Structured data-based Q&A System using surface patterns

Nicolas Kuchmann-Beauger, Marie-Aude Aufaure

Structured data-based Q&A System using surface patterns

Nicolas Kuchmann-Beauger[1,2] and Marie-Aude Aufaure[2]

[1] SAP Research, 157/159 rue Anatole France, 92309 Levallois-Perret, France,
[2] Ecole Centrale Paris, MAS laboratory, 92290 Chatenay-Malabry, France,
{marie-aude.aufaure, nicolas.kuchmann-beauger}@ecp.fr

Abstract. Question Answering systems, unlike other Information Retrieval systems, aim at providing directly the answer to the user, and not a list of documents in which the correct answer may be found. Our system is based on Data Warehouses and provides composite answers made of data tables and corresponding chart visualizations for Business Intelligence purposes. The question translation step is based on a new proposal for surface patterns that incorporate business semantic as well as domain-specific knowledge allowing a better coverage of questions.

Keywords: Question Answering, Information Extraction, linguistic patterns

1 Introduction

Question Answering (Q&A) systems aim at providing one answer to a user's question. The major difference from other Information Retrieval (IR) systems is that the result is not a list of documents where the correct answer has to be found, but the answer itself [8]. Surveys on large-scale search services [10] which represent IR systems show that users struggle to formulate queries: the average query is being reformulated 2,6 times, but this average query is concise: it contains 3,5 tokens. The context of Q&A is quite different, because the user expects from the system the exact answer to appear and in this case keywords are not enough to express complex information needs. The fact that Q&A systems return concise answers and not whole pieces of documents, require from Q&A techniques a deeper understanding of document content [2]. Communities concerned by this field of research are Natural Language Processing (NLP), Machine Learning (ML), and more generally IR and Artificial Intelligence (AI).

Most common systems are based on unstructured data, especially web documents. Our context is quite different, because we focus on structured data in warehouses. Typical users are employees of a company who want to query and analyze those data; these users generally want to have a quick overview of the data, but do not always exactly know how to express such queries, because the syntax of the technical query (e.g. SQL or MDX) is not that easy to employ.

Questions of these users are data-oriented (the expected answer is a table of values and an associated visualization) whereas questions in traditional Q&A

are often factual questions or explanatory questions (the expected answer is a sentence or a phrase expressing the answer).

The answer that we consider is not a sentence as it is the case in most Q&A systems, but rather a composite answer. For example, the question "Detail the sales in the US and compare it with France" would return in our context tables of values, charts showing specific comparisons between data and possibly recommendation of reports composed of relevant queries, as opposed to a well-formed sentence (which would be the case in a traditional Q&A system based on unstructured data). Open questions, like "Why are we not going well?" is not the scope of our work.

Most people familiar with IR tools are used to express queries using keywords, because most popular IR systems like search engines are based on the assumption that queries are composed of keywords. However, there are several concerns about this interaction. Simple queries could be expressed using an ordered list of terms, but not complex ones. Table 1 illustrates examples of complex Business Intelligence (BI) queries. The proposed system allows as input queries expressed

Table 1. Examples of complex BI queries from different fields

Field	Question
Acquisition	Can I measure if my marketing campaign was effective?
Attendance	What effect has the campaign had on attendance?
Referral	How many of the referrals were existing customers?
Discount analysis	What is the correct price for my products?

in Natural Language, but it remains possible to interact with the system using traditional keywords.

We have addressed the core matter of answering BI questions, and existing proposals from the IR or Q&A communities do not satisfy these specific needs. In particular, we hope getting better recall on BI questions thanks to our new linguistic pattern definition that do not rely on the classic hypothesis of syntactic isomorphy (see section 3).

The rest of this paper is structured as follows. Section 2 presents the related work in this field. Section 3 deals with adopted definitions and choices for representing linguistic patterns. Section 4 presents the architecture of the system and give details of the implementation. The evaluation criteria and our experiments are discussed in section 5. The conclusion and future work are presented section 6.

2 Related Work

Q&A is one of the first applications of Artificial Intelligence. Its goal is to answer questions expressed in Natural Language (NL). First Q&A systems were

based on data structured in databases [4, 18] but the great majority of such systems look for answers in textual documents, because of the huge avaibility of unstructured documents, especially Web documents.

Q&A systems base different strategies to map users' questions to answers, whether extracted from text corpora or retrieved in databases. Andrenucci and Sneiders [1] address the main research approaches related to Q&A: NLP, IR and template-based approaches. One weakness in systems providing answers from databases such as [7] is that grammars linking questions to answers are highly database-dependant.

Pattern-based approaches (also called template-based approaches) are popular in this field, beacause they lead to good results (the TREC-10 winner used only a list of surface patterns as external resource). One issue is the representation of such patterns. Sung et al. [17] distinguish between patterns that do not represent any semantic and patterns that retrieve semantic relationships among terms (called semantic patterns). Question patterns are usually associated to answer patterns (that locate the answer in textual documents). In this case, fine-grained answer typing is important in order to get precise answers. Patterns defined by Soubbotin [16] are rich patterns composed of predefined string sequences as well as unordered combinations of strings and definition patterns. Much work has also been done in the pattern learning area: Saiz-Noeda et al. [13] propose a learning approach based on the maximum entropy, and apply this to anaphora resolution. Ravichandran et Hovy [11] propose a method to learn automatically new patterns for unstructured data-based Q&A, and propose to exploit answers from the Web as well (occurate answers are returned by Web search services in the top positions).

3 Linguistic patterns, definitions and hypothesis

Morpho-syntactic patterns are extensively used, but there are very few comments on the definition of such patterns. Such linguistic patterns have been defined in the linguistic theory [5] as "a schematic representation like a mathematical formula using terms or symbols to indicate categories that can be filled by specific morphemes". Patterns used by Sneiders [14] are regular strings of characters where sets of successive tokens are replaced by entity slots (to be filled by corresponding terms in the real text). An innovation presented in [15] lies on the definition of a pattern composed of two subpatterns: a required pattern (or regular pattern) and a forbidden pattern corresponding to a pattern that must not be matched. Finkelstein-Landau et Morin [3] define formally morpho-syntactic patterns related to their Information Extraction (IE) task: they aim at extracting semantic relationship from textual documents. The definition is displayed formula 1.

$$A = A_1 \dots A_i \dots A_j \dots A_n \tag{1}$$

In this formula, A_k $k \in \{1, n\}$ denotes an *item* of the pattern which is a part of a text (no constraint *a priori* on the sentence boundaries). An *item* is defined as

an ordered set of *tokens*, which composes words[3]. In this approach the syntactic isomorphy hypothesis is adopted.

$$B = B_1 \dots B'_i \dots B'_j \dots B'_n \tag{2}$$

This hypothesis states the following assertion:

$$\exists(i,j) \quad \left.\begin{array}{r} A \sim B \\ win(A_1,\dots,A_{i-1}) = win(B_1,\dots,B_{j-1}) \\ win(A_{i+1},\dots,A_{i+1}) = win(B_{j+1},\dots,B_{j+1}) \end{array}\right\} \Longrightarrow A_i \sim B_j \tag{3}$$

which means that if two patterns A and B are *equivalent* (they match the same text), and if it is possible to split both patterns in a set of equivalent subpatterns or *windows*, then the remaining items of both patterns (A_i and B_j) share the same syntactic function (they are equivalent).

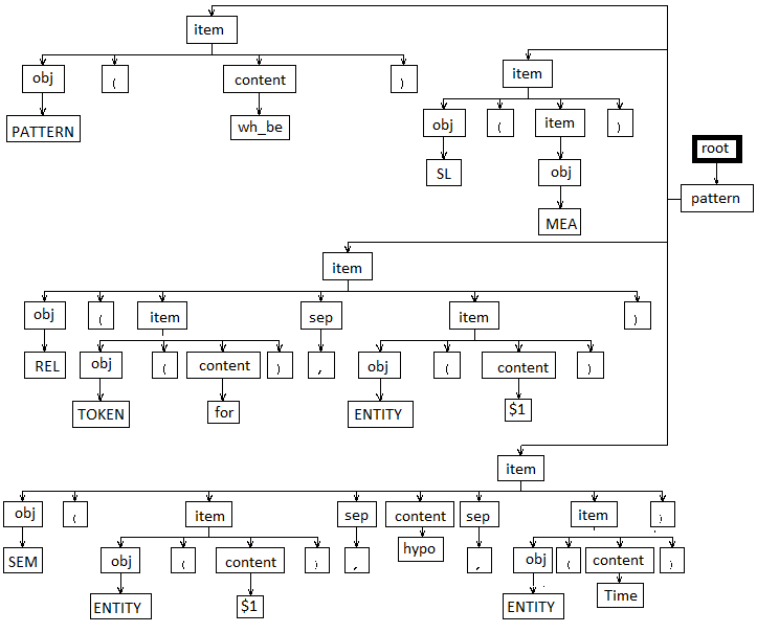

Fig. 1. Parse tree of the pattern corresponding to the question "What are the sales revenue in 2011?" generated by ANTLR [9]

We propose another formulation of patterns which are composite patterns, and we do not rely on the syntactic isomorphy. Moreover, we extend the classic notion of *surface patterns* (see for example [17]) which consists of pairs of question and answer syntactic pattern, where question patterns are patterns that

[3] Delimiting tokens is not an easy task in any language.

match users' questions and answer patterns are patterns that match sentences in the documents collection (in the case of structured data-based Q&A). In our work, we do not have any answer pattern but technical queries. The categories used to describe our patterns are tokens themselves (`TOKEN`), part-of-speech of tokens (`POS`), wh-question words (`WHQ`, widely used in our context) stems of tokens (`LEMMA`), terms related to a known concept in our domain ontology (`ENTITY`), objects defined in the data model of the underlying Data Warehouse (`SL(DIM)`, `SL(MEA)` or `SL(MEM)`), or references to existing patterns (`PATTERN`). We also allow the representation of the syntactic relation related to the syntactic hypothesis, which is the underlying syntagmatic order, and semantic relationship defined in the domain ontology. Another feature is the possibility to specify token references, which means that one token may be represented in the same pattern by more than one category, which is not possible in classical morpho-syntactic patterns. In addition, we also use the classical wildcards to specify cardinalities. We have defined a grammar to parse patterns.

The benefit of using our formulation is that the same pattern matches other formulations. If we consider the pattern `PATTERN(wh_be) SL(MEA) SL(MEM)`, possible matched questions could be "What are the sales revenue in 2011?", and "For 2011, what are the sales revenue?" as well. Figure 1 displays the parse tree of this pattern.

4 Architecture and implementation details

The proposed system interacts with a Data Warehouse (DW) through an abstraction layer (called Semantic Layer) on which queries are expressed regardless of the data connection. Technical queries are composed of objects from the layer, and aggregations, automatically computed, do not need to be expressed.

We present the architecture of the proposal in figure 2. We will focus on each component, Question Processing, Pattern Matching, Answer Processing and Answer Federation, but we will insist on the second one (Pattern Matching) which presents the main novelty of this paper.

4.1 Question Processing

The Question Processing component aims at analyzing the user's question. We use shallow NLP techniques to avoid time consuming processing. Our approach is based on linguistic patterns which compose the general-domain knowledge. Our assumption is that using a few patterns will be sufficient in most cases.

When a new question is submitted to the system and if the user hasn't specified the question language, it is analyzed. Then, the question is tokenized according to language rules defined in the SAP TextAnalysis language recognition tool. The NER identifies named entities in the user's question, including business entities. Additional knowledge is composed of a set of English question patterns that are matched against users' questions.

Technical queries that are associated to these initial patterns are used to produce the graphs representing the queries. These graphs are then used by the Answer Processing component to produce potential candidate answers. The last component of the system will be used to properly display the answers to the user (raw data and/or best associated visualization).

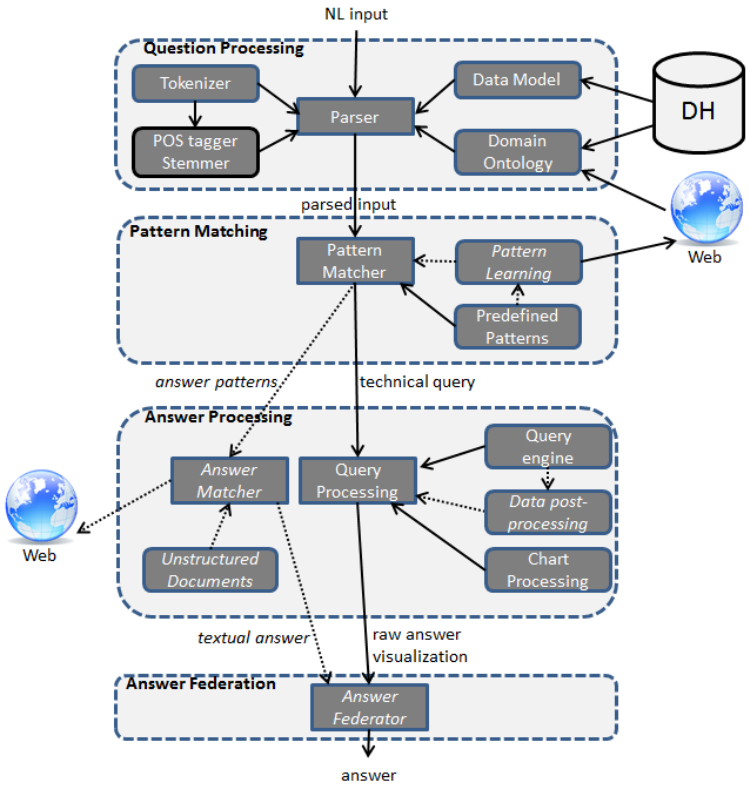

Fig. 2. Architecture of the proposal

4.2 Pattern Matching

This component analyzes the parsed user's input and retrieves similar patterns. The Pattern Learning approach is not fully implemented yet, but its goal is to build new patterns from users' parsed input when no similar existing pattern can be retrieved (this occurs when the most similar existing pattern presents a low similarity according to a custom threshold, in which case these similar patterns are not considered similar enough).

Consider the following pattern, which is part of the predefined patterns:

```
REL(WHQ, LEMMA(be)) SL(MEA) REL(TOKEN(in), ENTITY($1))
        SEM(ENTITY($1), hypo, ENTITY(Place))
```

One remarks that the order of the tokens in such patterns do not have any impact on the patterns themselves, the syntagmatic order being specified by the keyword `REL`. The keyword `SEM` indicates a semantic relationship defined in our domain ontology, and the identity constraint is specified by the keyword `$1` in this example. One associated initial question may be "What are the sales revenue in North America?". The parser produces a set of items, and the exact matching. Algorithm 1 instantiates the technical query associated to the pattern, and returns the set of associated answer patterns to be used for searching answers in unstructured documents. The subfunctions are explained below:

Algorithm 1 Exact pattern matching

```
var potentialQueries : Array = {}
for item ∈ userItems.getItems() do
  reachableItems ← item.getReachableItems()
  item.cardinality ← countSameItems(item, reachableItems)
end for
for pattern ∈ patterns do
  pattern.updatedReferences()
  var found : boolean = true
  for item ∈ pattern.getItems() do
    if ¬userItems.contains(item) then
      found ← false
    end if
  end for
  if found then
    potentialQueries.add(pattern.getQuery())
  end if
end for
```

- *getReachableItems* returns user items that appear *after* the considered item according to the position of each item (item position in the user's question) and the length of each item (the number of tokens that compose the item)
- *countSameItems* counts the number of identical items that appear after the considered item in the parsed user's question
- *updateReferences* replaces references of sub-patterns by the sub-patterns themselves, and links the items that make a reference to each other (which concerns items containing one `$i` ($i \in \mathbb{N}^+$) argument.
- *contains* is the matcher sub-function itself. It takes into account the type of the item, the name and arguments of the item (depending of the type of the item), the cardinality of both user and pattern items and the reference constraint if applicable

When no exact matching pattern is available, most similar patterns are considered and we made the assumption that the similarity measure should not consider that every token types are equivalent. We proposed the order displayed in table 2. This order corresponds to a weight in the similarity measure, that will lead to an evaluation to validate those weights.

Table 2. Weight order of token types when comparing patterns

Order	Token type
1	`SL(MEA)`
2	`SL(DIM)`
3	`SL(MEM)`
4	`REL`
5	`SEM`
6	`ENTITY`
7	`LEMMA`
8	`POS`
9	`TOKEN`

In our context, the most similar pattern selection can be seen as a maximization problem where we try to maximize the number of features (tokens) from the parsed user's question that also belong to the candidate pattern. We consider the following problem 4:

$$\begin{array}{ll} \max \sum_i w_{t(t_i)} & t_i \in t \subset \mathcal{T} \\ |t_i| < n & t_i \in t \\ \sum_i w_{t(t_i)} \leq 1 & \end{array} \tag{4}$$

where $t(t_i)$ denotes the type of the ith token in the candidate pattern, $w(k)$ the weight associated to the token type k, t the set of tokens that forms the candidate pattern, n the length of the user's question and $\mathcal{T}$ the set of possible tokens in patterns.

This allows us to match user's question to predefined patterns, even if exact matching is not possible. Moreover, this formulation seems more accurate than a classic similarity measure based on the distance between tokens of the potential pattern and the tokens of the predefined patterns, because in our context we do not want to rank patterns, we aim at selecting one most similar pattern. Another explanation for this choice, is that such measures as described in [12] rely on the edit distance measure, which is based on the assumption that the considered linguistic patterns share the syntagmatic order, which is not our hypothesis.

4.3 Answer Processing

Figure 3 is an example of the answer provided by the prototype that corresponds to the question "What are the sales revenue in New York and in Texas?". The

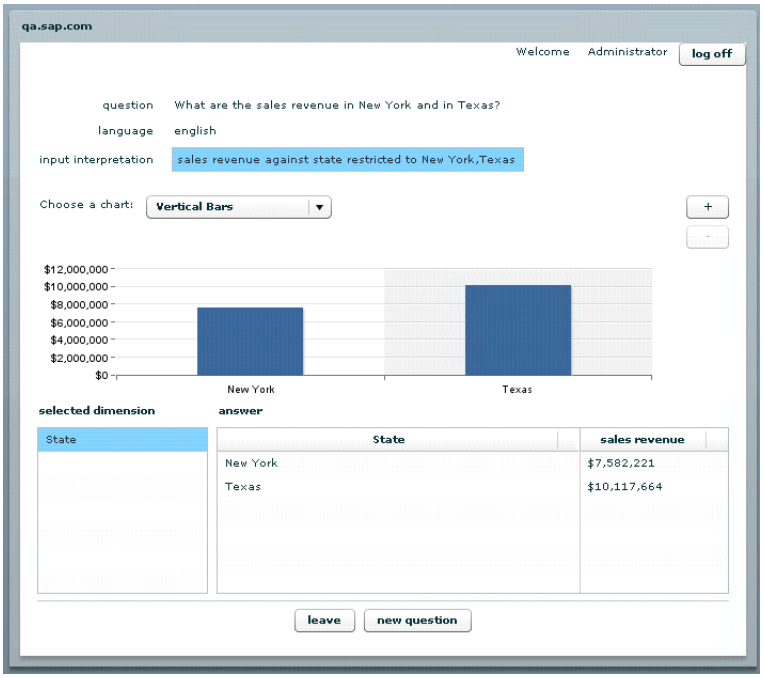

Fig. 3. Answer provided by the prototype

language is automatically identified (*English*) and the answer is composed of the raw values (a table) and the visualization of the answer. The "input interpretation" corresponds to the linguistic pattern that has been selected from the user's input.

Two components are at the moment fully implemented: the incorporation of the results of the query engine, and the chart processing. The former consists in invoking a query service using objects defined in the data model (from the data abstract layer), and interpreting the XML-output, and the latter returns a vector or binary image from the raw data.

The data post-processing has been partly implemented, but we believe this feature is vital in the context of structured-based Q&A.

4.4 Answer Federation

The Answer Federation component merges answers from different sources: answer from the Data Warehouse, and the answer from other unstructured or semi-structured documents. As an example, BI reports are analyzed to identify content relevant to the user's query, and the provided information is compared to the answer.

If relevant, those reports are then suggested in a recommender approach: users may be interested in navigating documents (containing more general in-

formation than the answer to the user's initial query), but our approach is to propose those documents in the end, as a complementary source of information. We propose the exact answer first, and encourage then users to explore the data and the information to satisfy better the information need.

5 Evaluation criteria and experiments

Q&A systems have been studied for decades, and numerous evaluation scenarios have been proposed. We will discuss the scenarios applicable to our system on the one hand, and our experiments results on the second hand.

5.1 Evaluation criteria and scenarios

When evaluating a whole system, different scenarios are possible:

1. evaluation of the system globally (*black box evaluation*): how does the system globally perform compared to an assumed ground truth?
2. evaluation of each sub-component (*white box evaluation*): from one input specifications and output specification, how well does the component proceed?

In the real life, things are not that simple. One huge restriction, is the lack of comparable systems: one should compare the system with other systems based on structured data and dedicated to BI questions; however, to the best of our knowledge, there is no competition comparable to TREC for open-domain questions for example.

This leads to two forms of evaluation, that sound applicable in our context:

- evaluation of the users' satisfaction
- evaluation of each component

Evaluation of users' satisfaction may be performed by directly asking to the user, but feedback from user experience show that users are not willing to waste time giving their opinion on the usability of the systems they have used. Another option consists in analyzing users' interaction with the system, which is an entire research area [6] and not the scope of the present paper.

5.2 Experiments

In order to mimit a real use of the system, we selected randomly 100 BI questions written by experts and linked with the DW, which contains data about sales of clothes in different stores.

The results are displayed table 3. The first line "no answer" corresponds to very complex questions that cannot be answered yet, because we do not reach the required analysis level for these questions. Example of such questions are: "Which products have the lartgest sales changes since last period?" or "What

Table 3. Results of the experiment

Kind	Result
No answer	6%
Already existing pattern	20%
New pattern defined	74%

is the total revenue change attributable to the 10 biggest revenue growers and decliners between 2004 and 2005?". The second category is made of questions that did not require any new pattern definition. The last category is composed of the remaining of questions, that required the definition of new patterns. One assumption presented in section 3 states that if several patterns are applicable to one question, the longest pattern is taken into consideration according to the weight order depending on the item types (see table 2). The situation where this assumption lead to a wrong question analysis has never been met in our experiment.

6 Conclusion and future work

We have implemented a Q&A system able to answer BI-questions expressed in NL or using keywords on data warehouses. The original proposal on pattern formulation leads to a better coverage of users' questions. The system does not need any setup effort. Shallow linguistic techniques that we use allow us to get a better understanding of the users' need.

We believe one major improvement will be the ability to handle unstructured and semi-structured documents, such as documents present in enterprise intranets, or documents located in users' repositories, such as BI reports.

The approach we are willing to adopt, is case-based reasoning in the context of pattern learning. This approach will learn automatically new linguistic patterns from users' input. Taking into account the context is also a major topic in our work; the considered context is the user-centered context and the global preferences and security roles that will be defined. The follow-up questions feature may also improve our results, since users may want to refer to previous questions, and because constraints on follow-up questions may be defined. Another interesting improvement will be the generation of a textual summary of the answer using the domain ontology.

References

1. Andrenucci, A., Sneiders, E.: Automated question answering: Review of the main approaches. In: Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2 - Volume 02. pp. 514–519. ICITA '05, IEEE Computer Society, Washington, DC, USA (2005)

2. Ferrández, A., Peral, J.: The benefits of the interaction between data warehouses and question answering. In: Daniel, F., Delcambre, L.M.L., Fotouhi, F., Garrigós, I., Guerrini, G., Mazón, J.N., Mesiti, M., Müller-Feuerstein, S., Trujillo, J., Truta, T.M., Volz, B., Waller, E., Xiong, L., Zimányi, E. (eds.) EDBT/ICDT Workshops. ACM International Conference Proceeding Series, ACM (2010)
3. Finkelstein-landau, M., Morin, E.: Extracting semantic relationships between terms: Supervised vs. unsupervised methods. In: In Proceedings of International Workshop on Ontological Engineering on the Global Information Infrastructure. pp. 71–80 (1999)
4. Green, Jr., B.F., Wolf, A.K., Chomsky, C., Laughery, K.: Baseball: an automatic question-answerer. In: Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference. pp. 219–224. IRE-AIEE-ACM '61 (Western), ACM, New York, NY, USA (1961)
5. Hayes, B., Curtiss, S., Szabolcsi, A., Stowell, T., Stabler, E., Sportiche, D., Koopman, H., Keating, P., Munro, P., Hyams, N., Steriade, D.: Linguistics: An Introduction to Linguistic Theory. Wiley-Blackwell (Feb 2001)
6. Joachims, T.: Optimizing search engines using clickthrough data. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD). pp. 133–142 (2002)
7. Li, Y., Yang, H., Jagadish, H.V.: Nalix: an interactive natural language interface for querying xml. In: SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. pp. 900–902. ACM, New York, NY, USA (2005)
8. Moldovan, D.I., Surdeanu, M.: On the role of information retrieval and information extraction in question answering systems. In: SCIE. pp. 129–147 (2002)
9. Parr, T.J., Quong, R.W.: Antlr: A predicated- *ll(k)* parser generator. Softw., Pract. Exper. 25(7), 789–810 (1995)
10. Pass, G., Chowdhury, A., Torgeson, C.: A picture of search. In: Jia, X. (ed.) Infoscale. ACM International Conference Proceeding Series, vol. 152, p. 1. ACM (2006)
11. Ravichandran, D., Hovy, E.H.: Learning surface text patterns for a question answering system. In: ACL. pp. 41–47 (2002)
12. Ruiz-Casado, M., Alfonseca, E., Castells, P.: Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia. Data Knowl. Eng. 61, 484–499 (June 2007)
13. Saiz-Noeda, M., Suárez, A., Palomar, M.: Semantic pattern learning through maximum entropy-based wsd technique. In: Proceedings of the 2001 workshop on Computational Natural Language Learning - Volume 7. ConLL '01, Association for Computational Linguistics, Stroudsburg, PA, USA (2001)
14. Sneiders, E.: Automated Question Answering: Template-based Approach. Ph.D. thesis, Royal Institute of Technology, Sweden (2002)
15. Sneiders, E.: Automated email answering by text pattern matching. In: Loftsson, H., Rögnvaldsson, E., Helgadóttir, S. (eds.) IceTAL. Lecture Notes in Computer Science, vol. 6233, pp. 381–392. Springer (2010)
16. Soubbotin, M.M.: Patterns of potential answer expressions as clues to the right answers. In: TREC (2001)
17. Sung, C.L., Lee, C.W., Yen, H.C., Hsu, W.L.: An alignment-based surface pattern for a question answering system. In: IRI. pp. 172–177. IEEE Systems, Man, and Cybernetics Society (2008)
18. Woods, W.A.: Progress in natural language understanding: an application to lunar geology. In: AFIPS '73: Proceedings of the June 4-8, 1973, national computer conference and exposition. pp. 441–450. ACM, New York, NY, USA (1973)