Automated Identification of Business Rules in Requirements Documents

Richa Sharma
School of Information Technology
IIT Delhi
India

Jaspreet Bhatia
School of Information Technology
IIT Delhi
India

K.K. Biswas
Dept. of Computer Science and Engineering, IIT Delhi
India

Abstract— **Business Rule identification is an important task of Requirements Engineering process. However, the task is challenging as business rules are often not explicitly stated in the requirements documents. In case business rules are explicit, they may not be atomic in nature or, may be vague. In this paper, we present an approach for identifying business rules in the available requirements documentation. We first identify various business rules categories and, then examine requirements documentation (including requirements specifications, domain knowledge documents, change request, request for proposal) for the presence of these rules. Our study aims at finding how effectively business rules can be identified and classified into one of the categories of business rules using machine learning algorithms. We report on the results of the experiments performed. Our observations indicate that in terms of overall result, support vector machine algorithm performed better than other classifiers. Random Forest algorithm had a higher precision than support vector machine algorithm but relatively low recall. Naïve Bayes algorithm had a higher recall than support vector machine. We also report on evaluation study of our requirements corpus using stop-words and stemming the requirements statements.**

Keywords—Business Rules; Requirements; Machine Learning

I. INTRODUCTION

Business Rules represent the policies and the regulations followed by organizations for smooth operation of their business. Terry Moriarty defines business rules as constraints placed upon the business [1]. Tony Morgan defines business rule as "a compact statement about an aspect of the business. It is a constraint in the sense that a business lays down what must or must not be the case" [2]. Ronald Ross defines business rule as "a discrete operational business policy or practice. A Business rule may be considered a user requirement that is expressed in non-procedural and non-technical form. A business rule represents a statement about business behavior." [3]. The Object Management Group defines a rule as claim of obligation or of necessity and, a business rule as a rule under business jurisdiction [4]. Business Rules Community widely accepts the definition produced by Business Rules Group as "business rule is a statement that defines or constrains some aspects of the business, intended to assert business structure, or to control or influence the behavior of the business. A business rule cannot be broken down or decomposed further into more detailed business rules. If reduced any further, there would be loss of important information about the business" [5]. Though defined differently, business rules constitute the policies governing operations in an organization. Atomicity and discreteness are important aspects of business rules as suggested by various authors. Identifying and understanding the business rules discretely is, however, a challenging task.

With increasing adoption of Information Technology in organizations, business rules are now part of the information systems responsible for carrying out the organization's operations. Business rules only serve as the basis for functional requirements for the information systems. It is, therefore, important for requirements engineers to comprehend the business policies underlying the documented functional requirements. Requirements for information system are captured in the form of natural language (NL) documents, referred to as Software Requirements Specification (SRS). Other possible forms in which requirements for information system are captured or documented include Request for Proposal (RFP), Domain Knowledge Documents (DKD) and Change Requests (CR). These documents are also written in natural language. The vagueness and ambiguity issues of NL further aggravates the problem of identifying business rules in the requirements documentation. Secondly, business rules, which provide the rationale for functional requirements, are not explicitly stated in these documents. Even if these rules are stated explicitly, it is possible that either they are not atomic in nature or are vague. It is possible that one requirements statement correspond to multiple categories of business rules resulting in violation of atomicity of business rules as well as leading to vagueness. We, therefore, propose an approach that will assist requirements engineers in identifying those requirements statements that correspond to certain business rules.

Our approach is based on machine learning and classification algorithms that classify and report the category of business rules existing in a requirement statement. Business rules (BRs) have been studied extensively in business analysts' community and various viewpoints of BRs category have been proposed. After performing literature study on various viewpoints of BRs, we have identified 9 categories of these rules. Our categories are also influenced by the need of information required from the perspective of developing an information system. If the requirements statements are marked with appropriate business rule category, then requirements analysts can further refine those statements, if required. Our study aims at exploring following research questions:

978-1-4799-2572-8/14/$31.00 ©2014 IEEE

1. Which machine learning algorithm performs best for identifying business rules in the requirements documents?

2. Do the word formulations impact classifier performance?

3. Which document category serves better in terms of capturing business rules?

The rest of the paper is organized as follows. Section II presents an overview of the background for this paper. Then, section III discusses our approach and the evaluation study. Section IV presents the results, observations and limitations of our study. This is followed by discussion and conclusion in section V.

II. BACKGROUND

A. Business Rules

As discussed in section I, BRs represent operational or governing policies of an organization. The introduction of Information Technology motivated practitioners to introduce information system perspective to the existing business perspective to BRs. The business perspective pertains to the constraints that apply to the behavior of people in organization as well as the processes followed in the organization; and, the information system perspective considers the facts which are recorded as data and constraints on changes to values of those facts [6]. Different authors [6], [7], [8] have presented different categories of BRs with certain degree of overlaps.

[6] categorizes business rules into three broad categories:

1. Structural Assertions – These refer to a defined concept or, a factual statement that expresses some aspect of the structure of an organization. Business Terms and Fact - are the two sub-categories of structural assertions. Facts can be base facts or derived facts.

2. Action Assertions – These refer to specific constraints or conditions that limit or control the actions of an enterprise. Action assertions are further classified as – Constraint, Condition and Authorization. Another classification types considers action assertion as action controlling assertion and, action enabling assertion.

3. Derivation – Derivation is a statement of knowledge derived from other knowledge in business.

[7] identifies following categories of rules:

1. Presentation Rules – These rules are concerned with the interactions with information system users.

2. Database Rules – These rules refer to database design and possible constraints on the data.

3. Application Rules – These rules are related to the processing logic for business.

[8] observes following business rule categories:

1. Term – refers to word, phrase or sentence that has a specific meaning for the business.

2. Fact – refers to an association between two or more fact relating terms. Facts connect things in business.

3. Constrains – refer to circumstances under which a business event should occur.

4. Action Enablers – refer to circumstances that enable or trigger the organization to take an action.

5. Derivations – refer to deriving new knowledge from existing knowledge (usually mathematical expressions).

6. Inferences – refer to knowledge transformations from one form to another, often taken from non-numeric data.

The above-mentioned literature study indicates overlaps in the classification of business rules. Structural assertions as proposed in [6] are similar to terms and facts proposed in [8] and, these represent the requisite ontology for the organization. Such similarity exists between action assertions in [6], application rules in [7] and constraints and action enablers in [8]. Presentation rules and Database rules are not explicitly part of classification scheme in [6] and [8]. For the purpose of our study, we have combined these different categories of BRs and removed overlaps between them. Considering the information as well as business perspective, we have identified following 9 categories of business rules based on the literature study and our requirements corpus:

1. Presentation Rules – These rules present the guidelines for user interaction with the software and the layout of the user interface.

2. Term – These are the defining terms of the business environment.

3. Fact – Facts represent any factual information about the business.

4. Attribute – These represent the attributes that define an object or an entity in the business environment.

5. Role – These rules represent the roles of various entities in the business environment.

6. Condition – These rules describe the conditions for an event or action to happen.

7. Constraint – These rules describe any constraints for an event or action to happen or an assertion that must always be true.

8. Privilege or Authorization – These rules describe various privileges that different users of the system can enjoy.

9. Period – The rules representing time-period related information and, are sub-types of term. Period has been introduced to represent separately those facts that have some time-period associated information for an action or an event.

The motivation behind introducing these categories is to assist analysts by presenting him with a classified distribution of business rules. This, in turn, can help analysts in observing and analyzing the relevant business rules in requirements statements quickly and efficiently

B. Business Rules and Requirements

Information Systems are now common in enterprise organizations and are responsible for implementing business operations and, consequently business rules. Information systems are developed on the basis of requirements gathered during requirements engineering phase of software development lifecycle. The question: how do requirements and business rules relate has been explored and analyzed in [9]. The author argues that it is the business rules what functional requirements should know – the guideline, the assumption, the control underlying the functionality. Consider the requirement statement: *System should be able to process loan which is in active state*. This requirement statement does not explicitly mention what is meant by '*loan in active state*' and what is meant by '*processing*'. Even if these are mentioned explicitly, it is possible some other related conditions that hold true when loan is active are still missed because 'processing' part does not refer to those conditions.

It is, therefore, easy to miss or overlook the underlying BRs or functionality behind a requirements statement. Ambiguity, vagueness and contradicting information can further be misleading. Developers, often, tend to make assumption for such missing or contradicting information at the time of development as reported in [10]. Such knowledge has been referred to as implicit or tacit knowledge [11] in requirements study. The cost of missing the underlying business rules (tacit knowledge) can be very high in terms of defects surfacing in the later phases of software development lifecycle. Capturing business rules during requirements engineering phase can considerably reduce number of the potential defects in the information system. Motivated by the need of uncovering and analyzing the business rules at the time of requirements analysis, we have explored the possibility of leveraging classification algorithms for automatically identifying business rules from various forms of requirements documents.

C. Classification Algorithms

Sentence classification algorithms have been widely used for various purposes like spam mail detection [12], news articles classification (with focus on terrorist incidents) [14], non-functional requirements extraction [15] etc. These algorithms are either based on keyword-search or machine learning classification. We are interested in classifying the requirements statements into one of the categories of BRs without making use of keyword-search as keywords can be domain-specific. We found that machine learning classification algorithms can provide a foundation for the purpose of our study. Machine learning is about the construction and study of systems that can learn from the data. Though there are various algorithms for machine learning but these algorithms are classified into two broad categories: supervised and unsupervised learning. Supervised learning makes use of the guiding function that maps inputs to desired outputs (also referred to as labels, because these are often provided by human experts labeling the training set). On the other hand, unsupervised learning models a set of inputs by grouping or clustering common instances/patterns.

For our study, we have made use of supervised machine learning technique. We have taken labeled or annotated documents as input to our study. We have made use of Support Vector Machine, Naïve Bayes, BayesNet and Random Forest algorithms in our study. Naïve Bayes classifier is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. It assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. Despite this assumption, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. Bayesian network, in contrast, makes use of conditional dependencies. Support Vector Machines classifier is also a supervised learning model that works by identifying optimal separator between two classes. Random forests are an ensemble learning method for classification (and regression) that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

We have used precision, recall and F1 measure to compare the results of these learning algorithms. Precision is the fraction of retrieved instances that are correct and recall is the fraction of correct instances that are retrieved. We define precision, P and recall, R for our study as:

P = True Positive / (True Positive + False Positive)

R = True Positive / (True Positive + False Negative)

Here, True Positives indicate correct predictions. False Positives are incorrectly labeled as belonging to the class.

False Negatives are the predictions which were not labeled as belonging to the positive class but should have been.

F-measure considers both the precision and the recall of the test. The F-measure can be interpreted as a weighted average of the precision and recall, where it reaches its best value at 1 and worst score at 0. F-measure is defined as:

F-measure = 2x (P x R) / (P + R)

III. Our Contribution

In this section, we present our approach towards automatic extraction of BRs in the requirements documentation. We first created requirements corpus, formulated the research questions and then, conducted the experimental study in view of the questions articulated.

A. Creating Requirements Corpus

We have built our requirements corpus by collecting documents from domains like academics, loan, insurance, medical and control systems. The sources of our documents include industry affiliation and public domain. We have used three SRS documents, two Request for Proposal (RFP) documents, two Domain Knowledge Documents (DKD) and two Change Requests (CR).

We have made use of text version of these files without making any changes to the original file. We simply copied the documents to text file. The only information in the documents which we have not processed is the pictorial and tabular representation of the information. We, then, annotated the corpus for the presence of different rule categories. We allowed

one statement to belong to more than one category of the BRs. The task of annotation was performed by 5 subjects to ensure that our results are not influenced by an individual's thought process. The subjects chosen are research scholars – all of them have done courses on Software Engineering and Business Modeling, and two of them had industry experience too. The details on the size of the documents is detailed out in table – I. Our corpus had nearly 1318 presentation rules, 1752 constraints, 695 terms, 357 conditional statements; 313 facts, 230 statements for attributes. The count for role and period rules was 63 and 48 respectively. These details are based on the labeling performed by the first author of the paper. Since manual annotation is subjective, therefore it could be a potential threat to the validity of our results. We mitigated this threat by discussing the business rule category definitions thoroughly in a meeting to ensure that each subject is familiar and comfortable with the rule descriptions. We also performed validity check for annotation by labeling random sample of 100 statements in that meeting and performing peer review of those annotations. The result of peer review revealed that there are not drastically differing views of the rule-labeling. We, then, performed our experiments on annotated requirements corpus.

TABLE I. REQUIREMENTS CORPUS DETAILS

Document	Type	Size (No. of statements)
Doc1	SRS	248
Doc2	SRS	798
Doc3	SRS	1165
Doc4	RFP	861
Doc5	RFP	1893
Doc6	DKD	171
Doc7	DKD	107
Doc8	CR	109
Doc9	CR	155
Total		**5507**

B. *Evaluation Study*

We performed our experiments by applying Naïve Byes, BayesNet, Support Vector Machine and Random Forest algorithms to our annotated corpus. We evaluated the classification results with n-fold cross-validation and computed precision, recall and F-measure for each of the classifier. In n-fold cross-validation, data is distributed randomly into n folds where each fold is approximately of equal size and equal response classification. We have used 10-fold cross-validation as recommended by [15]. We have used Weka[1] tool for our study. We loaded the files in Weka and converted the annotated statements to word vectors using Weka filter. As a first step in our study, we used the documents in their original form, i.e. without any change to the word form or applying any filter. The results of the first step of our study are presented in table – II. As the next step to our evaluation study, we made use of stop-words at the time of classification. Stop-words refer to a list of words that should be filtered out during classification due to either commonality of words or domain-specific generality of words. We first considered determiners only (a, an, the) as stop-words and, found improvements in results as indicated in table III.

TABLE II. 10 – FOLD CROSS VALIDATION STUDY

Classifier	Precision	Recall	F-measure
SMO	**0.680**	**0.591**	**0.624**
BayesNet	0.517	0.568	0.526
Naïve Bayes	0.398	**0.739**	0.481
Random Forest	**0.780**	0.459	0.540

TABLE III. 10 – FOLD CROSS VALIDATION WITH DETERMINERS AS STOP WORDS

Classifier	Precision	Recall	F-measure
SMO	**0.687**	**0.589**	**0.627**
BayesNet	0.543	0.568	0.527
Naïve Bayes	0.406	**0.733**	0.489
Random Forest	**0.784**	0.433	0.518

There are other sets of stop-words used for filtering purpose while processing natural language text. We chose one of such existing lists, referred to as "Glasgow"[2] list of stop-words. The results obtained by using these stop-words are presented in table IV. We noticed that using Glasgow stop-words, the classification performance did not show any improvement; instead, the precision and recall values lowered marginally as compared to the results in table - II.

TABLE IV. 10 – FOLD CROSS VALIDATION WITH GLASGOW LIST OF STOP WORDS

Classifier	Precision	Recall	F-measure
SMO	**0.666**	**0.549**	**0.593**
BayesNet	0.537	0.568	0.509
Naïve Bayes	0.431	**0.661**	0.496
Random Forest	**0.766**	0.479	0.556

The classification of natural language sentences depends on several features of sentences like the words themselves, parts of speech, stems, lemmas etc. We have utilized stem form of words in our study. We have modified the standard Porter[3] algorithm for stemming the word form and, included the stems of words available in WordNet[4]. If stem form is not available in wordnet, then Porter's stem form has been used. Table V summarizes the results of using stemming the documents.

[1] http://www.cs.waikato.ac.nz/ml/weka/

[2] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words
[3] http://tartarus.org/martin/PorterStemmer/
[4] http://wordnet.princeton.edu/

Encouraged by the results obtained after stemming and after applying determiners as stop-words separately, we further evaluated the combination of both of these approaches; the results of which are postulated in table VI:

TABLE V. 10 – FOLD CROSS VALIDATION WITH STEM FORM OF WORDS

Classifier	Precision	Recall	F-measure
SMO	**0.687**	**0.595**	**0.631**
BayesNet	0.529	0.582	0.533
Naïve Bayes	0.394	**0.746**	0.478
Random Forest	**0.789**	0.446	0.527

TABLE VI. 10 – FOLD CROSS VALIDATION WITH STEM FORM OF WORDS AND DETERMINERS AS STOP WORDS

Classifier	Precision	Recall	F-measure
SMO	**0.690**	**0.598**	**0.634**
BayesNet	0.494	0. 628	0.518
Naïve Bayes	0.425	**0.739**	0.483
Random Forest	**0.796**	0.452	0.533

We discuss the observations from our evaluation study in the next section.

IV. OBSERVATIONS AND LIMITATIONS

A. Observations

We present our observations in the light of the research questions under study:

1. *Which machine learning algorithm performs best for identifying business rules in the requirements documents?*

To answer this question, we consider the results presented in the above five tables from table II to table VI. The results presented in these tables are an average of the classification results from each of the subject's annotated corpus and, the results have been rounded up to third place of decimal. The individual subjects' results were earlier averaged for all the business categories and, then these results were further averaged across all the subjects. Since these are average results for our corpus, therefore, we can infer that overall, SMO has performed better than other classifiers. It has got highest F-measure among other classifiers and, it remains consistently high for different stop-words filtering as well as stem form of words. However, recall is highest for Naïve Bayes and precision is highest for Random Forest classification algorithms. Since our classification goal is to identify the BRs from the requirements document corpus, we would like to have higher recall as given by Naïve Bayes classifier. In that sense, Naïve Bayes classifier has outperformed other classifiers.

2. *Do the word formulations impact classifier performance?*

Results from table II to table VI are again considered to answer this question. From these tables, we observe that the performance of all classifiers improved by applying stop-words filter, stemmed form of words as well as combination of both of these. SMO's performance increased considerably, though precision of Random Forest increased marginally only using determiners as stop-words. However, the recall of Naïve Bayes decreased minimally as indicated in table - II. The results of applying Glasgow stop-words list are not encouraging though. The stemmed word forms also improved performance of the classifiers though F-measure for Naïve Bayes and Random Forest reduced marginally. However, the improvement in performance of classifiers is comparable to the results obtained using determiners as stop-words. F-measure for SMO, precision for random-forest and, recall for Naïve Bayes have shown significant improvement against the results from original word forms.

The results from table VI indicate that combining stemmed word forms and determiners as stop-words has also improved the performance of classifiers against original word forms and using these two approaches individually. However, the recall of Naïve Bayes reduced marginally as compared to the recall using stemmed word forms and, is same as the recall from the original word form. However, there is an overall increase in performance of SMO and the precision of random forest has also increased considerably. These results indicate that word formulations do impact the results of classifying the requirements statements into different BRs categories.

3. *Which document category serves better in terms of capturing business rules?*

Requirements are initially documented in the form of RFP document and then, are elaborated in SRS documents after a detailed analysis of RFP and interaction sessions with client teams. Requirements Engineers, often, refer to DKD documents for reference and any clarifications, if required. RFP document lays down the as-is processes in the organization as well as the expectations from the proposed information system. SRS document is the detailed form of expectations from RFP. DKD documents contain business policies. However, these documents do not state rules or expectations from the perspective of information systems. Another form of requirements documentation is CR. CRs contain minor to moderate changes expected in the existing information system. As the description of these documents indicate, RFP and SRS relatively serve better in terms of capturing the BRs.

Fig. 1 below shows the distribution of different types of BRs across different forms of requirements documentation. Series 1 represents the distribution of BRs in SRS; series 2 represents the same in RFP documents; and, the distribution of BRs in CR and DKD is represented through series 3 and series 4 respectively. These distribution curves have been obtained by taking average of precision accuracy of classification of the BRs using SMO. We chose SMO for the purpose as overall, SMO classification accuracy has been better in our study. The distribution in Fig. 1 supports the fact that RFP and SRS are comparatively better resources of finding business rules from the perspective of both the organizational policies as well as the expectations from an information system.

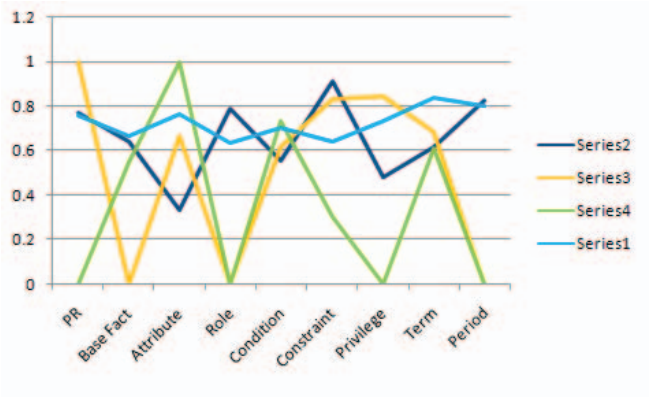

Figure 1. Distribution of Business Rules across Requirements Documents.

B. *Limitations*

One of the limitations of our work is that we have not been able to process the pictorial and tabular representations of the information present in requirements documents. While copying from original documents to textual version, we lose these representations. Analysts, often, prefer to write SRS in the form of tabulated use-cases. The structural information of the organization representing roles, attributes etc. are, at times, represented pictorially. Manual annotation is also one of the limitations of our approach. However, this limitation can be mitigated by considering average results for the inputs taken from various subjects. We have been able to mitigate this limitation in our evaluation study by following similar approach.

V. Discussion And Conclusion

In this paper, we have presented preliminary study towards automatically identifying business rules at the time of requirements gathering and analysis. Business Rules classification has been there in business community and has been a manual effort so far. Business rules form the background of the requirements for an information system. As discussed earlier, these are often not stated and understood clearly or, are overlooked in requirements documents. We have proposed an approach towards automatically identifying business rules in the requirements documents. Our approach can help analysts in refining the requirements statements, in case the statement does not represent atomic business rule or, is ambiguous or, contradictory. We further aim to improve our study by making use of heuristics for classification. We also aim to study our approach in a particular business domain by collecting documents from different sources in that domain only.

References

[1] T. Moriarty T., “The Next Paradigm.”, Database Programming and Design, vol. 6, no. 2, 1993, pp. 66-69.

[2] T. Morgan, Business Rules and Information Systems,Addison-Wesley 2002.

[3] R. G. Ross, The Business Rule Book (IInd Edition), Business Rule Solutions, 1997.

[4] http://www.omg.org/spec/

[5] E. Gottesdiener, "Capturing Business Rules", Software Development Magazine: Management Forum, vol. 7, no. 12, December 1999.

[6] D. Hay and K.A. Healy, “Defining Business Rules ~ What Are They Really?”, GUIDE Business Rules Project, Final Report - revision 1.2, GUIDE International Corporation, Chicago, July 2000.

[7] C.J. Date, “What Not How: The Business Rule Approach to Application Development”, Addison-Wesley, 2000.

[8] E. Gottesdiener , “Business Rules Show Power, Promise,” Application Development Trends, vol. 4, no. 3, 1997, pp. 36-42.

[9] E. Gottesdiener , “Business Rules Rule,” QSS (now IBM TeleLogic) Newletter, 2000.

[10] O., Albayrak, H., Kurtoglu and M., Biaki, “Incomplete Software Requirements and Assumptions Made by Software Engineers”, Asia-Pacific Software Engineering Conference, (APSEC 2009), pp.333-339.

[11] L. Ma, B. Nuseibeh, P. Piwek, A. De Roeck, A. Anne and Willis, “On presuppositions in requirements”, In Proceedings: 2009 Second International Workshop on Managing Requirements Knowledge (MaRK'09), September 2009, Atlanta, Georgia, USA.

[12] I., Androutsopoulos, G., Paliouras, V. Karkaletsis, G. Sakkis, C., D., Spyropoulos and P., Stamatopoulos, “Learning to Filter Spam E-Mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach”, In proceedings of workshop on Machine Learning and Textual Information Access, H. Zaragoza, P. Gallinari and M., Rajman (Eds.), 4th European Conference on Principles and Practice of Knwoledge Discovery in Databases (PKDD), 2000, pp. 1-13.

[13] R., Mason, B., McInnis and S. Dalal, “Machine Learning for Automatic Identification of Terrorist Incidents in WorldWide News media”, International Conference on Intelligence and Security Informatics, 2012, pp. 11-14.

[14] L. Slankas and J. Williams, “Automated Extraction of NonFunctional Requirements in Available Documentation”, In International Workshop on Natural Language Analysis in Software Engineering (NaturaLise), colocated with International Conference on Software Engineerin, ICSE-2013, pp. 9-16.

[15] J. Han, M. Kamber, and J. Pei, “Data Mining: Concepts and Techniques”, 3rd ed. Morgan Kaufmann, 2011..