International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference on Industrial Automation And Computing (ICIAC- 12th & 13th April 2014)

RESEARCH ARTICLE OPEN ACCESS

Knowledge Acquisition using Tree Based Machine Learning Model: A Survey

Satvik Satish Bahaley [1], Dr. M. A. Pund [2]
[1] PG Student, Department of Computer Science and Engineering, Prof. Ram Meghe Institute of Technology and Research, Badnera, Maharashtra, India.
thesatvik@gmail.com
[2] Professor, Department of Computer Science and Engineering, Prof. Ram Meghe Institute of Technology and Research, Badnera, Maharashtra, India.
pundmukesh@gmail.com

ABSTRACT
Knowledge Acquisition is a well-known and upcoming field of Computer Science. The similar concept of Information Retrieval is mature and has many concepts which help in the Knowledge Acquisition procedure. Information Retrieval is responsible for simply extraction of raw data in an organized manner (tabular form) which we call Information. Information alone is not good enough when Humans want to interact with Machines and hence the need is to translate this retrieved information into a human understandable, readable and grammatically correct Natural Language, e.g. English. Knowledge Acquisition in itself has potential to enhance the Natural Language Processing, to a more mature and concrete level.
Keywords - Knowledge Acquisition, A Priori Algorithm, Tree Based Machine Learning, Decision Trees and Natural Language Representation.

I. INTRODUCTION

Knowledge Acquisition alone is not as good as when we combine it with Machine Learning, especially the Tree Based Machine Learning [17]. The idea to use Tree Based Machine Learning is derived from the Decision Trees [23], [26] and Connectionist Approach. The Tree structure is excellent to segregate, classify and do hierarchical extraction of information and hence knowledge. When the system has to analyze multiple text documents, the Tree structure will help us to classify these documents on the basis of relevance of information these documents carry. As we extract this information and form higher levels of tree nodes we get more concentrated information, which is void of redundancies and is more specific than level below, so on till the root, which will have supposedly the most precise knowledge extracted from available input of documents. The Machine Learning in Trees will help the system classify better with every iteration. Slowly the system will learn at each tree level and increase its precision and recall of the information and hence the knowledge. The learning procedure here is of dynamic nature and somewhat real time because the system is improving at each level of tree. This approach is rather different from the contemporary tree based learning but uses trees and hence the name. Adding these above concepts together will make the system precision higher than its subsystems individually can achieve. The paradigms of Tree Structures, Information Retrieval [1], [2], [3], Knowledge Acquisition [14], [15] and Natural Language Processing together are powerful enough to give very natural and precise results.

II. LITERATURE REVIEW

1. Knowledge Acquisition

Knowledge acquisition [19] is a method of learning, first proposed by Aristotle in his seminal work "Organon". Aristotle proposed that the mind at birth is a blank slate, or tabula rasa. As a blank slate it contains no knowledge of the objective, empirical universe, nor of itself. As a method, it is opposed to the concept of "a priori" knowledge, and to "intuition" when conceived as religious revelation.

Knowledge acquisition is the process of extracting, structuring and organizing knowledge from one source, usually human experts, so it can be used in software such as an ES. This is often the major obstacle in building an ES.

There are three main topic areas central to knowledge acquisition that requires consideration in all ES projects. First, the domain must be evaluated to determine if the type of knowledge in the domain is suitable for an ES. Second, the source of expertise must be identified and evaluated to ensure that the specific level of knowledge required by the project is provided. Third, if the major source of expertise is a person, the specific knowledge acquisition techniques and participants need to be identified.

Knowledge Acquisition Technique

At the heart of the process is the interview. The heuristic model of the domain is usually extracted through a series of intense, systematic interviews, usually extending over a period of many months [27]. Note that this assumes the expert and the knowledge engineer are not the same person. It is generally best that the expert and the knowledge engineer not be the same person since the deeper the experts' knowledge, the less able they are in describing their logic. Furthermore, in their efforts to describe their procedures, experts tend to rationalize their knowledge and this can be misleading.

General suggestions about the knowledge acquisition process are summarized in rough chronological order below:

- Observe the person solving real problems.
- Through discussions, identify the kinds of data, knowledge and procedures required to solve different types of problems.
- Build scenarios with the expert that can be associated with different problem types.
- Have the expert solve a series of problems verbally and ask the rationale behind each step.
- Develop rules based on the interviews and solve the problems with them.
- Have the expert review the rules and the general problem solving procedure.
- Compare the responses of outside experts to a set of scenarios obtained from the project's expert and the ES.

Note that most of these procedures require a close working relationship between the knowledge engineer and the expert.

Practical Considerations

The preceding section provided an idealized version of how ES projects might be conducted. In most instances, the above suggestions are considered and modified to suit the particular project. The remainder of this section will describe a range of knowledge acquisition techniques that have been successfully used in the development of ES.

Operational Goals

After an evaluation of the problem domain shows that an ES solution is appropriate and feasible, then realistic goals for the project can be formulated. An ES's operational goals should define exactly what level of expertise its final product should be able to deliver, who the expected user is and how the product is to be delivered. If participants do not have a shared concept of the project's operational goals, knowledge acquisition is hampered.

Pre-training

Pre-training the knowledge engineer about the domain can be important. In the past, knowledge engineers have often been unfamiliar with the domain. As a result, the development process was greatly hindered. If a knowledge engineer has limited knowledge of the problem domain, then pre-training in the domain is very important and can significantly boost the early development of the ES.

Knowledge Document

Once development begins on the knowledge base, the process should be well documented. In addition to tutorial a document, a knowledge document [48] that succinctly states the project's current knowledge base should be kept. Conventions should be established for the document such as keeping the rules in quasi-English format, using standard domain jargon, giving descriptive names to the rules and including supplementary, explanatory clauses with each rule. The rules should be grouped into natural subdivisions and the entire document should be kept current.

Questionnaires

When specific information is needed, a questionnaire can sometimes be used effectively. Questionnaires are generally used in combination with other techniques such as interviews.

2. Decision Trees

Decision trees [17], [26] are widely recognized to be useful tools for the knowledge engineer in prototyping knowledge representations. In addition, they can be useful in knowledge acquisition on several different levels. Some knowledge engineers have found that experts can more readily relate to decision trees than rules.

Rule Development

Although complex representation techniques might eventually be used, rules are generally easier to use for characterizing knowledge during knowledge acquisition. Prototypic rules should be developed as soon as possible to serve as a focal point for directing the course of the knowledge acquisition process. The initial knowledge base can be developed from written materials or from example cases described by the expert during early unstructured interviews. Initial rules should be treated as approximations and their wording should be general to avoid pressuring the expert. As additional cases are described during interviews, the rule base can be expanded. Once a stable rule base begins to develop, it can provide feedback for structuring interviews. Initially the rules and procedures can be traced through by hand with the expert considering each step. The same pattern of tracing through rules should continue once a version of the knowledge base is developed on a computer and it frequent use should become part of the process.

Recognition of the central role of knowledge acquisition in the development of ES is an unavoidable prerequisite for any knowledge engineering project. If domains are defined and experts chosen with this in mind, a project's chances for success will be greatly increased. Once an appropriate domain is identified and a cooperative, available expert with the necessary stamina is found, then the practical approaches to knowledge acquisition outlined should be of help.

There are several points that deserve to be emphasized:

- Projects need to be well planned. The knowledge engineer has the responsibility initially to define explicit operational goals that are consistent with the resources available to a project. Early in the interview process, the purpose of the project and the roles of the participants should be carefully discussed. The discussion should lead to a consensus opinion on what is expected in a final product, who its users should be and how it should be delivered. As the project develops, the operational goals should be consciously reconsidered on a regular basis.
- The knowledge acquisition process should be carefully planned to match the requirements of the project's domain expert. For example, time lines that allow for the necessary pre-training, unstructured interviewing and the structured interview phases can be developed. Documentation procedures to be used during the project should be defined.
- Specific interviews or knowledge gathering events should be planned to accomplish specific goals. In pre-training, the goal might be to identify several realistic scenarios and during the first unstructured interviews one goal might be to develop a glossary of the expert's terminology. Later, the goals might be to obtain specific bits of information to explain apparent discrepancies. In planning individual interviews, the knowledge engineer should try to get explicit feedback.

Regardless of its size or the intended application, the knowledge acquisition process cannot be avoided. Recognition of this is the first step toward the successful development of a functional ES.

3. Information Retrieval

Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text (or other content-based) indexing.

Efficiency in dealing with information retrieval (IR) [5], [6], [9] tools is related to the consideration of relevant semantic data describing terms and concepts in the specific do-main considered. This kind of resources are often taken from an external and generic module (Aussenac-Gilles and Mothe,2004),which implies that we probably lose a number of interesting properties we would be able to recover if semantic processing was directly performed on the text collection we are dealing with.

In a full-text search, a search engine [10] examines all of the words in every stored document as it tries to match search criteria

Improved querying tools

- Keywords: Document creators (or trained indexers) are asked to supply a list of words that describe the subject of the text, including synonyms of words that describe this subject. Keywords improve recall, particularly if the keyword list includes a search word that is not in the document text.
- Field-restricted search: Some search engines enable users to limit free text searches to a particular field within a stored data record, such as "Title" or "Author."
- Boolean queries: Searches that use Boolean operators (for example, "encyclopedia" AND "online" NOT "Encarta") can dramatically increase the precision of a free text search. The AND operator says, in effect, "Do not retrieve any document unless it contains both of these terms." The NOT operator says, in effect, "Do not retrieve any document that contains this word." If the retrieval list retrieves too few documents, the OR operator can be used to increase recall; consider, for example, "encyclopedia" AND "online" OR "Internet" NOT "Encarta". This search will retrieve documents about online encyclopedias that use the term "Internet" instead of "online." This increase in precision is very commonly counter-productive since it usually comes with a dramatic loss of recall.
- Phrase search: A phrase search matches only those documents that contain a specified phrase, such as "Wikipedia, the free encyclopedia."
- Concept search: A search that is based on multi-word concepts, for example Compound term processing. This type of search is becoming popular in many e-Discovery solutions.
- Concordance search: A concordance search produces an alphabetical list of all principal words that occur in a text with their immediate context.

- Proximity search: A phrase search matches only those documents that contain two or more words that are separated by a specified number of words; a search for "Wikipedia" WITHIN2 "free" would retrieve only those documents in which the words "Wikipedia" and "free" occur within two words of each other.
- Regular expression: A regular expression employs a complex but powerful querying syntax that can be used to specify retrieval conditions with precision.
- Fuzzy search will search for document that match the given terms and some variation around them (using for instance edit distance to threshold the multiple variation)
- Wildcard search: A search that substitutes one or more characters in a search query for a wildcard character such as an asterisk. For example using the asterisk in a search query "s*n" will find "sin", "son", "sun", etc. in a text.

4. Natural Language Representation

The process to generate text can be as simple as keeping a list of canned text that is copied and pasted, possibly linked with some glue text. The results may be satisfactory in simple domains such as horoscope machines or generators of personalized business letters. However, a sophisticated NLG [21], [22] system needs to include stages of planning and merging of information to enable the generation of text that looks natural and does not become repetitive. The typical stages of natural language generation, as proposed by Dale and Reiter, are:

Content determination: Deciding what information to mention in the text. For instance, in the pollen example above, deciding whether to explicitly mention that pollen level is 7 in the south east.

Document structuring: It is overall organization of the information to convey. For example, deciding to describe the areas with high pollen levels first, instead of the areas with low pollen levels.

Aggregation: Merging of similar sentences to improve readability and naturalness. For instance, merging the two sentences Grass pollen levels for Friday have increased from the moderate to high levels of yesterday and Grass pollen levels will be around 6 to 7 across most parts of the country into the single sentence Grass pollen levels for Friday have increased from the moderate to high levels of yesterday with values of around 6 to 7 across most parts of the country.

Lexical choice: Putting words to the concepts. For example, deciding whether medium or moderate should be used when describing a pollen level of 4.

Referring expression generation: Creating referring expressions that identify objects and regions. For example, deciding to use in the Northern Isles and far northeast of mainland Scotland to refer to a certain region in Scotland. This task also includes making decisions about pronouns and other types of anaphora.

Realization: The actual text, which should be correct according to the rules of syntax, morphology, and orthography. For example, using will be for the future tense of to be.

5. Tree Based Machine Learning

A decision tree is a simple representation for classifying examples. Decision tree learning is one of the most successful techniques for supervised classification learning. For this section, assume that all of the features have finite discrete domains, and there is a single target feature called the classification. Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees (TDIDT) is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

Decision tree advantages

Amongst other data mining methods, decision trees have various advantages:

- *Simple to understand and interpret.* People are able to understand decision tree models after a brief explanation.
- *Requires little data preparation.* Other techniques often require data normalisation, dummy variables need to be created and blank values to be removed.
- *Able to handle both numerical and categorical data.* Other techniques are usually specialised in analysing datasets that have only one type of variable. (For example, relation rules can be used only with nominal

variables while neural networks can be used only with numerical variables.)

- *Uses a white box model.* If a given situation is observable in a model the explanation for the condition is easily explained by boolean logic. (An example of a black box model is an artificial neural network since the explanation for the results is difficult to understand.)
- *Possible to validate a model using statistical tests.* That makes it possible to account for the reliability of the model.
- *Robust.* Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.
- *Performs well with large datasets.* Large amounts of data can be analysed using standard computing resources in reasonable time.

6. Concept Mining

Concept mining [7], [8] is an activity that results in the extraction of concepts from artifacts. Solutions to the task typically involve aspects of artificial intelligence and statistics, such as data mining and text mining. Because artifacts are typically a loosely structured sequence of words and other symbols (rather than concepts), the problem is nontrivial, but it can provide powerful insights into the meaning, provenance and similarity of documents. Traditionally, the conversion of words to concepts has been performed using a thesaurus, and for computational techniques the tendency is to do the same. The thesauri used are either specially created for the task, or a pre-existing language model, usually related to Princeton's WordNet. The mappings of words to concepts are often ambiguous. Typically each word in a given language will relate to several possible concepts. Humans use context to disambiguate the various meanings of a given piece of text, where available machine translation systems cannot easily infer context. For the purposes of concept mining however, these ambiguities tend to be less important than they are with machine translation, for in large documents the ambiguities tend to even out, much as is the case with text mining. There are many techniques for disambiguation that may be used. Examples are linguistic analysis of the text and the use of word and concept association frequency information that may be inferred from large text corpora. Recently, techniques that base on semantic similarity between the possible concepts and the context have appeared and gained interest in the scientific community.

7. Association Rule Learning

Association Rule Learning [43], [44], [52] is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, Rakesh Agrawal et al [51], introduced association rules for discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule $\{\mathrm{onions}, \mathrm{potatoes}\} \Rightarrow \{\mathrm{burger}\}$ found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy hamburger meat. Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection, Continuous production, and bioinformatics. As opposed to sequence mining, association rule learning typically does not consider the order of items either within a transaction or across transactions.

Following the original definition by Agrawal et al. the problem of association rule mining is defined as: Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of n binary attributes called items. Let $D = \{t_1, t_2, \ldots, t_m\}$ be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent(right-hand-side or RHS) of the rule respectively.

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is $I = \{\mathrm{milk}, \mathrm{bread}, \mathrm{butter}, \mathrm{beer}\}$ an d a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table to the right. An example rule for the supermarket could be $\{\mathrm{butter}, \mathrm{bread}\} \Rightarrow \{\mathrm{milk}\}$ meaning that if butter and bread are bought, customers also buy milk.

Note: this example is extremely small. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

TABLE I
Example of Association Rule Learning.

Transaction ID	Milk	Bread	Butter	Beer
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

8. Apriori Algorithm

Apriori [30], [31], [32] is a classic algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing). Each transaction is seen as a set of items (an itemset). Given a threshold C, the Apriori algorithm identifies the item sets which are subsets of at least C transactions in the database.

Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found.

Apriori uses breadth-first search and a Hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length $k-1$. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent k-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

The pseudo code for the algorithm is given below for a transaction database T, and a support threshold of ϵ. Usual set theoretic notation is employed, though note that T is a multiset. C_k is the candidate set for level k. Generate() algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. $count[c]$ accesses a field of the data structure that represents candidate set C, which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets, and counting their frequencies.

$$
\begin{array}{l}
\mathrm{Apriori}(T,\epsilon)\\
\quad L_1 \leftarrow \{\text{large } 1-\text{itemsets}\}\\
\quad k \leftarrow 2\\
\qquad \textbf{while } L_{k-1} \neq \ \mathit{emptyset}\\
\qquad\quad C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \in \bigcup L_{k-1} \wedge b \notin a\}\\
\qquad\quad \textbf{for} \text{ transactions } t \in T\\
\qquad\qquad C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}\\
\qquad\qquad \textbf{for} \text{ candidates } c \in C_t\\
\qquad\qquad\quad \mathit{count}[c] \leftarrow \mathit{count}[c] + 1\\
\qquad\quad L_k \leftarrow \{c \mid c \in C_k \wedge \ \mathit{count}[c] \geq \epsilon\}\\
\qquad\quad k \leftarrow k+1\\
\qquad \textbf{return } \bigcup_k L_k
\end{array}
$$

III. Pitfall of Existing approaches/ techniques/ methods

The most widely used technique in Association Mining is Apriori Algorithm which retrieves the similar information based on association. This information is mostly raw and unorganized and hence the proposed system plans to retrieve the information but in more generalized and human readable form. The proposed system should extract the common information which is of importance to general populace and not to experts.

Limitations of Apriori Algorithm

Apriori, while historically significant, suffers from a number of inefficiencies or trade-offs, which have spawned other algorithms. Candidate generation generates large numbers of subsets (the algorithm attempts to load up the candidate set with as many as possible before each scan). Bottom-up subset exploration (essentially a breadth-first traversal of the subset lattice) finds any maximal subset S only after all $2^{|S|}-1$ of its proper subsets.

Later algorithms such as Max-Miner try to identify the maximal frequent item sets without enumerating their subsets, and perform "jumps" in the search space rather than a purely bottom-up approach.

There is no existing system which uses Information Retrieval and Data Mining concepts such as Apriori Algorithm for the purpose of Knowledge Acquisition. These techniques are popular in simple Data Mining but are not used in conjunction with Natural Language Representation for the use of layman. The Apriori Algorithm is a tree based algorithm which is fundamentally a type of Decision Tree. Using Apriori for Information Retrieval is a difficult task and is yet to be used together on Natural Language such as English.

IV. Conclusion

The above mentioned concepts can be collaboratively used to get the desired system functioning. The concepts should not be used as they are and only the core fundamentals are to be implemented in some cases. The concept of Knowledge Acquisition is relatively broad and is mixed implementation of other concepts from Information Retrieval using Natural Language Processing paradigms, also the Tree Based Machine Learning fundamentals are used for Classification purposes and segregation of document on the Information criteria. Concept Mining is an advanced implementation of Information Retrieval and the tasks of Concept Mining are also applied to extract the information from documents.

Apriori Algorithm makes a very good base algorithm for the task of Information Retrieval and thus Knowledge Acquisition. An effort is being made on the same lines to come up with a suitable modification to Apriori Algorithm to be implemented for Knowledge Acquisition purposes.

REFERENCES

[1] Frakes, William B. (1992). Information Retrieval Data Structures & Algorithms. Prentice-Hall, Inc. ISBN 0-13-463837-9.
[2] Singhal, Amit (2001). "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35–43.
[3] Zhu, Mu (2004). "Recall, Precision and Average Precision".
[4] Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich (2008). Introduction to Information Retrieval. Cambridge University Press.
[5] Svenonius, Elaine (2000). The Intellectual Foundation of Information Organization (1 ed.). The MIT Press. ISBN 0-262-19433-3.
[6] R. Navigli, Word Sense Disambiguation: A Survey, ACM Computing Surveys, 41(2), 2009.
[7] Fausto Giunchiglia, Uladzimir Kharkevich, and Ilya Zaihrayeu. Concept Search, In Proceedings of European Semantic Web Conference, 2009.
[8] Laplanche, R., Delgado, J., Turck, M., Concept Search Technology Goes Beyond Keywords, Information Outlook, July 2004.
[9] Robertson, S. E., Sparck Jones, K., Simple, Proven Approaches to Text Retrieval, Technical Report, University of Cambridge Computer Laboratory, December 1994.
[10] Croft, B., Metzler, D., Strohman, T., Search Engines, Information Retrieval in Practice, Addison Wesley, 2009.
[11] Baeza-Yates R, Navarro G. "Fast Approximate String Matching in a Dictionary". Proc. SPIRE'98. IEEE CS Press. pp. 14–22.
[12] Galil, Zvi; Apostolico, Alberto (1997). Pattern matching algorithms. Oxford [Oxfordshire]: Oxford University Press. ISBN 0-19-511367-5.
[13] Yuen-Hsien Tseng, " Generic Title Labeling for Clustered Documents", Expert Systems With Applications, Vol. 37, No. 3, 15 March 2010, pp. 2247-2254 .
[14] Dale, Robert; Reiter, Ehud (2000). Building natural language generation systems. Cambridge, U.K.: Cambridge University Press. ISBN ISBN 0-521-02451-X
[15] Portet F, Reiter E, Gatt A, Hunter J, Sripada S, Freer Y, Sykes C (2009). "Automatic Generation of Textual Summaries from Neonatal Intensive Care Data". Artificial Intelligence 173 (7–8): 789–816.
[16] Galitsky, Boris (2013). "A Web Mining Tool for Assistance with Creative Writing". Advances in Information Retrieval. Lecture Notes in Computer Science 7814: 828–831.
[17] Galitsky B, de la Rosa JL, Dobrocsi G. (2012). "Inferring the semantic properties of sentences by mining syntactic parse trees". Data & Knowledge Engineering. 81-82: 21–45.
[18] Sauper, Christina and Barzilay, Regina (2009). "Automatically Generating Wikipedia Articles: A Structure-Aware Approach". Proceedings of ACL.
[19] Mining the Web: discovering knowledge from hypertext data by Soumen Chakrabarti 2002 ISBN 1-55860-754-4 page 289
[20] Using computers in linguistics: a practical guide by John Lawler, Helen Aristar Dry 198 ISBN 0-415-16792-2 page 209
[21] Pragmatics and natural language understanding by Georgia M. Green 1996 ISBN 0-8058-2166-X

[22] Understanding language understanding by Ashwin Ram, Kenneth Moorman 1999 ISBN 0-262-18192-4 page 111
[23] Rokach, Lior; Maimon, O. (2008). Data mining with decision trees: theory and applications. World Scientific Pub Co Inc. ISBN 978-9812771711.
[24] Breiman, Leo; Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software. ISBN 978-0-412-04841-8.
[25] Rokach, L.; Maimon, O. (2005). "Top-down induction of decision trees classifiers-a survey". IEEE Transactions on Systems, Man, and Cybernetics, Part C 35 (4): 476–487. doi:10.1109/TSMCC.2004.843247
[26] Murthy S. (1998). Automatic construction of decision trees from data: A multidisciplinary survey. Data Mining and Knowledge Discovery
[27] http://en.wikipedia.org/wiki/Knowledge_Acquisition
[28] Piatetsky-Shapiro, Gregory (1991), Discovery, analysis, and presentation of strong rules, in Piatetsky-Shapiro, Gregory; and Frawley, William J.; eds., Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA.
[29] Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207.
[30] Hipp, J.; Güntzer, U.; Nakhaeizadeh, G. (2000). "Algorithms for association rule mining --- a general survey and comparison". ACM SIGKDD Explorations Newsletter 2: 58.
[31] Tan, Pang-Ning; Michael, Steinbach; Kumar, Vipin (2005). "Chapter 6. Association Analysis: Basic Concepts and Algorithms". Introduction to Data Mining. Addison-Wesley. ISBN 0-321-32136-7.
[32] Pei, Jian; Han, Jiawei; and Lakshmanan, Laks V. S.; Mining frequent itemsets with convertible constraints, in Proceedings of the 17th International Conference on Data Engineering, April 2–6, 2001, Heidelberg, Germany, 2001, pages 433-442
[33] Agrawal, Rakesh; and Srikant, Ramakrishnan; Fast algorithms for mining association rules in large databases, in Bocca, Jorge B.; Jarke, Matthias; and Zaniolo, Carlo; editors, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, pages 487-499
[34] Zaki, M. J. (2000). "Scalable algorithms for association mining". IEEE Transactions on Knowledge and Data Engineering 12 (3): 372–390.
[35] Hájek, Petr; Havel, Ivan; Chytil, Metoděj; The GUHA method of automatic hypotheses determination, Computing 1 (1966) 293-308
[36] Hájek, Petr; Feglar, Tomas; Rauch, Jan; and Coufal, David; The GUHA method, data preprocessing and mining, Database Support for Data Mining Applications, Springer, 2004, ISBN 978-3-540-22479-2
[37] Omiecinski, Edward R.; Alternative interest measures for mining associations in databases, IEEE Transactions on Knowledge and Data Engineering, 15(1):57-69, Jan/Feb 2003
[38] Aggarwal, Charu C.; and Yu, Philip S.; A new framework for itemset generation, in PODS 98, Symposium on Principles of Database Systems, Seattle, WA, USA, 1998, pages 18-24
[39] Brin, Sergey; Motwani, Rajeev; Ullman, Jeffrey D.; and Tsur, Shalom; Dynamic itemset counting and implication rules for market basket data, inSIGMOD 1997, Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1997), Tucson, Arizona, USA, May 1997, pp. 255-264
[40] Piatetsky-Shapiro, Gregory; Discovery, analysis, and presentation of strong rules, Knowledge Discovery in Databases, 1991, pp. 229-248
[41] Brin, Sergey; Motwani, Rajeev; Ullman, Jeffrey D.; and Tsur, Shalom; Dynamic itemset counting and implication rules for market basket data, inSIGMOD 1997, Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1997), Tucson, Arizona, USA, May 1997, pp. 265-276
[42] Tan, Pang-Ning; Kumar, Vipin; and Srivastava, Jaideep; Selecting the right objective measure for association analysis, Information Systems, 29(4):293-313, 2004
[43] Webb, Geoffrey I. (2007); Discovering Significant Patterns, Machine Learning 68(1), Netherlands: Springer, pp. 1-33
[44] Gionis, Aristides; Mannila, Heikki; Mielikäinen, Taneli; and Tsaparas, Panayiotis; Assessing Data Mining Results via Swap Randomization, ACM Transactions on Knowledge Discovery from Data (TKDD), Volume 1, Issue 3 (December 2007), Article No. 14

[45] Witten, Frank, Hall: Data mining practical machine learning tools and techniques, 3rd edition
[46] Z. H. Deng, Z. Wang, and J. Jiang. A New Algorithm for Fast Mining Frequent Itemsets Using N-Lists [1]. SCIENCE CHINA Information Sciences, 55 (9): 2008 - 2030, 2012.
[47] Z. H. Deng and Z. Wang. A New Fast Vertical Method for Mining Frequent Patterns [2]. International Journal of Computational Intelligence Systems, 3(6): 733 - 744, 2010.
[48] Rauch, Jan; Logical calculi for knowledge discovery in databases, in Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery, Springer, 1997, pp. 47-57
[49] Hájek, Petr; and Havránek, Tomáš (1978). Mechanizing Hypothesis Formation: Mathematical Foundations for a General Theory. Springer-Verlag. ISBN 3-540-08738-9.
[50] Webb, Geoffrey I. (1995); OPUS: An Efficient Admissible Algorithm for Unordered Search, Journal of Artificial Intelligence Research 3, Menlo Park, CA: AAAI Press, pp. 431-465
[51] Bayardo, Roberto J., Jr.; Agrawal, Rakesh; Gunopulos, Dimitrios (2000). "Constraint-based rule mining in large, dense databases". Data Mining and Knowledge Discovery 4 (2): 217–240.
[52] Webb, Geoffrey I. (2000); Efficient Search for Association Rules, in Ramakrishnan, Raghu; and Stolfo, Sal; eds.; Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000), Boston, MA, New York, NY: The Association for Computing Machinery, pp. 99-107