Opening the Knowledge Tombs - Web Based Text Mining as Approach for Re-evaluation of Machine Learning Rules

Milan Zorman[1,2], Sandi Pohorec[1], and Boštjan Brumen[1]

[1] University of Maribor, Faculty of Electrical Engineering and Computer Science,
Smetanova ulica 17,
2000 Maribor, Slovenia
[2] Centre for Interdisciplinary and Multidisciplinary Research and Studies of the University of Maribor, Krekova ulica 2,
2000 Maribor, Slovenia
{milan.zorman,sandi.pohorec,brumen}@uni-mb.si

Abstract. Growth of internet usage and content provides us with large amounts of free text information, which could be used to extend our data mining capabilities and to collect specialist knowledge from different reliable sources.

In this paper we explore the possibility for a reuse of 'old' data mining results, which seemed to be well exploited at the time of their formation, but are now laying stored in so called knowledge tombs. By using the web based text mined knowledge we are going to verify knowledge, gathered in the knowledge tombs.

We focused on re-evaluation of rules, coming from symbolic machine learning (ML) approaches, like decision trees, rough sets, association rules and ensemble approaches.

The knowledge source for ML rule evaluation is the web based text mined knowledge, aimed to complement and sometimes replace the domain expert in the early stages.

Keywords: knowledge tombs, data mining, re-evaluation, rules, supervised machine learning.

1 Introduction

In this paper we explore the possibility to begin a reuse of 'old' data mining results, which seemed to be well exploited at the time of their formation, but are now nothing else than knowledge tombs.

Although the data mining community collectively uses and attacks the data tombs created by different institutions and individuals, it also creates vast amounts of potential knowledge, which, considering the available capabilities, may seemed to be well used and exploited at the time of the research.

But is that still true after 5, 10, 15 or 20 years? What seemed to be well exploited basis, may now represent a valuable ore for further data and knowledge mining.

B. Catania, M. Ivanović, and B. Thalheim (Eds.): ADBIS 2010, LNCS 6295, pp. 533–542, 2010.

Just ask yourself or the first data mining researcher you run into, how many chunks of information/rules/decision trees/... you/he produced in the last decade. The answers will easily reach over a few ten or hundred thousand. By leaving them on your hard drives, you are slowly creating new knowledge tombs, with little prospects to be re-opened again.

But how can we re-evaluate knowledge from knowledge tombs? Evaluation of knowledge in the form of rules was mainly a manual job, usually limited to domain experts included in the initial study or research. Human factors, like limited availability, subjectivity, mood, etc., often influenced the evaluation and increased the possibility of missed opportunities.

Growth of internet usage and content in the last 10 years (according to some sources is world average from 400% on [1, 2]) provides us with large amounts of free, unstructured text information, which could be used to extend our data mining capabilities and to collect specialist knowledge from different more or less reliable sources.

And internet sources are the answer - using the web based text mined knowledge to verify knowledge, gathered in the knowledge tombs is approach we will present in this paper.

2 Knowledge Tombs and Forms of Rules for Re-evaluation

At the beginning, let us define the knowledge tombs.

Researchers and users of the artificial intelligence approaches, usually produce different sorts of knowledge, extracted from various data sources. After a quite intensive period of time, when all the evaluations and usage of the knowledge is done, that knowledge is stored in some sort of electronic form and with some 'luck' never to be used again.

Luckily, the times change and with the advancement of technology, expert knowledge is becoming more and more easily accessible through the information highway – the internet. In this paper we are going to address the most common form of internet knowledge, the free, natural language texts and present ways to mine it for knowledge, which will be used for re-evaluation.

Which knowledge tombs are we going to open? Typically, all white box, 'rule producing' machine learning and data mining approaches are the ones that produce knowledge in a form, appropriate for re-evaluation with our method.

In the following subsections, we are going to present the most typical representatives of the symbolic and ensemble approaches and their knowledge representations. The latter is crucial for us, because we need to know exactly what type of knowledge can be found in knowledge tombs.

2.1 Symbolic Machine Learning Approaches

Symbolic machine learning approaches try to capture knowledge in as symbolic a form as possible to provide a very natural and intuitive way of interpretation. Typically, decision trees, association rules, rough sets, and ensemble approaches are used for knowledge extraction. Ensemble methods usually perform better in comparison with single methods regarding classification accuracy, but they produce larger amounts of rules, which makes them the target group for our knowledge re-evaluation approach.

2.1.1 Decision Trees

Decision trees[3] are one of the most typical symbolic machine learning approaches, which have been present on the machine learning scene since the mid-1980s, when Quinlan presented his ID3 algorithm.[4] Decision trees take the form of a hierarchically-bound set of decision rules that have a common node, called a root. The rest of the tree consists of attribute (tests) nodes and decision (leaf) nodes labelled with a class or decision. Each path from a root node to any decision node represents a rule in a decision tree. Because of the very simple representation of accumulated knowledge they also give us the explanation of the decision that is essential in medical applications.

Top-down decision tree induction is a commonly used method. Building starts with an empty tree. Afterwards, a 'divide and conquer' algorithm is applied to the entire training set, where the most appropriate attribute is selected. Selection is based on a purity measure that determines the quality of the attribute split and represents a vital part of the method's background knowledge. A typical purity measure is some sort of derivative of an entropy function. Another very useful advantage of decision trees is the fact that they do not use all available attributes from the training set, but only those that are necessary for building a tree. Reducing the number of attributes (also called horizontal reduction) has very valuable consequences, since it provides information about which attributes are sufficient for a description of the problem and which are redundant.

The knowledge accumulated in the decision tree is represented in the form of a tree of rules, which can be easily transformed into a set of rules and also into a set of used attributes.

What we are interested in, are the rules, which can be easily obtained from the decision tree – each path from the root of the tree to a decision node, gives us one rule.

2.1.2 Association Rules

The association rule approach searches for frequently occurring and, therefore, interesting relationships and correlation relationships among attributes in a large set of data items.[5] Association rules show attribute-value conditions that occur together frequently in a given dataset. Association rules provide information in the form of 'if-then' statements. The rules are computed from the data and are of a probabilistic nature. The 'if' part of the rule is called the antecedent, while the 'then' part is called the consequent. The antecedent and consequent are sets of items that may not have any items in common. Each association rule also has two numbers that express the degree of uncertainty about the rule.

The first number is called the support and is the number of transactions that include all items in the antecedent and consequent parts of the rule. The second number is called the confidence of the rule and is the ratio between the number of transactions that include all items in the consequent, as well as the antecedent and the number of transactions that include all items in the antecedent.

Extracted knowledge of association rules is presented in the form of rules.[5] The difference between this method and the other presented approaches is in the consequent, where decision trees and rough sets use only one consequent attribute for all rules they produce on one data set.

2.1.3 Rough Sets

Rough sets are a symbolic machine learning approach based on classic set theory, which were introduced by Pawlak et al.[6] in 1995. They are often compared with other techniques, especially to statistical analysis and other machine learning methods. It is claimed that rough sets perform better on small data sets and on sets where data distribution significantly differs from uniform distribution.[6] In many cases, detailed data is not required to make decisions as approximate or rough data would be sufficient. Rough sets use reducts, a sufficient subset of attributes, to generate a rule set covering objects from the training set. Reducts are sets of attributes that are actually a subset of the entire set of attributes available in the training set. The rules that are obtained from the rough set approach can be either certain or uncertain. Certain rules are used in cases where there is background in consistent training objects. In contrast, uncertain rules are produced in cases where the training objects are inconsistent with each other. The latter situation usually presents a big hindrance for other machine learning approaches.

The main application areas of the rough set approach are attribute reduction, rule generation, classification and prediction.[6] Both rules and the set of attributes (reduct) are explicitly expressed, so there is no additional effort needed to extract the rules.

2.1.4 Ensemble Methods

Hybrid approaches in machine learning rest on the assumption that only the synergistic combination of different models can unleash their full power. The intuitive concept of ensemble approaches is that no single classifier can claim to be uniformly superior to any other, and that the integration of several single approaches will enhance the performance of the final classification.

To overcome some of disadvantages and limitations of a single method, it is sometimes enough to use different models of the same machine learning approach; e.g. using many different decision trees, neural networks or rough sets for the same training set.[7] In other cases, the approach relies on combining different machine learning or general problem solving methods. Typically, any machine learning methods can be combined in the form of an ensemble, but if symbolic machine learning methods are used, the condition about the possibilities of interpretation and explanation are satisfied. The two most popular ensemble techniques are called bagging[7] and boosting.[8] In simple terms, multiple classifiers are being combined into a voting body that is capable of making decisions with higher accuracy than any single classifier included in the voting body.

Bagging uses random selection of training objects with replacement from the original data set, to form a subset of objects, used for induction.[7] If random selection is used together with replacement, there is a possibility that in the training subset some training objects from original data set occur more than once and some do not occur at all. The drawback of bagging is its random selection, where luck is relied upon to select an appropriate subset of training objects. By counting the votes, the class with the most votes can be assigned to the unseen object. Bagging always improves classification accuracy of any included single classifier.

Boosting uses the idea behind bagging and extends it further. Boosting tries to overcome the problem of random selection in bagging by assigning weights to the training objects and looking back to see how successful the previously induced classifier was.[8] This makes this approach incremental. If a training object was classified incorrectly, its

weight was increased, and if it was classified correctly, its weight was decreased. Weights play the main role in selecting training objects for the next iteration of classifier induction, since the focus is on training objects that have higher weights and are, therefore, harder to classify correctly. The final classifier consists of earlier classifiers that are better on 'easier' training objects with lower weights and latter classifiers, which are more specialized in classifying 'harder' cases with higher weights.

Since both bagging and boosting can be used on wide variety of basic machine learning methods, we are going to limit ourselves only to symbolic white box approaches, where the rules are at hand or can be easily extracted from each member of the ensemble.

2.2 Web Based Text Mining

Text mining is a relatively new research area with first attempts going back approximately 15 years. In that time some authors tackled the problem of extracting company names from a free text. The amount of electronic texts and potential knowledge grew over all expectation since the beginning of internet. It is also known that most of the knowledge found on the internet is unrelated and unstructured. But even on that field a big step forward was made in the recent years – systems for textual information extraction became a key tool for news aggregators, internet people searchers, thesaurus generators, indexing tools, identification of proteins, etc. That kind of systems are not 'smarter' than people, but have two great advantages: they are consistent and much faster than people.

Most of the systems work as unilingual version, but the real challenge represent multilingual and language invariant approaches. Most multilingual approaches have language recognition preprocessors. There are some parallels between mining different languages and specific knowledge domains (using the same language) like medicine, biomedicine, and military.

We believe, that the using the knowledge from different internet sources can take some burden off the shoulders of domain experts and enable faster knowledge acquisition.

In general, we can describe the text mining with the workflow in Fig. 1. Since text mining is a natural language processing, it can be accessed from the same levels as the natural language analysis [9]:

- **Prosody** analyses rhythm and intonation. Difficult to formalize, important for poetry, religious chants, children wordplay and babbling of infants.
- **Phonology** examines sounds that are combined to form language. Important for speech recognition and generation.
- **Morphology** examines word components (morphemes) including rules for word formation (for example: prefixes and suffixes which modify word meaning). Morphology determines the role of a word in a sentence by its tense, number and part-of-speech (POS).
- **Syntax** analysis studies the rules that are required for the forming of valid sentences.
- **Semantics** studies the meaning of words and sentences and the means of conveying the meaning.
- **Pragmatics** studies ways of language use and the effects the language has on the listeners.

In our work we will be focusing on the morphology, syntax and semantic levels.

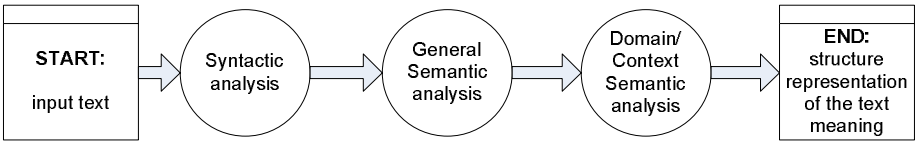

Fig. 1. Natural language analysis process

When considering means to acquire knowledge from natural language sources the analysis is a three step process: **syntactic analysis**, **meaning analysis** (semantic interpretation; generally in two phases) and the **forming of the final structure** that represents the meaning of the text.

In the following section we will continue with a somewhat simplified example of knowledge extraction from natural language texts. The example will be based on natural language passages and how to extract formalised data from them. The two main steps in the process are the following:

1. Acquisition of the natural language resources and pre-processing.
2. Knowledge extraction and formalization.

2.2.1 Acquisition of the Natural Language Resources and Pre-processing

The acquisition process is the process in which we define the source of data and means to acquire it. It can be as simple as gathering the documents to a central storage point or it can be the implementation of a web crawler that will investigate the target web sites and transfer the data to the central storage point.

The essential steps in the preprocessing are two. The first is the transformation of the documents to plain text and the second is the tokenization. While the former is an entirely technical issue that can be successfully solved without significant effort, the latter requires much thorough approach and is far from trivial. The tokenisation is essential for the passage and sentence level semantic analysis. However some semantic information is required for the successful resolution of the meaning of punctuation (for instance whether a period ends a sentence or just an abbreviation). The simple implementation where the period is assumed to end a sentence proved to achieve a successful tokenization rate of just under 90% on the Penn Treebank corpora. Unfortunately that is not enough, and a higher success rate is required because of an error in tokenisation, which is usually magnified by several orders in the analysis phase.

2.2.2 Knowledge Extraction and Formalisation

The first step in knowledge extraction is the part-of-speech (POS) analysis. It can be performed with the use of existing POS taggers, although it is highly language dependant. In the example we are presenting we will assume that the source documents have been gathered, transformed to plaintext and tokenised to individual sentences. The sentences to be used for semantic analysis can be classified by statistical metrics. Let us assume that a sentence has been identified. The sentence is stated as:

"Eating as little as a quarter of an ounce of chocolate each day may lower your risk of experiencing heart attack or stroke!".

The POS analysis provides the tags listed in Table 1.

Table 1. POS tags of a news sentence

Word	Tag	Word	Tag	Word	Tag
Eating	VBG	*as*	RB	*little*	JJ
as	IN	*a*	DT	*quarter*	NN
of	IN	*an*	DT	*ounce*	NN
of	IN	*chocolate*	NN	*each*	DT
day	NN	*may*	MD	*lower*	VB
your	PRP$	*risk*	NN	*of*	IN
experiencing	VBG	*a*	DT	*heart*	NN
attack	NN	*or*	CC	*stroke*	VB

Abbreviations: IN - Preposition or subordinating conjunction, JJ - Adjective, MD - Modal, NN - Noun, singular or mass, PRP$ - Possessive pronoun, RB - Adverb, VB - Verb, base form, VBG - Verb, gerund or present participle

Semantic interpretation uses both the knowledge about word meanings (within the domain) and linguistic structure. The analysis produces a representation of the meaning of the sentences in an internal format. This can be visualised as in Fig. 2.

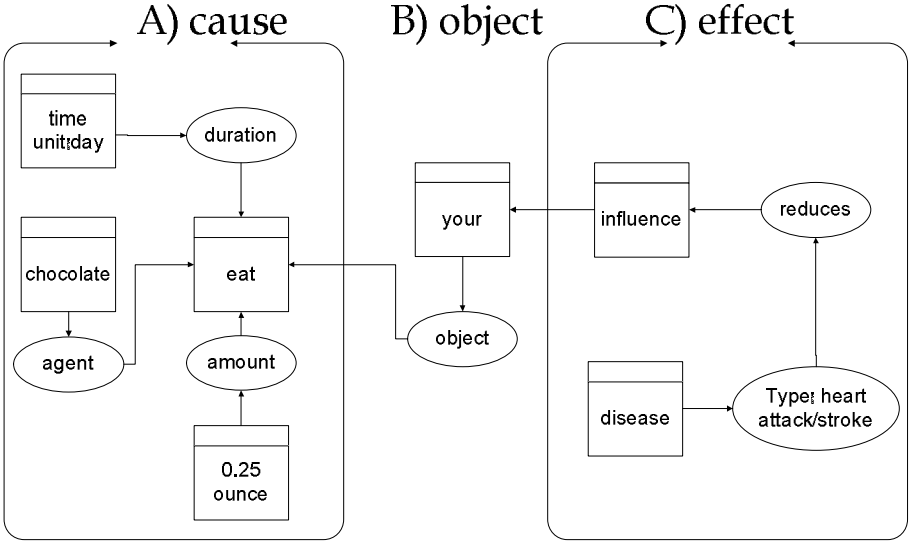

Fig. 2. Internal representation of the meaning of the sentence

The process of the analysis is described in the following. The sentence is separated into two distinct categories: cause (IF) and effect (THEN). Both are associated with the object. In the figure the application used knowledge that *ounce* is a unit of amount, *day* is a unit of time and that a person normally eats chocolate not the other way around. So combining this knowledge produced the resulting representation of knowledge in the sentence. The agent (the one that influences) is *chocolate*, the object (the recipient of the action) is the word *your* and the action (agent to object) is *eating*. Combining that to *eat* is associated with the domain concept of amount and that *ounce* is a unit of amount the application can effectively reason that the meaning of the cause part (Fig. 2 segment A) of the sentence is: *object that eats a 0.25 ounce of chocolate in a period of one day*. The effect side (Fig. 2 segment C) has the meaning of: *the object experiences the influence of reduced possibility of a disease of type heart attack/stroke*. This internal representation is then generalized with the addition of known concepts. The object *yours* is a possessive pronoun and is therefore mapped to a person which is marked as "*patient*„ in the domain.

The amount of *quarter of an ounce* is mapped to the primary unit for amount in the domain, (*grams*) with the use of a conversion factor. So ¼ of an ounce becomes 7.08738078 grams. The resulting semantic net (Fig. 2) with the additional information is the final interpretation of the domain specific world knowledge learned from this sentence.

The form shown in Fig. 2 is the form that can be used for the formalisation of knowledge. The formalisation is the process of storing the knowledge in a formal, machine readable form that can be used as the need arises by various types of intelligent systems. A common formalisation approach is the transformation to rules. For the example we have been following a rule would be in the following form:

```
RULE chocolate consumption influence
  IF typeof (object) IS patient
    AND typeof (action) IS eat
    AND action::target IS chocolate
    AND  quantityof (action) IS 7g
    AND  timespan (action) IS 24h
  THEN typeof(consequence) IS influence
    AND      consequence::target IS disease
    AND  typeof(disease) IS heart attack/stroke
    AND relationship (consequence,
consequence::target) IS reduced risk
```

This is the final formalization of acquired knowledge. In this form the knowledge is fully machine readable, providing there are inferring rules that define how to evaluate the value entities (*typeof*, *quantityof*,…). This format can be stored and used as need arises.

What about the rules, we want to re-evaluate? Even they must be checked and put into context. To do that, we have to process the descriptions of the underlying database, which usually contain the descriptions of the problem domain and the attributes used in the database and (consequently) in the rules produced by the machine learning method. This process is a simpler version of the process, described above, since the rules are already in the formal, machine readable structured form.

2.2.3 Re-evaluation of the Rules and Search for New Knowledge

Finally we come to the part, where we can start to compare the rules, which are in the same formal form, but come from different data sources.

Our goal is to find support for machine learning rules, which we brought from our knowledge tomb in the text we mined from our natural language resource and is now also in a form, suitable for comparison. We will be looking for the highest match between the cause(s) and effect(s) between individual representatives from both sets of rules.

After a series of automatic procedures, there is again time, to involve a human operator, a domain expert, who will examine machine learning rules with the top ranked support, found on the web. His task is to extract new and potentially new knowledge by examining the machine learning rules and by standing rules with text sources, from which they were derived.

Explanations in the natural language are there to encourage thinking from different perspectives and hopefully provide a reasonable explanation and help at revealing new or unconscious knowledge. It is up to expert's expertise to recognize, expose and reason about the possible new knowledge, but now with higher level of support, than years ago, when the knowledge tomb was created.

3 Discussion and Conclusions

Knowledge mining from databases of solved cases with machine learning methods is nothing new in community involved with artificial intelligence. Used approaches are very different, but most of them have a common point –machine learning approaches and their outcomes. Results of such approaches are in general sets of rules with some additional information, which are capable to generalize knowledge from database of solved cases or to determine associations between attributes in the data base.

It was usual practice that the generated set of rules was checked by a domain expert (for example a medical doctor), which used his knowledge and experience to evaluate rules as senseless or sense, and the latter to known or potentially new. The last type of knowledge is the most interesting for us, because by discovering new knowledge we can find solutions to unsolved cases or find alternative solutions for solved problems. It is our experience that new knowledge is very hard to find so we decided to find a way to automatically support a part of knowledge evaluation.

Re-evaluation of ML rules is a process which increases the potential of already generated, but disregarded rules and hopefully triggers the 'Aha!' effect which accompanies the transformation of a rule in to a new knowledge.

The described re-evaluation should not become only one-time event, but should become a process which takes part on a regular basis.

Faster and more consistent knowledge verification reduces the need for manual domain expert work and shortens the cycle *'searching for potential knowledge – verifying potential knowledge – using new knowledge'*.

Of course there are some concerns, which we are aware of and present a pit fall for the re-evaluation process. With the increased amount of information, available on the internet, there is also a vast number of sources we cannot trust and because of that, the involvement of a domain expert in the final stage of knowledge re-evaluation remains a necessity.

References

1. Internet Growth Statistics - Today's road to e-Commerce and Global Trade (2010), http://www.internetworldstats.com/emarketing.htm
2. Malik, O.: Big Growth for the Internet Ahead, Cisco Says (2010), http://gigaom.com/2008/06/16/big-growth-for-internet-to-continue-cisco-predicts/
3. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
4. Quinlan, J.R.: Induction of decision trees. Machine Learning, 81–106 (1986)
5. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press, Cambridge (1991)
6. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., et al.: Rough sets. Communications of the ACM 38, 89–95 (1995)
7. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
8. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference, pp. 148–156. Morgan Kauffman, San Francisco (1996)
9. Luger, G.F.: Artificial intelligence, Structure and Strategies for Complex Problem Solving, 5th edn. Pearson Education Limited, USA (2005)