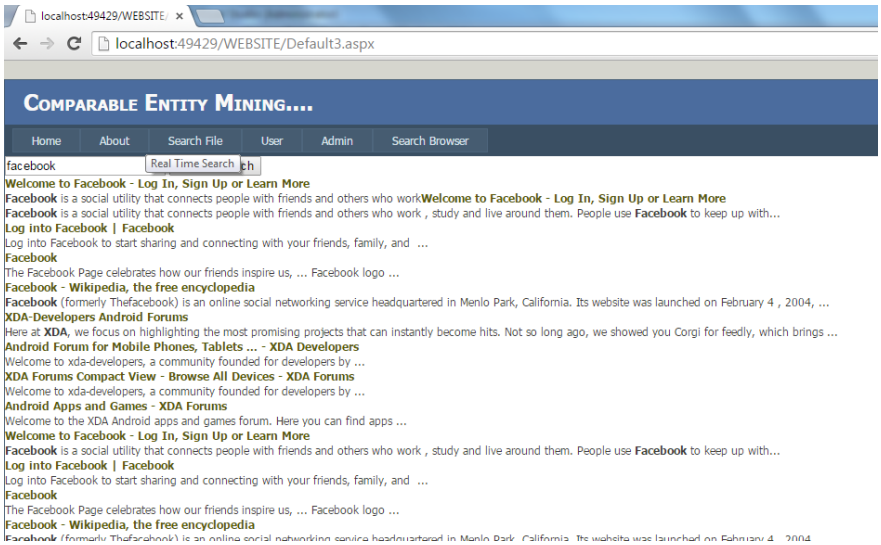


Fig: Offline search

The figure shows about how offline and online working can be done by the user .

VII. Conclusion

The system is developed to solve the problem of decision making using comparable entity mining. The system identifies comparative questions and extracts comparator pairs simultaneously with the help of novel weakly supervised method. By supplying large un-labelled data and the bootstrapping process, it found several unique comparator pairs and many extraction patterns. The underlying mining results which are comparable can be used for a e-commerce applications. Also, results provide useful information to companies which want to identify their competitors. Also, these results can provide useful information to companies which want to identify their competitors. It can help in fast surfing and searching. Comparable entity mining helps in decision making process, along with this it helps in making search entity simple and ranking wise abstraction of data. It will compare the correctness of the entity with multiple search engines and provide desired result. Additionally, latest IEP which is accumulated in database can be used for the offline working. The system fetches content of current web page, stores and updates data to database, so that user can browse data online as well as offline

References

[1]. Shasha Li, Chin-Yew Lin, Young-In Song and Zhoujum Li. 2013.Comparable Entity Mining From Comparative Questions. National University of Defense Technology, Changsha, China
[2]. ZornitsaKozareva, 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.
[3]. Nitin Jindal and Bing Liu.2006a.Identifying comparative sentences in text documents. In Proceedings of SIGIR '06, pages 244–251.
[4]. Ellen Riloff.1996 Automatically Generating Extraction Pattern from Untagged Text
[5]. 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.
[6]. Guoping Hu1, Jingjing Liu2, Hang Li, Yunbo Cao, Jian-Yun Nie3, and Jianfeng Gao. A Supervised Learning Approach to Entity Search. Microsoft Research Asia, Beijing, China.
[7]. Nitin Jindal and Bing Liu. Mining Comparative Sentences and Relations.
[8]. Greg Linden, Brent Smith and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, pages 76-80.
[9]. Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In Proceedings of WWW '02, pages 517–526.
[10]. DragomirRadev, Weiguo Fan, Hong Qi, and Harris Wu and AmardeepGrewal. 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]. Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text.