**Comparable Entity Mining from Comparative Questions**

**ABSTRACT**

Comparing one thing with another is a typical part of human being conclusion making method. On the other hand, it's not always an easy task to know what to compare in addition to which are the solutions. With this cardstock, we provide the novel strategy to automatically mine comparable entities from comparable questions which users posted online to address this difficulty; we experience a weakly monitored bootstrapping method for comparative question identification and comparable entity extraction by using large collection of online question archive. The trial and error final results present our approach achieves F1-measure associated with 82. 5 percent in comparison issue identification in addition to 83. 3 percent in comparable entity extraction. Both equally drastically outperform a current state-of the-art work approach. Additionally, our position final results present very importance to help user’s comparison intents in web.

**1. INTRODUCTION**

Comparing alternative options is one essential step in decision-makings that we carry out every day. For example, if someone is interested in certain products or services such as digital cameras or treatments, he or she would want to know what the alternatives are and how they compare to each other before making a purchase decision. 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 user’s 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” or as “market segments that their products are targeting,” 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, 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.

The goal of this work is, mining comparators from comparative questions and furthermore, provides and rank comparable entities for a user’s input entity appropriately. 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 potentially comparable entities. We leverage this insight and develop a weakly supervised bootstrapping method to identify comparative questions and extract comparators simultaneously.

**Objective of the Project**

Comparing one thing with another is a typical part of human decision making process. However, it is not always easy to know what to compare and what are the alternatives. In this paper, we present a novel way to automatically mine comparable entities from comparative questions that users posted online to address this difficulty. To ensure high precision and high recall, we develop a weakly supervised bootstrapping approach for comparative question identification and comparable entity extraction by leveraging a large collection of online question archive. The experimental results show our method achieves F1-measure of 82.5 percent in comparative question identification and 83.3 percent in comparable entity extraction. Both significantly outperform an existing state-of the-art method. Additionally, our ranking results show highly relevance to user’s comparison intents in web.

**2. LITERATURE SURVEY**

The work on comparator mining is connected to the research on entity and relation extraction in information extraction. Exclusively, a large amount of relevant work is done by Jindal and Liu [1], [15] on mining comparative sentences and relations. Their ISSN(Online): 2320-9801 ISSN (Print): 2320-9798 International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2014 10.15680/ijircce.2014.0212018 Copyright to IJIRCCE www.ijircce.com 7253 methods applied class sequential rules (CSR) and label sequential rules (LSR) learned from annotated corpora to identify comparative sentences and extract comparative relations correspondingly in the news and review domains. Bootstrapping methods proved to be very effectual in previous information extraction research [6], [8]. Bootstrapping technique is applied to pull out entities with a specific relation. INFORMATION EXTRACTION (IE) deals with locating specific pieces of data in natural-language documents, thereby mining structured and meaningful information from unstructured and/or a semi-structured one is called as Information Extraction [5]. One type of IE, named entity recognition, involves identifying references to particular kinds of objects such as names of people, companies, and locations. There are mainly three methods used for information extraction as [6], [7], [8] given below, 1. Rule based Extraction: One approach of IE is to automatically learn pattern-based extraction rules for identifying each type of entity or relation. For example, the system developed by Rapier in [9]. Patterns are expressed in an enhanced regular-expression language; and a bottom-up relational rule learner is used to induce rules from a corpus of labeled training examples. Inductive Logic Programming (ILP) [10] has also been used to learn logical rules for identifying phrases to be extracted from a document [11], [12]. 2. Pattern based extraction: Pattern based approaches build on annotated text fragments (the patterns), where words/phrases are labeled with linguistic information, e.g. POS-tag, word lemma, or syntactic information. Those patterns are matched against linguistically annotated text to detect relationships [13]. 3. Supervised Learning: Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). However, supervised training of accurate entity and relation extractors is costly, requiring a substantial number of labeled training examples for each type of entity and relation to be extracted. Because of this, many researchers have explored semi-supervised learning methods that use only a small number of labeled examples of the predicate to be extracted, along with a large volume of unlabeled text [14]. All the above information extraction methods can be used for comparator methods as in [2], [3], [4] by Li Shasha, Jindal and Liu in [1],[15]. A. Design Considerations: Supervised comparative mining method was proposed by Jindal and Liu [1], [15] which is a baseline for comparison. It focuses mainly on two rules mentioned as Class Sequential Rule (CSR) & Label Sequential Rule (LSR) as described below. a. Class Sequential Rule (CSR): It is a classification rule which maps a sequence pattern S (s1, s2 . . . sn) (a class C. C is either comparative or noncompetitive). Every CSR is associated with two parameter support and confidence. b. Label Sequential Rule (LSR): It maps an input sequence pattern S (s1, s2 . . . si . . . sn) to a labeled sequence S (s1, s2 . . . li . . . sn) by replacing token si in the input sequence with a designated label (li) and this token is referred as the anchor. Jindal and Liu [1] method have been proved effective in their experimental setups. However, it has the some drawbacks as given below, the performance of Jindal and Liu’s method depends mainly on a set of comparative sentence indicative• keywords [3]. 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 CSRs and LSRs introduced by Jindal and Liu in [15] mostly a combination of POS tags and keywords. It is surprise that their rules achieved high precision but low recall. ISSN(Online): 2320-9801 ISSN (Print): 2320-9798 International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2014 10.15680/ijircce.2014.0212018 Copyright to IJIRCCE www.ijircce.com 7254 B. A Weakly Supervised Method for Comparator Mining: To resolve the conflict in extracting comparative questions and its comparator with high precision as well as with high recall a Weakly Supervised Bootstrapping method is introduced by Li Shasha in [2].

**Bootstrapping Method**

To end this conflict inside getting rid of comparison issues and its particular comparator along with excessive accurate and also along with excessive recall a new Weakly Supervised Bootstrapping technique is introduced by Li Shasha in [2]. 1. Indicative Extraction Patterns Mining: Indicative Extraction Pattern (IEP) is a sequential pattern which is used for identification of comparative questions along with comparator extraction 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. If a question matches multiple IEPs, the longest IEP is used. Therefore, instead of manually creating a list of indicative keywords, we create a set of IEPs automatically, referred as weakly supervised method which is iterative. The two key steps in this algorithm are pattern generation and pattern evaluation. 2. Pattern Generation: The weakly supervised IEP mining is highly based on two key assumptions as [3], [16], [17]. 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 key assumptions, bootstrapping algorithm designed. To generate sequential patterns, Li Shasha in [1],[3] used surface text mining method introduced in [2]. In this method, comparators in the question are replaced by symbol $Cs in any given comparative question and its comparator pair. The symbol #start is attached to the beginning of the each sentence and the symbol #end at the end of sentence. Li Shasha in [3] used some heuristic rules and phrase chunking for diversity reduction of sequence data and mine potential patterns. Following three kinds of sequential patterns can be generated from sequences of questions as: a. Lexical Patterns: These patterns indicate sequential patterns consisting of only words and symbols ($C, #start, and #end). b. Generalized Patterns: A lexical pattern is too specific for matching. So lexical patterns are generalized by replacing one or more words their POS tags. c. Specialized Patterns: Pattern specialization is done by adding POS tags to all comparator slots. For example, from the lexical pattern ''and the question 'Paris or London?', '' will become specialized pattern. ISSN(Online): 2320-9801 ISSN (Print): 2320-9798 International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2014 10.15680/ijircce.2014.0212018 Copyright to IJIRCCE www.ijircce.com 7255 Note that in this method, lexical patterns are used to generate generalized patterns and the combined set of generalized patterns and lexical patterns are used to generate specialized patterns [1],[3]. 3. Pattern evaluation: Bootstrapping gives very few reliable comparator pairs in its early stage. Hence for discovering more reliable pair’s, pattern evolution operation is performed. In this case, the value might be underestimated which could affect the effectiveness of on distinguishing IEPs from non-reliable patterns. This problem is mitigated by a look-ahead procedure. The next step is to rank possible comparators for a user’s input [1], [3]. C. Comparator Extraction: By employing IEPs, it is easy to identify comparative questions and collect comparator pairs from available data. For given question and an IEP, comparator extraction process is described in [1], [2], [3], [4] as follows: 1. Generate sequence for the comparative question: If the IEP is a pattern without generalization, then tokenize the questions and the list of resulted tokens is the sequence. Otherwise, phrase chunking is needed. The sequence is a list of resulted chunks. 2. Check whether sequence of the question matches with the given pattern: If IEP is a specialized pattern, the POS tag sequence of extracted comparators should follow the constraints specified by the pattern. However, a result of [3] shows about 67 % comparative questions can match to multiple patterns, and from 11 % comparative questions, we can extract different comparator pairs. Li Shasha in [3], [4] examined three different strategies to solve the issue of comparator extraction. D. Comparator Ranking: The comparability and graph based methods are examined rank possible comparators for user’s input [1], [3], [18] which are described below, 1. Comparability-Based Ranking Method: Frequent comparison of entity with particular entity would make comparator more interesting. 2. Graph Based Ranking Method: Frequency is consider as efficient parameter for comparator ranking but the frequency-based ranking method [3] can suffer when an user input occurs rarely in collection of questions; for example, suppose all possible comparators to the input are compared only once in questions. In this case, this method may fail to results correct ranking result. Hence in addition to it representing ability should also be considered. We regard a comparator representative if it is frequently used as a baseline while making comparison of interested entity. Graph based page rank method is one of the solutions to get ability. A comparator can be considered as valuable comparator in ranking if it is compared to too many other important comparators including the input entity. Based on this idea, Page Rank algorithm is examined to rank comparators for a given input entity, which combine frequency and represent ability [3].

**Existing System**

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. 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.

**Disadvantage:**

1. Things also get more complicated when an entity has several functionalities.
2. difficult to decide if two entities are comparable or not

**Proposed System**

To our best knowledge, this is the first attempt to specially address the problem on finding good comparators to support users’ comparison activity. We are also the first to propose using comparative questions posted online that reflect what users truly care about as the medium from which we mine comparable entities. Our weakly supervised method achieves 82.5 percent F1-measure in comparative question identification, 83.3 percent in comparator extraction, and 76.8 percent in end-to-end comparative question identification and comparator extraction which outperform the most relevant state-of-the-art method by Jindal and Liu significantly.

**Advantages of Proposed System:**

1. Provide and rank comparable entities for a user’s input entity appropriately.
2. The results would be very useful in helping users’ exploration of alternative choices by suggesting comparable entities based on other users’ prior requests.

**HARDWARE REQUIREMENTS:**

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System : Windows XP
* Programming Language : JAVA
* Front End : AWT, Swing
* Back End : MySql

**Class diagram:**

****

**Use case diagram:**



Use case Diagram for User:



**Sequence diagram:**



**Collaboration diagram:**

****

**Component diagram:**

Component Diagram for Admin:

****

Component Diagram for User:



**Deployment diagram:**

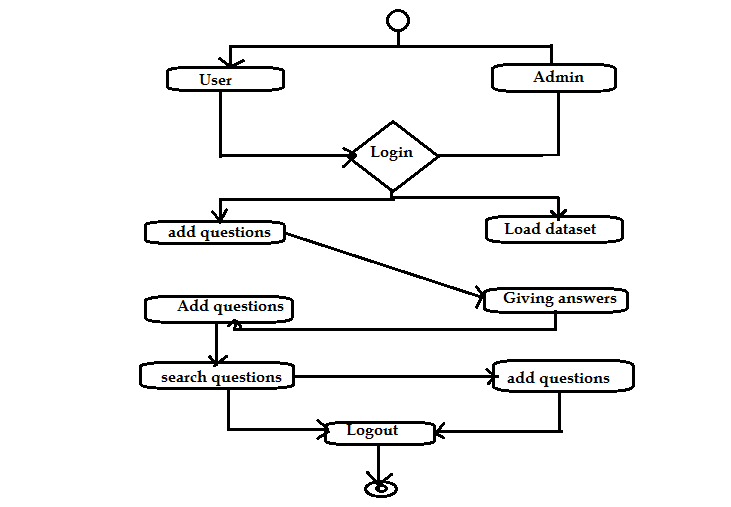
Deployment Diagram for Admin:



Deployment Diagram for User:

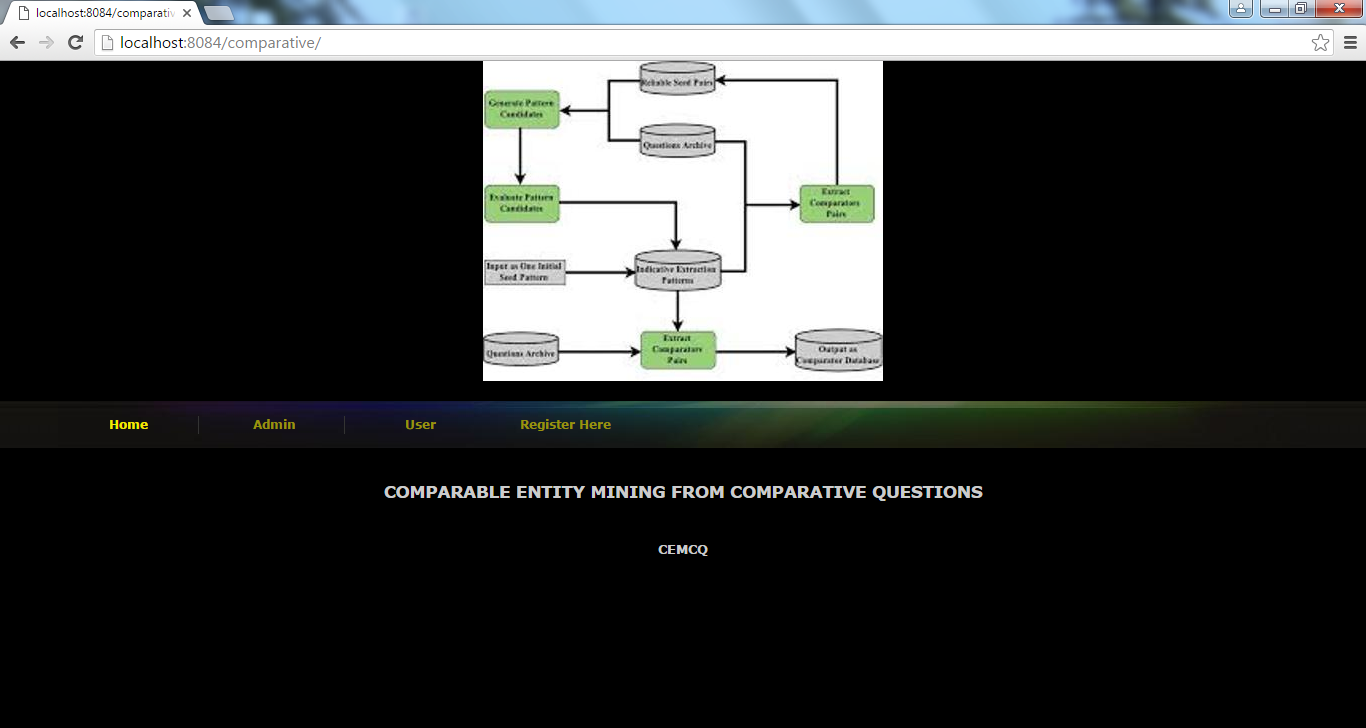


**Activity diagram:**

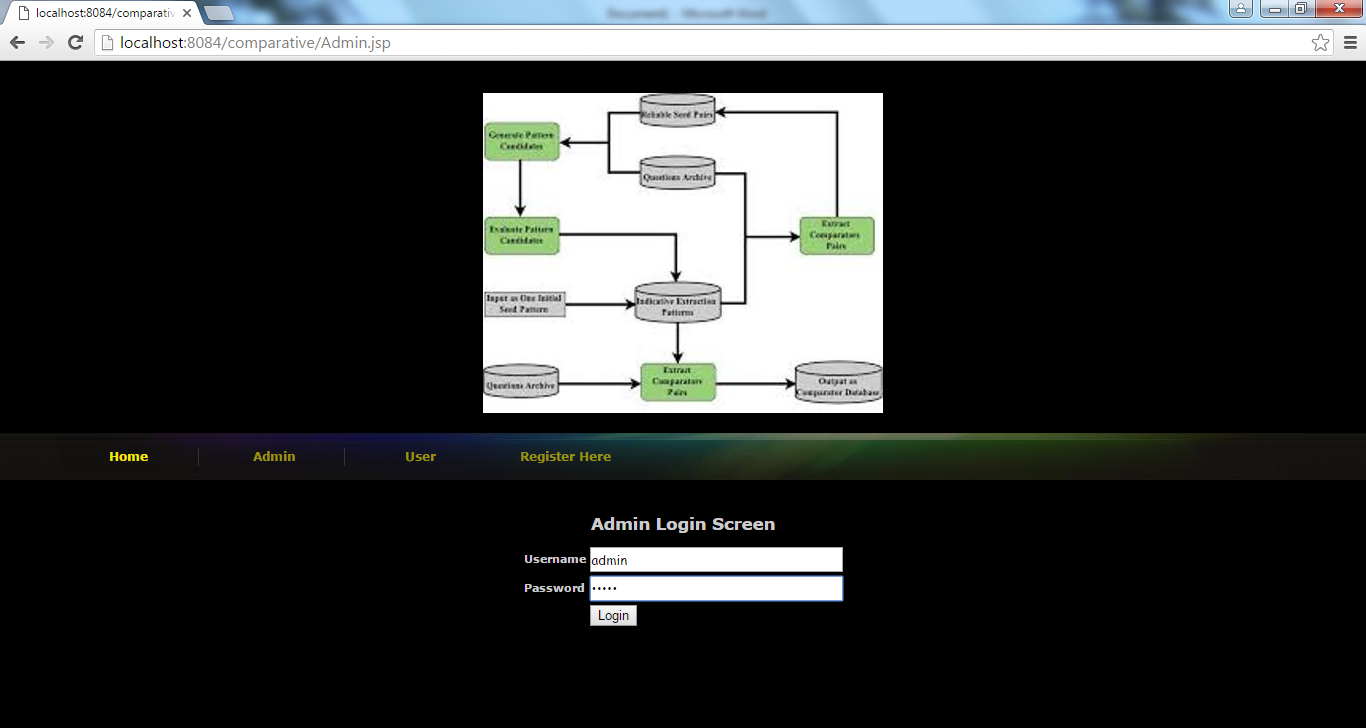


**7. SCREEN SHOTS**

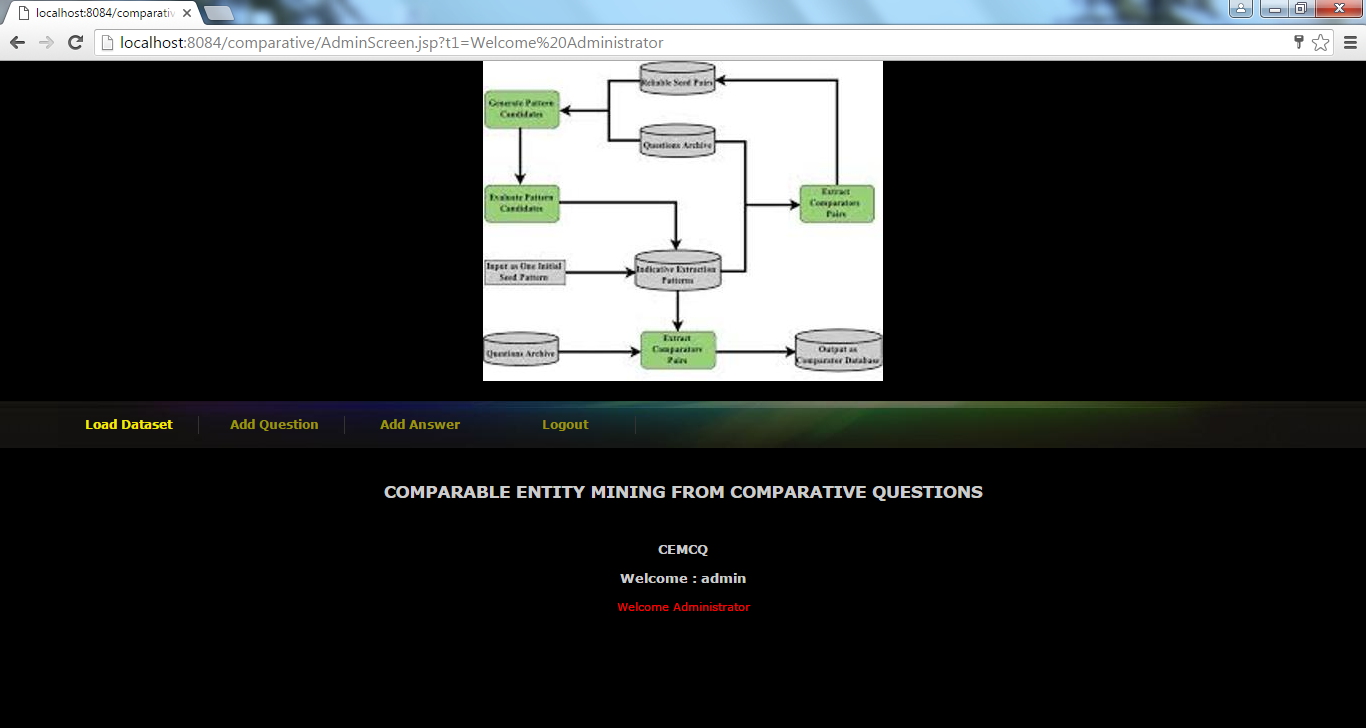
Index page



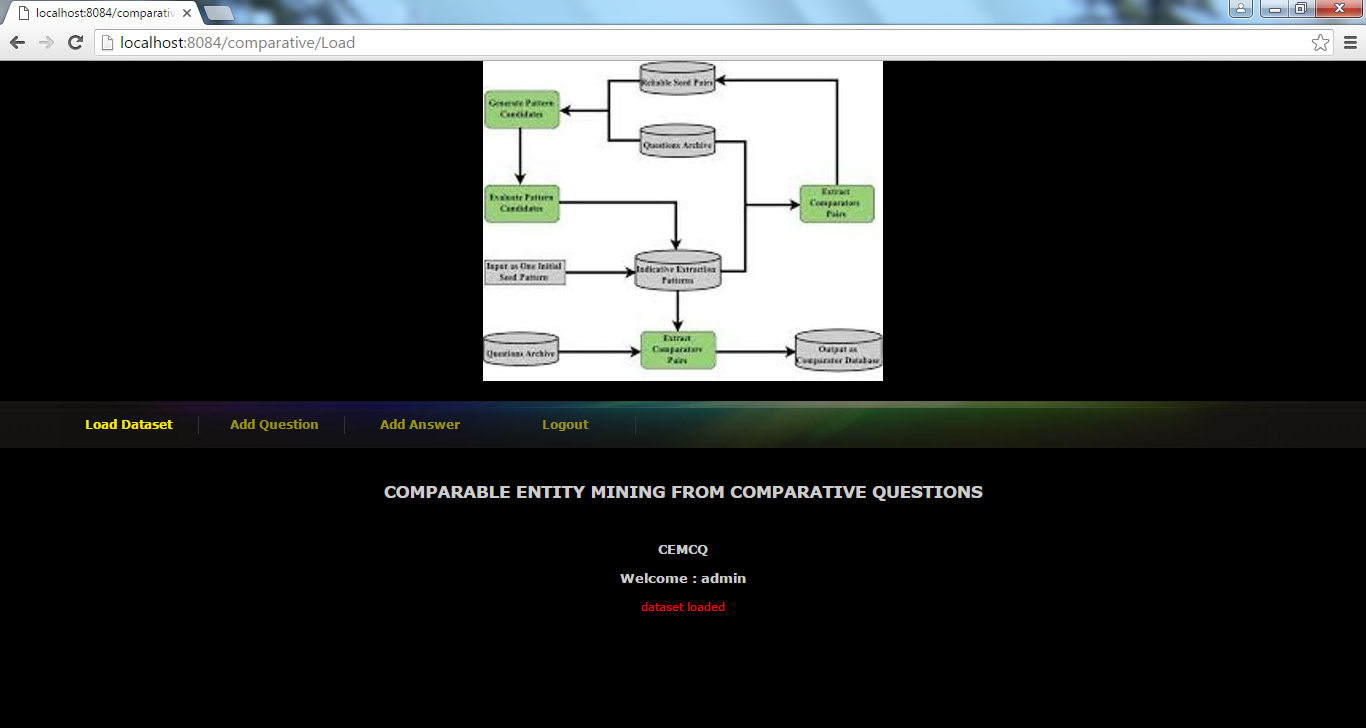
Admin Login



Admin Screen

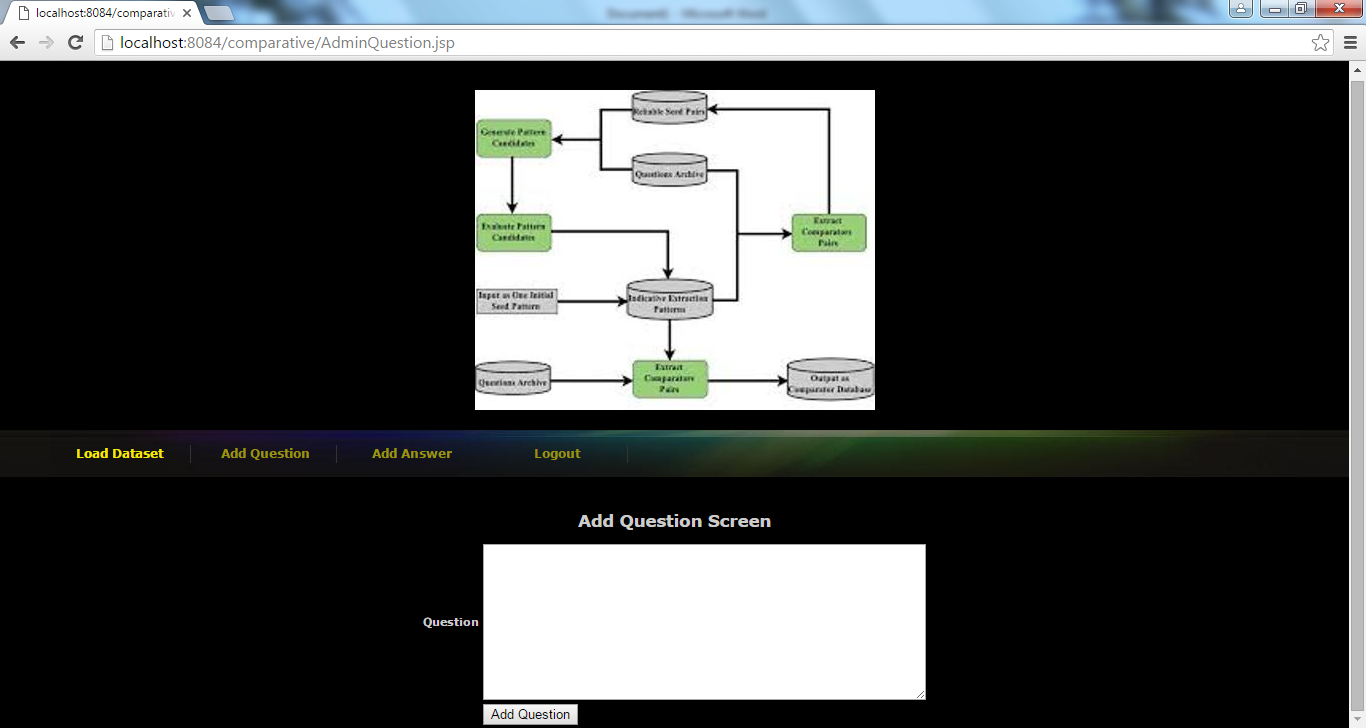


Using load dataset function admin load dataset questions and all other questions users or admin has added manually. While loading dataset application will take only comparative entities

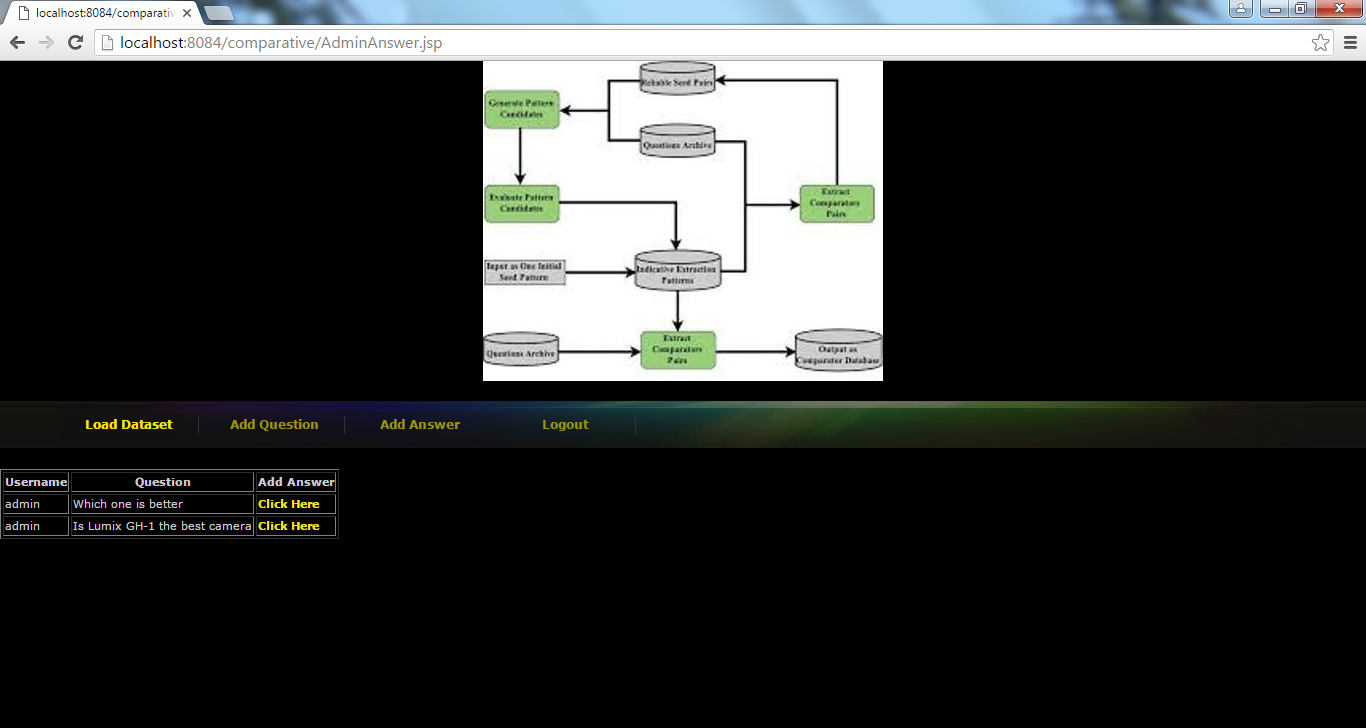


Above screen after loading dataset

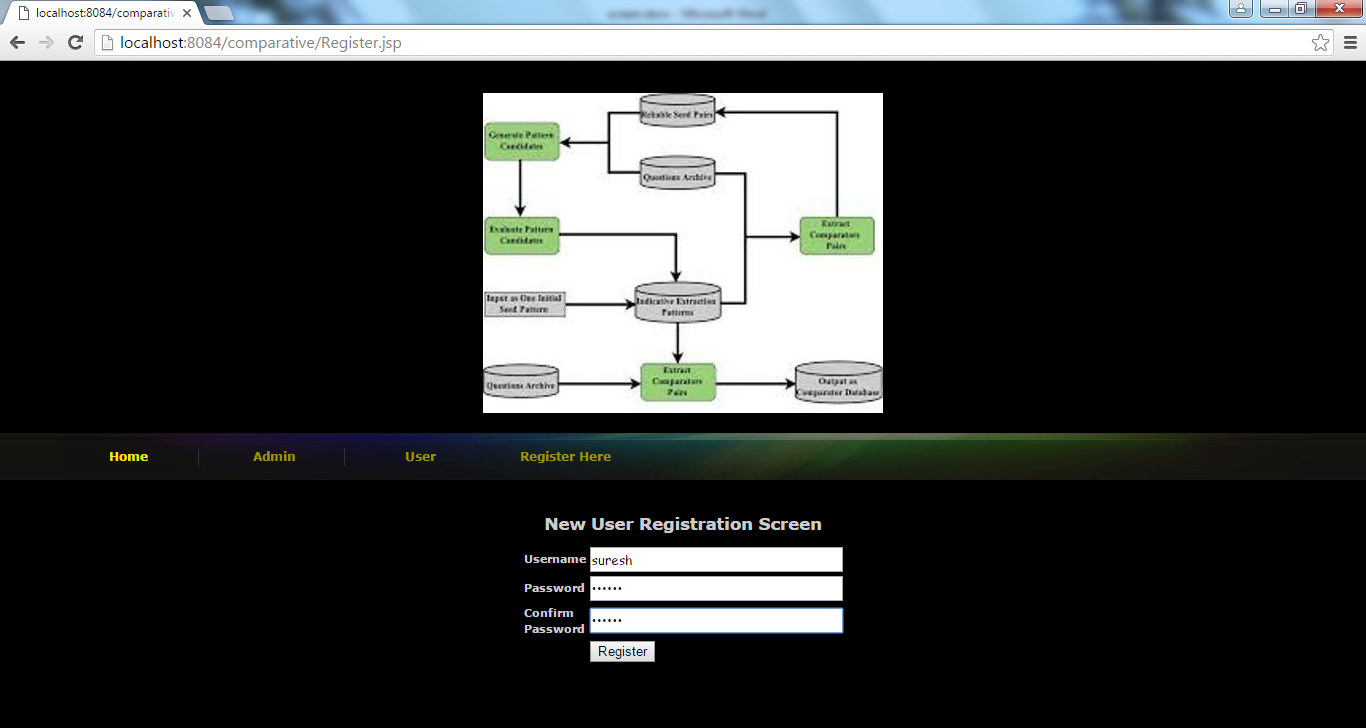
Admin to add question



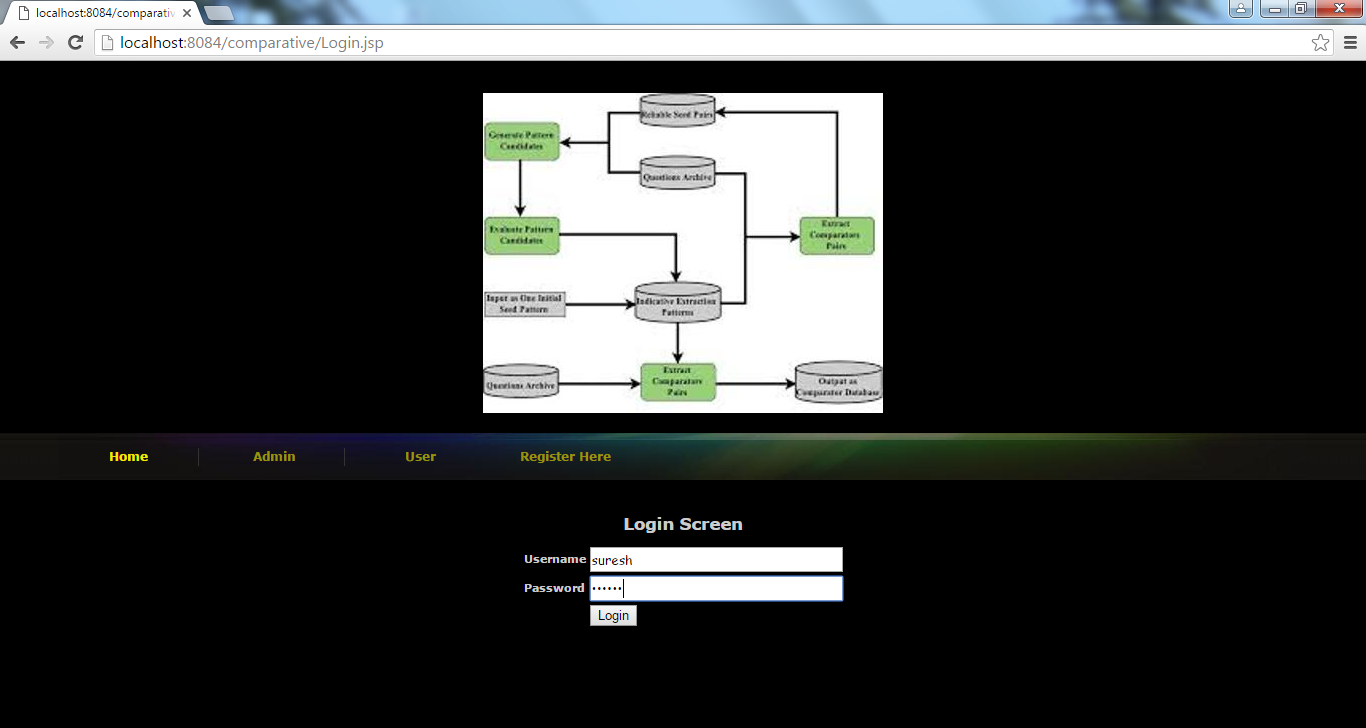
Admin to give answer



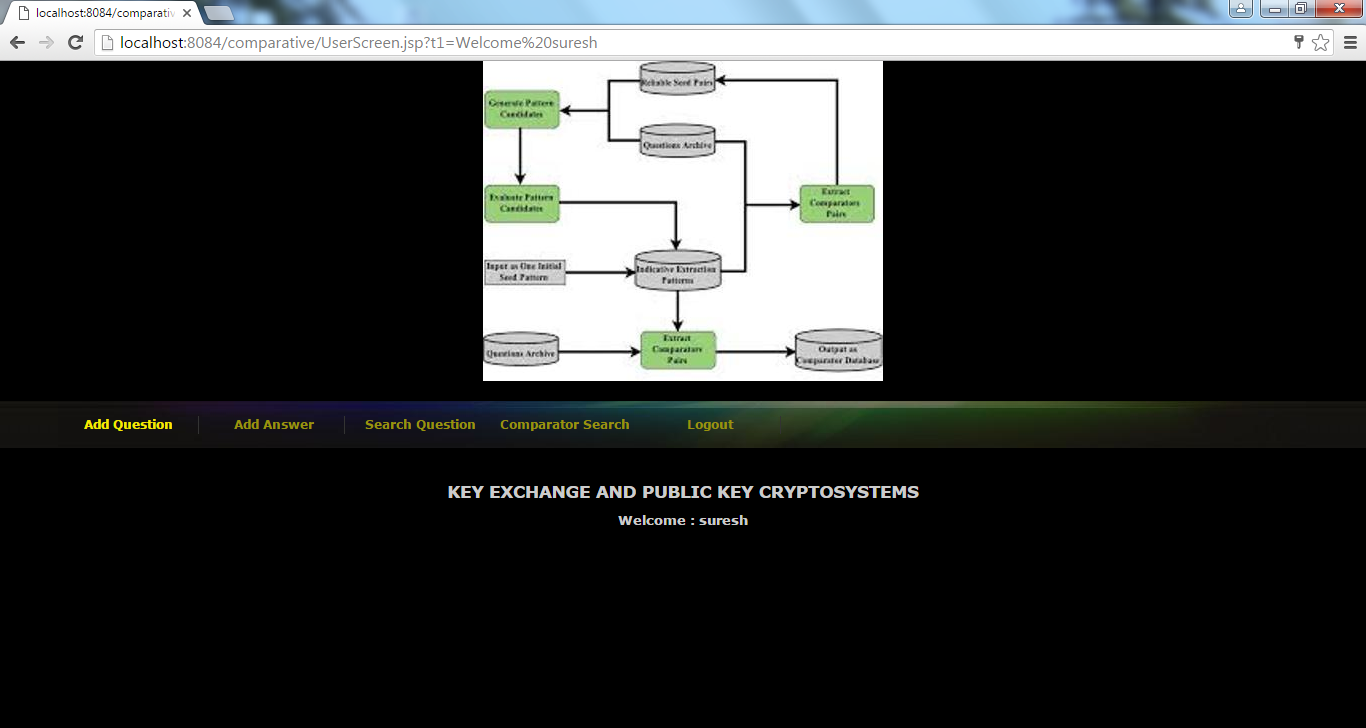
New user registration



New user login

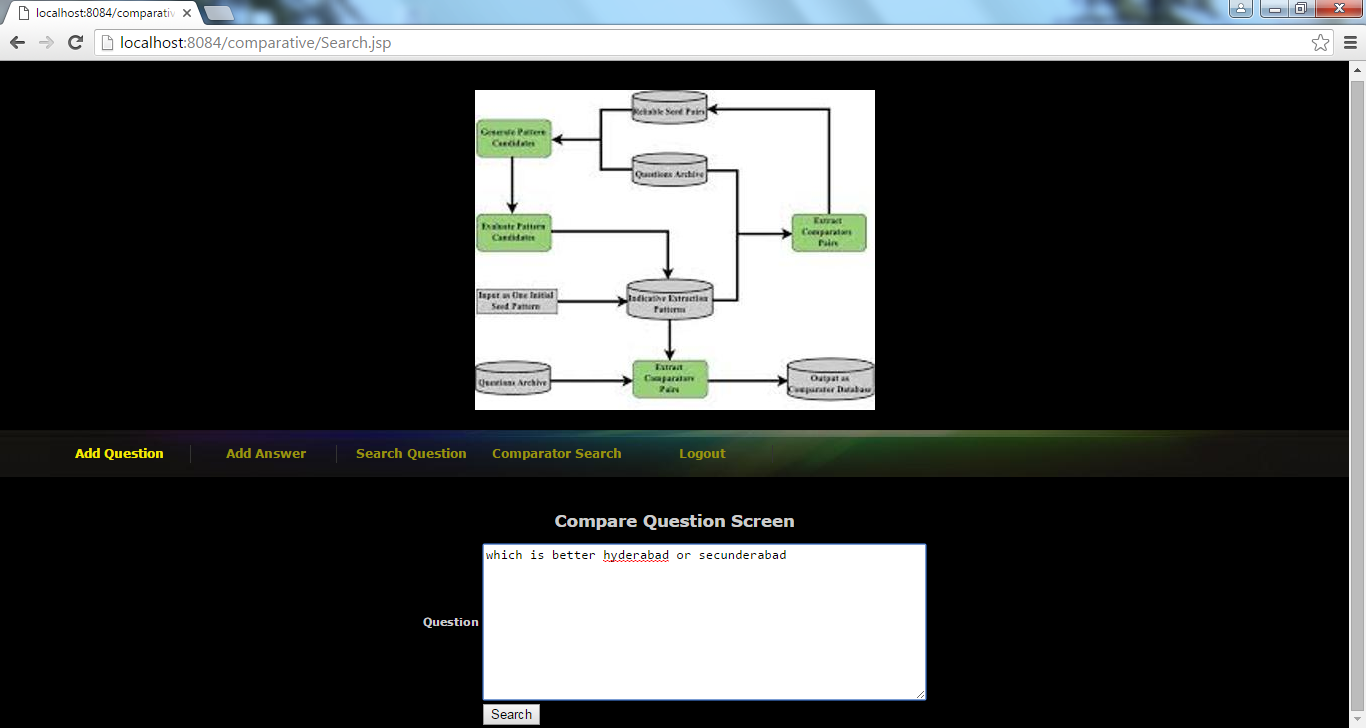


User screen

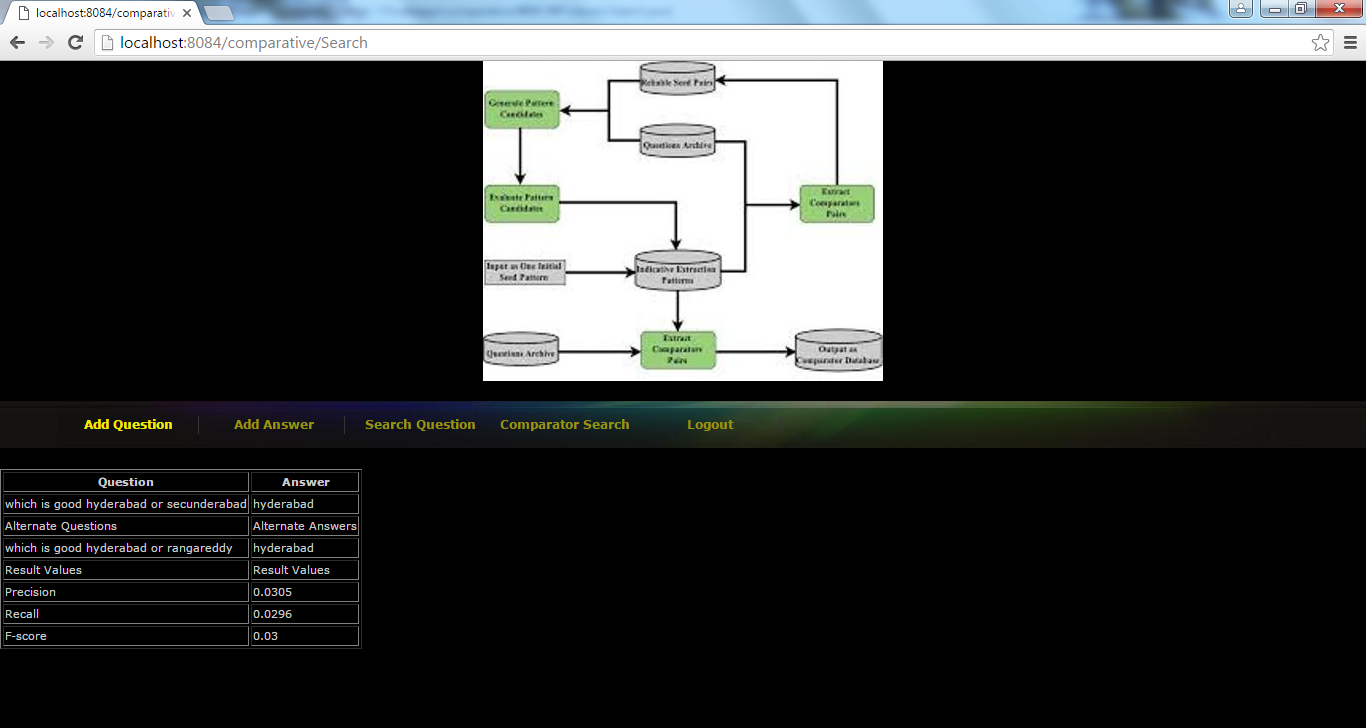


Users also can add and answer question

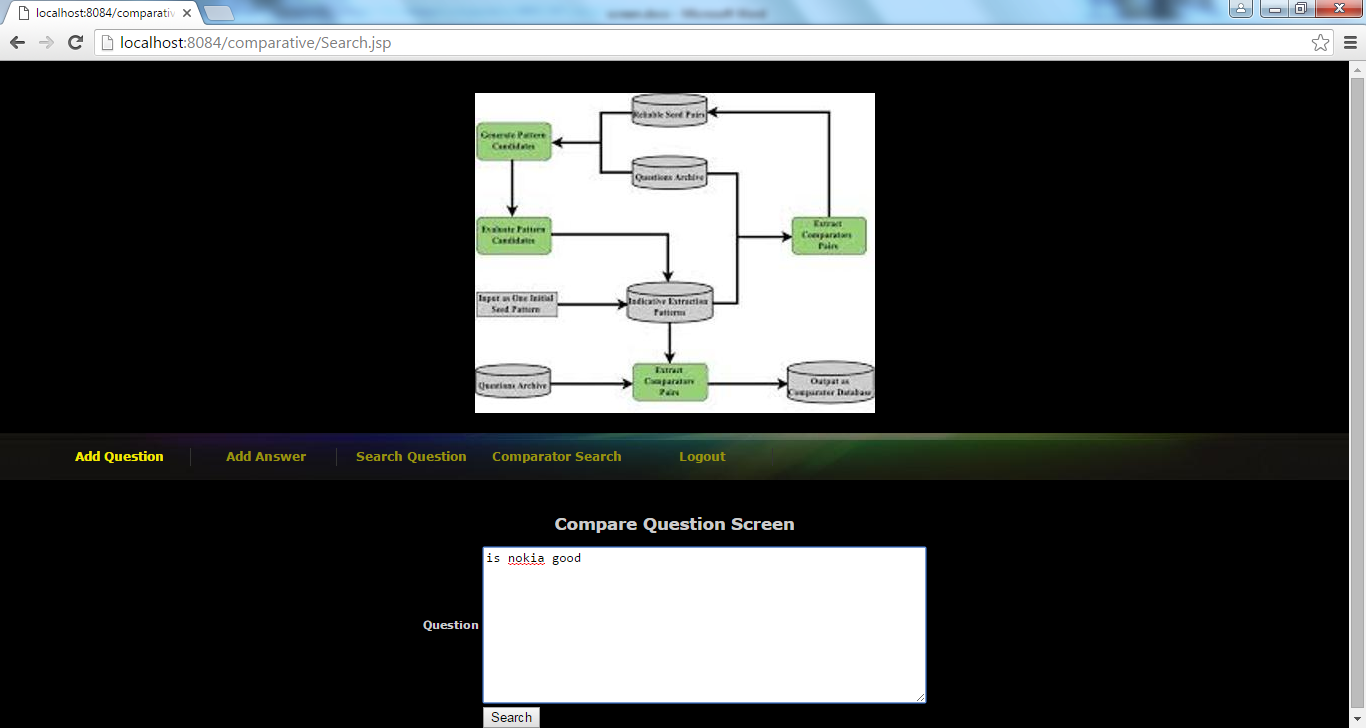
Search question screen



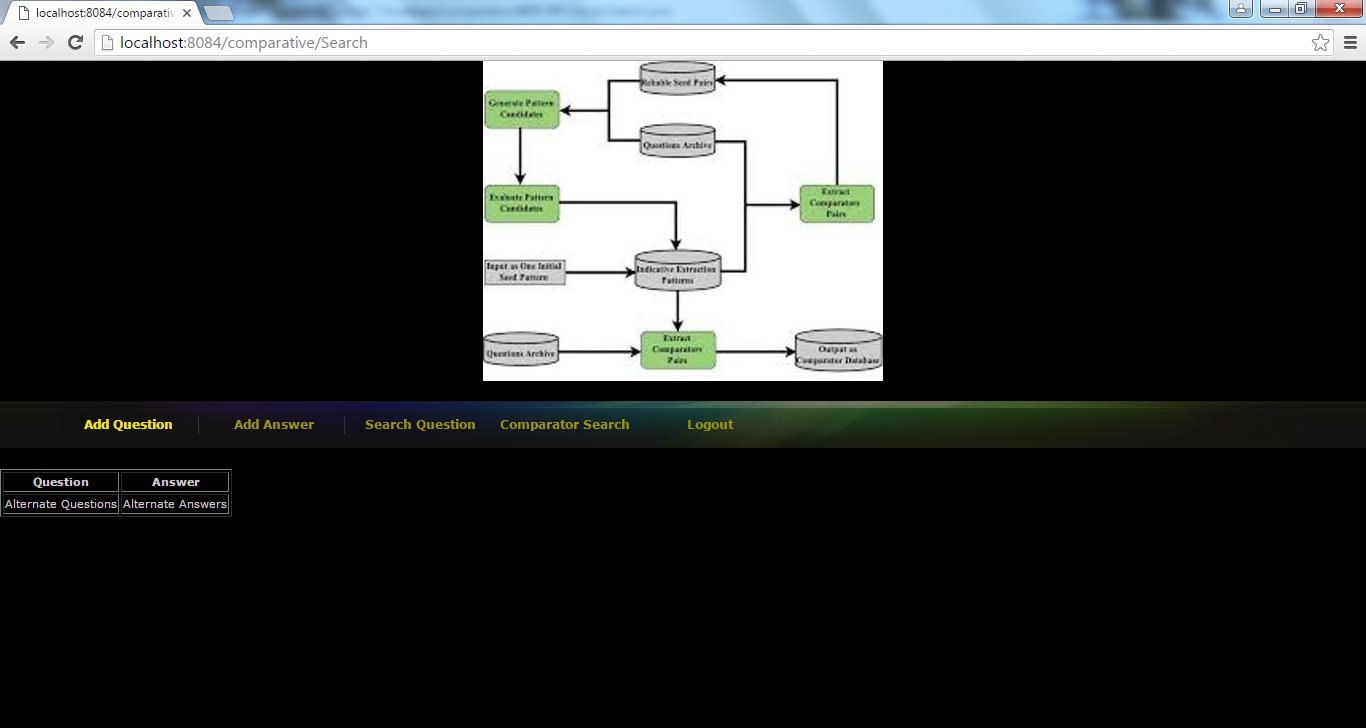
Query answer



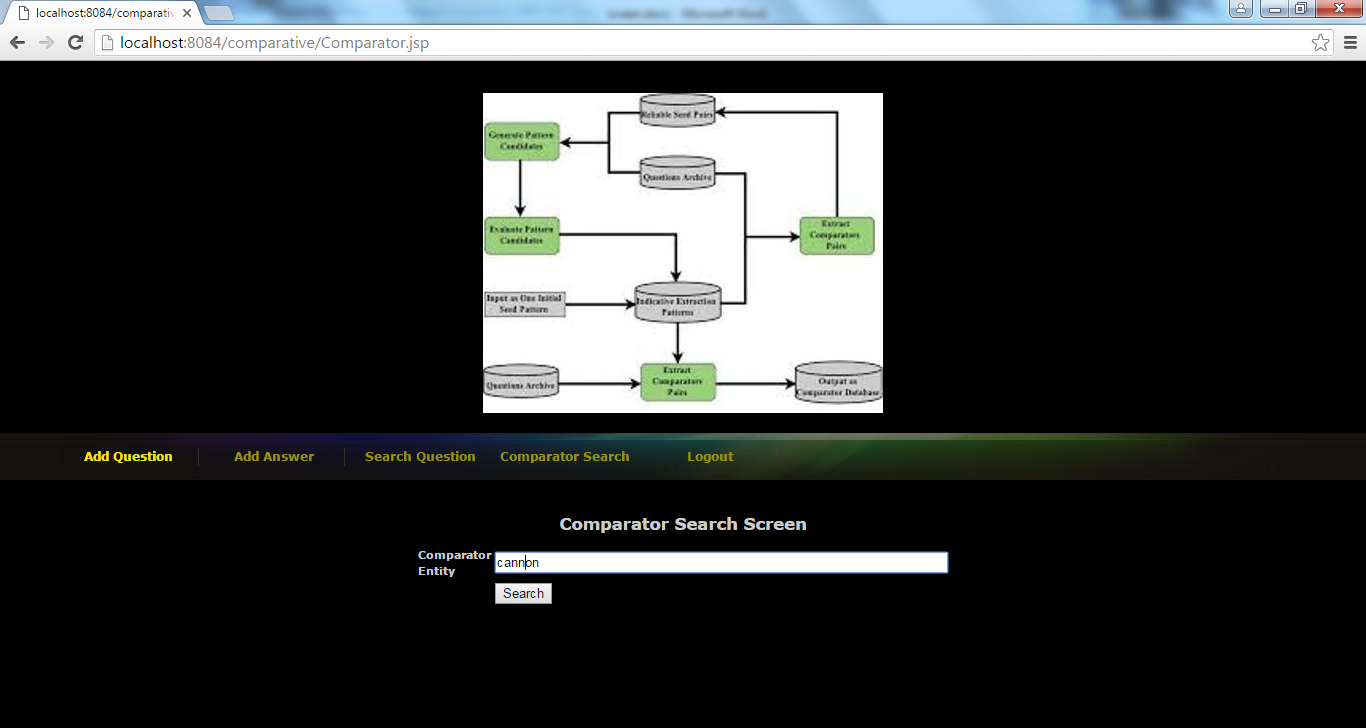
Un-comparative question



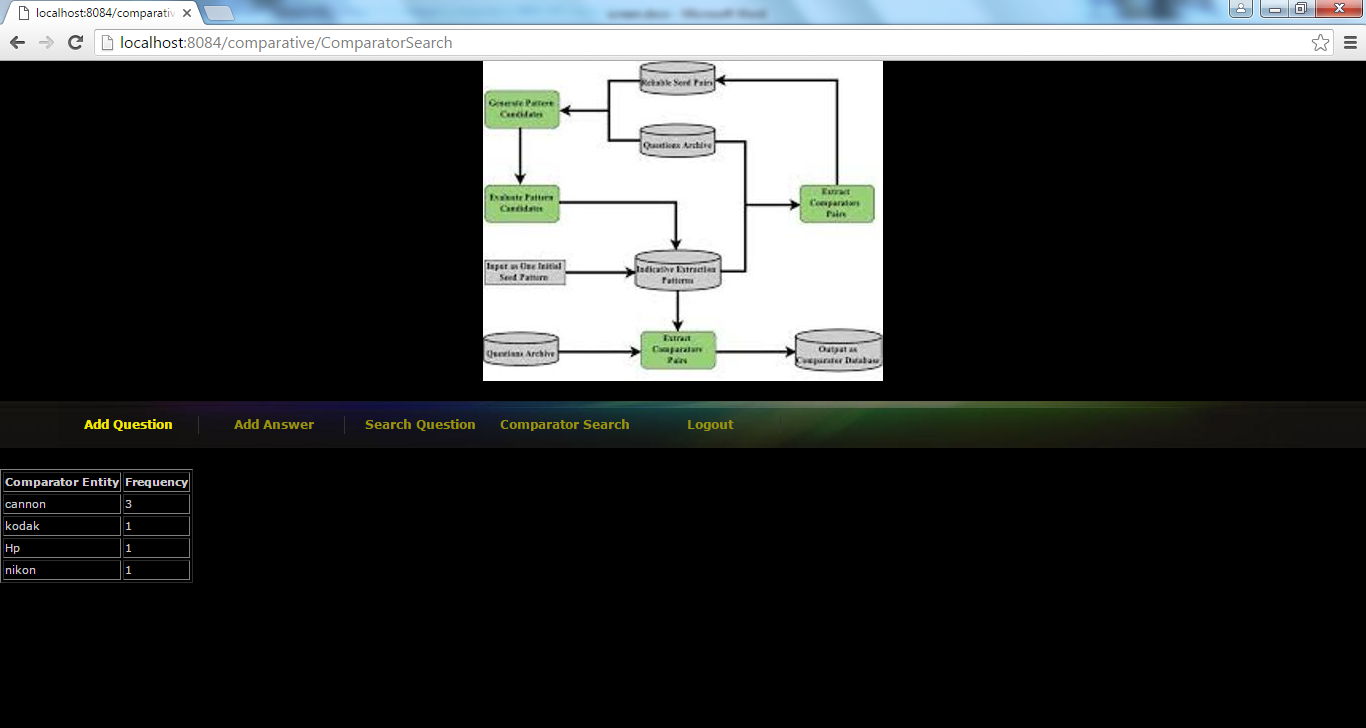
No answer for above queries



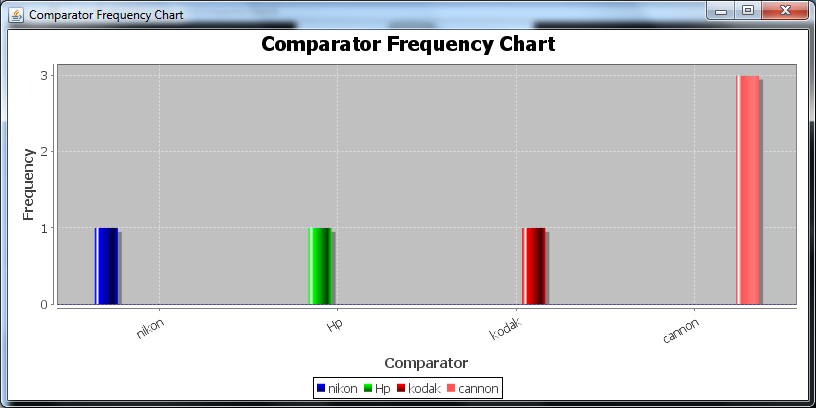
Comparator search screen



Answer for above comparator



Graph for above answer



**8. CONCLUSION**

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. 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.

**REFERENCES**

[1] M.E. Califf and R.J. Mooney, “Relational Learning of Pattern- Match Rules for Information Extraction,” Proc. 16th Nat’l Conf. Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence (AAAI ’99/IAAI ’99), 1999.

[2] C. Cardie, “Empirical Methods in Information Extraction,” Artificial Intelligence Magazine, vol. 18, pp. 65-79, 1997.

[3] D. Gusfield, Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge Univ. Press, 1997.

[4] T.H. Haveliwala, “Topic-Sensitive Pagerank,” Proc. 11th Int’l Conf. World Wide Web (WWW ’02), pp. 517-526, 2002.

[5] G. Jeh and J. Widom, “Scaling Personalized Web Search,” Proc. 12th Int’l Conf. World Wide Web (WWW ’02), pp. 271-279, 2003.

[6] 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.

[7] N. Jindal and B. Liu, “Mining Comparative Sentences and Relations,” Proc. 21st Nat’l Conf. Artificial Intelligence (AAAI ’06), 2006.

[8] Z. Kozareva, E. Riloff, and E. Hovy, “Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs,” Proc. Ann. Meeting of the Assoc. for Computational Linguistics: Human Language Technologies (ACL-08: HLT), pp. 1048-1056, 2008.

[9] 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.

[10] 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.