Relation Identification in Business Rules for Domain-specific Documents

Abhidip Bhattacharyya[*]
University of Colorado Boulder
Boulder, CO
abhidip.bhattacharyya@colorado.edu

Pavan Kumar Chittimalli
TCS Innovation Labs, TRDDC
Pune, India
pavan.chittimalli@tcs.com

Ravindra Naik
TCS Innovation Labs, TRDDC
Pune, India
rd.naik@tcs.com

ABSTRACT

This paper focuses on an approach to mine business rules from documents and facilitates a methodology to represent them in a formal notation. Businesses are operated abiding by some rules and complying with respect to regulation and guidelines. The business rules are often written using English in operating procedures, terms and conditions, and various other supporting documents. The manual analysis of these rules for activities like impact analysis, maintenance, business transformation leads to potential discrepancies, ambiguities, and quality issues. In this paper, we discuss our approach of mining relations among the rule intents (atomic facts) defined for business rules. We also present our preliminary studies on a couple of openly available documents.

CCS CONCEPTS

• **Information systems → Information extraction**; **Clustering and classification**; **Business intelligence**; • **Software and its engineering → Software maintenance tools**; *Specification languages*;

KEYWORDS

Business Rule Extraction, Document Mining, Natural Language Processing, Maximum Entropy

ACM Reference Format:
Abhidip Bhattacharyya, Pavan Kumar Chittimalli, and Ravindra Naik. 2018. Relation Identification in Business Rules for Domain-specific Documents. In *Proceedings of Innovations in Software Engineering Conference, Hyderabad, India, February 9–11, 2018 (ISEC '18),* 6 pages.
https://doi.org/10.1145/3172871.3172884

1 INTRODUCTION

Businesses are driven by the rules. IT system that automates the processes of the business implements these business rules. These rules, are usually created by business analyst, typically reside in documents written in natural language (predominantly in English). The business rules are extracted from the documents for conducting reviews, implementing in IT system and analyzing for various business activities. This process of manual extraction often poses problems because of tremendous effort and the size of the documents.

Semantics of Business Vocabulary and Rules™ (SBVR™) [4] is a standard for business rule representation by Object Management Group (OMG) [2]. SBVR is a Controlled Natural Language (CNL) and describes rules considering only a set of predefined business vocabularies. SBVR provides a natural language interface with first order logic (FOL). These semantics of SBVR makes it ideal for representing business rules.

In our previous work [8], we have proposed a method to extract *rule intents* using *dependency tree* structure of a rule sentence. The *rule sentence* is a sentence from the document that represents a *fact* or combination of facts. A *rule intent* is an atomic fact or predicate present in a rule sentence.

Extending our earlier work, in this paper, we present an approach where try to capture logical relations among rule intents using a directional graph. We propose *this graph* as an intermediate step in eventually extracting business rules aligning with SBVR™ standard. After extracting rule intents and relations, a combination of business vocabulary and the techniques proposed by Bajwa et al. [6] can be a potential way forward to create the SBVR rules automatically. Moreover an approach like P.Chittimalli and K.Anand [10] may be helpful to verify the consistency of rule intents represented using SBVR. The main contributions of our paper are given below:

(1) Use maximum entropy classifier to extract pairwise relation between rule intents.
(2) Create graph using relations as edges and rule intents as nodes, and reduce the graph to a single node using heuristics.
(3) Generate SBVR like output using SimpleNLG API.

2 RELATED WORK

The research problems like extracting rules from legacy code [17, 25, 26] and knowledge extraction from documents [11, 19, 20] have been explored widely. There is not much of work exists in the area of business rule extraction from documents other than [15]. The existing methods to mine rules however can broadly be classified into two categories 1) NLP techniques using shallow parsing 2) NLP techniques using finer level of models.

The first category of techniques [21, 24, 27] uses shallow parser [9]. The three works mentioned above use a domain dictionary to classify the verbs into some predefined class depending on semantic

[*] Author has done this work while he was working in TRDDC. He is currently pursuing his PhD at University of Colorado Boulder.

ISEC '18, February 9–11, 2018, Hyderabad, India

ACM ISBN 978-1-4503-6398-3/18/02…$15.00
https://doi.org/10.1145/3172871.3172884

equivalence. Their techniques serve specific purposes and are centred on particular types of documents. They largely benefit from the structure of the documents. The second category of related work goes much finer level than shallow parsing. The method proposed by S.Ghaisas et al. [14] focused on retrieving rule intents from requirement documents by matching them against patterns made up of sequence of Part Of Speech (POS) tags, key words and their repetitions denoted by wild characters like '*', '+' etc. Other than the above two categories, some other miscellaneous works that address the problem of extracting knowledge from documents using case grammar [23], genetic algorithm [5] and machine learning [28].

All the above techniques basically focus on specific kind/class of documents, taking advantage of the structure and format of the document, while most of the techniques make use of predefined templates. The first category of the related work uses shallow parser with predefined templates, making these techniques tightly coupled and dependent on the document structure. The second category of work includes predefined templates using POS tag sequence, making it vulnerable to the noise. The POS tag sequence can produce incorrect rule intents due to noisy word sequences.

3 MOTIVATING EXAMPLE

In this section, we use the KYC (Know Your Customer) example to illustrate business rule intent extraction and identifying the relations among them.

```
All cross-border wire transfers must be accompanied by accurate and meaningful originator information.
```

The first task in our approach is to automatically extract relevant atomic facts from the *rule sentence*. The facts extracted from the rule sentences are shown below.

$f_1 : \mathit{isCrossBorder}(\texttt{wire transfer})$

$f_2 : \mathit{isAccurate}(\texttt{originator information})$

$f_3 : \mathit{doAccompany}(\texttt{wire transfer, originator information})$

$f_4 : \mathit{isMeaningful}(\texttt{originator information})$

Table 1: Rule intents of the sentence from earlier example

The above example document fragment has total 4 rule intents. The next step in the workflow is identifying the relations among the extracted rule intents. The relation among rule intents extracted for the example are shown below:

$$Rule_1 : f_1 \rightarrow f_2 \wedge f_3 \wedge f_4$$

Subsequently, the extracted rules (rule intents and relations between them) will be converted to SBVR models, which is a machine manipulatable format, to perform analysis for verification & validation [10]. The SBVR in Structured English (SE) for the example rule sentence is shown as:

transfer

originator information
General Concept: information

wire transfer
General Concept: transfer

wire transfer *is* cross-border
originator information *is accurate*
originator information *is meaningful*
wire transfer *is accompanied by* originator information

It is obligatory that if wire transfer *is* cross-border then wire transfer *is accompanied by* originator information
and originator information *is accurate*
and originator information *is meaningful*

4 APPROACH

In this section, we present and formally describe our approach.

The block diagram in Figure 1 illustrates our approach. In the first phase, Rule Sentence Extraction (RSE), we train using Trigram language model and segregate rule sentences from the noises. We use two language models; one is trained on rule sentences and the other is trained on noise sentences.

In the second phase, the Rule Intent Extraction (RIE), we extract *rule intents* from the rule sentences identified in the previous phase. We have defined a set of heuristic rules [8] to extract the atomic facts from a rule sentence using dependency tree parsing. Rule intents of the sentence presented in Section 3 is given in Table 1.

We have adopted Maximum Entropy (Max-Ent) classifier [13] to mine logical relations between rule intents from a rule sentence. Max-Ent classifier proved to be prolific in '*extracting relations among named entities*', a popular NLP research problem [16, 18, 22]. We have considered only five relations AND, OR, IMPLICATION, ARGUMENT and NULL for classification. The AND, OR, IMPLICATION represents the logical relations and ARGUMENT relation represents a *rule intent* being parameter to another. The NULL relation indicates that the two *rule intents* are independent with no relation among them. In our experiments, we have trained the Max-Ent classifier with 30 features. The pairwise relations for the rule intents given in Table 1 are shown below:

$$\begin{aligned} IR_1 &: f_2 \wedge f_3, \\ IR_2 &: f_3 \wedge f_4, \\ IR_3 &: f_4 \wedge f_2, \\ IR_4 &: f_1 \rightarrow f_3 \end{aligned}$$

To further analyse the rule intents, we form a *rule relation graph* for each rule sentence where a *rule intent* is considered as *node* and the relation between pair of rule intents is a labelled *edge*. In first step of graph pruning, the ARGUMENT edge is considered for merging, where in turn we copy every incoming and outgoing edge of the first

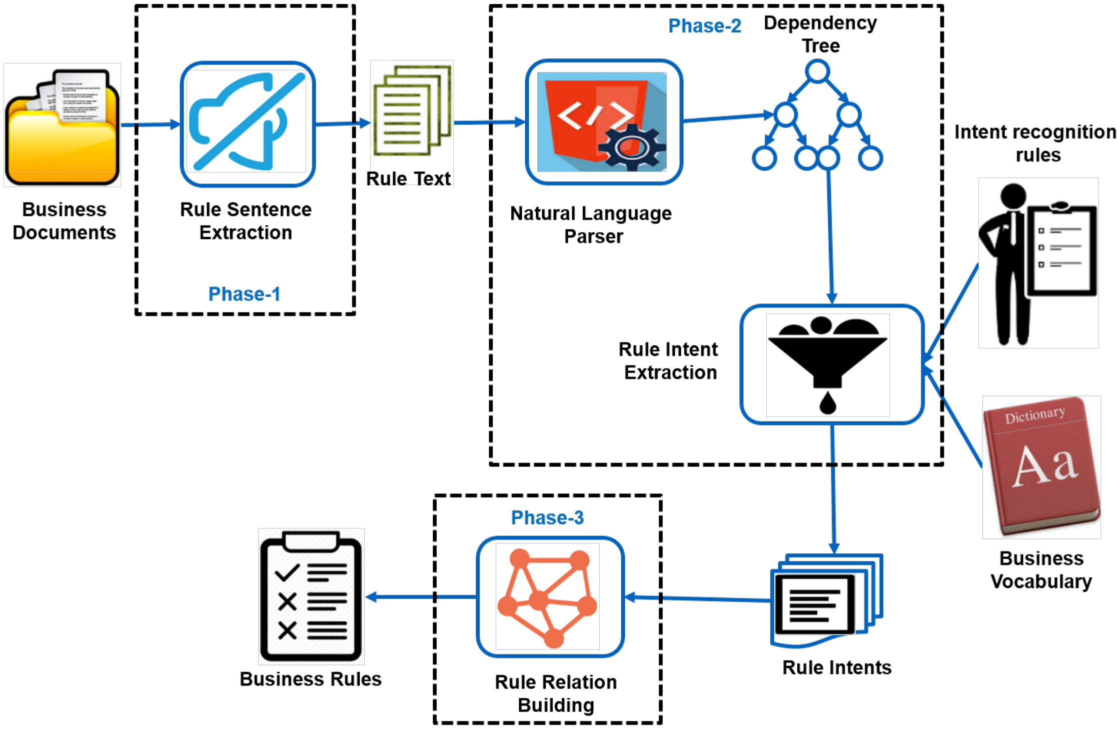

Figure 1: Block diagram of our approach.

node to latter and delete the former. In second step, we merge the *satellite nodes* (node that has only one neighbour). While merging *satellite node* to its neighbour, we change the label of the node to the string rule intent of satellite node + edge label + rule intent of this neighbour node, where '+' indicates string concatenation. We do not merge any node with 'IMPLICATION' edge at this stage. In the third step, we resolve any triangle relation dependence in the graph. If all the edges of the triangle are of same type, we merge them any order. Otherwise, we start with the edge which differs. In any practice, a triangle graph with three different edges is not a possible scenario. Ultimately we merge all *satellite nodes* to create a new node after reduction. Using this reduced graph and SimpleNLG we generate the output as shown in Motivating Example Section.

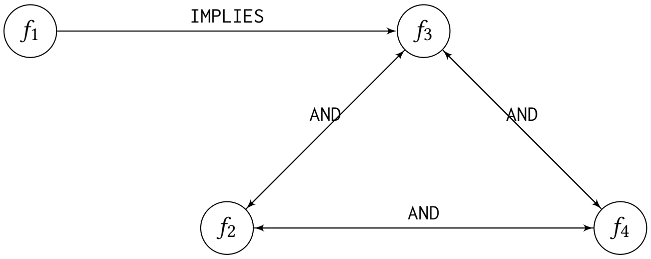

Figure 2: The rule relation graph shown for the motivating example

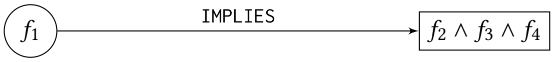

Figure 3: The reduced dependency tree of rs_1

5 EXPERIMENT AND RESULTS

In this section, we discuss the technologies used in the implementation of our prototype system and the experimental studies conducted.

5.1 Implementation

We have used Stanford coreNLP parser [12] for parsing the rule sentences, POS tagging, and for creating the dependency tree. The RIE phase is a completely home grown product. OpenNLP MaxEnt [7] has been used for relation extraction. We have built a prototype tool to integrate the above phases, while using SimpleNLG API to generate SBVR Structured English (SE). The experiments have been performed with the prototype on Windows 7 machine with CORE*i*5 processor and 2 GB RAM.

5.2 Experimental Study

We have evaluated our prototype tool with two sets of subjects: 1) Know Your Customer (KYC) document consists of guidelines for banks about collecting various details of their customer in conducting their business [3]. 2) The requirements for a fictitious car rental company EU-RentACar [1]. The contents of this document has noise free rule sentences and hence we have not used it for noise elimination by language model. The document had 71 rule sentences.

In evaluating language model for RSE, we measured the efficacy on the basis of three parameters 1) *recall*, 2) *precision*, and 3) *accuracy*. The *recall* is defined as the ratio of number of True Positive Instances (TPI) detected to the number of Actual Positive Instances (API) is shown in 1.

$$recall = \frac{|TPI|}{|API|} \tag{1}$$

The *precision* is defined as ratio of number of TPI to the total number of TPI and False Positive Instances (FPI) is shown in 2.

$$precision = \frac{|TPI|}{|TPI| + |FPI|} \tag{2}$$

The *accuracy* is measured as ratio of Total Instances matches to that of Actual Instances is shown in 3.

$$accuracy = \frac{|TPI| + |TNI|}{|API| + |ANI|} \tag{3}$$

Subject	Recall	Precision	Accuracy
KYC document	0.8415	0.9042	0.8047

Table 2: Result of Rule Sentence Extraction

Table 2 shows our results on language model using above measures. The performance of Rule Intent Extraction (RIE) is measured for recall and precision with equations (1) and (2), and mentioned in [8] and presented in Table 3.

Subject	Recall	Precision
KYC document simple sentences	0.8108	0.7834
EU-rent car Sentences	0.78	0.76
KYC document complex sentences	0.557	0.5114

Table 3: Result of Rule Intent Extraction

The performance of Relation Extractor is measured by discarding NULL relations. If a rule sentence has '*n*' *rule intents* then there would be nC_2 relations extracted having a majority NULL relations among them. In this study, we counted the number of relations other than NULL, as total extracted relations by classifier, the correct number relations, number of miss-interpreted relations. We measure the accuracy as the ratio of correct relation extracted by our system to all the extracted relations (excluding NULL) by the system. In case of Eu-RentACar case study, the sentences accuracy is 0.803. For simpler sentences from KYC document the accuracy is 0.708. The accuracy for complicated sentences is as bad as 0.65 (from KYC).

Currently, the experimentation with creating Structured English (SE) from relation graph is undergoing. We do not have any bench mark data or metric to measure the accuracy of the Graph Building and Rule Synthesis stage. Our current implementation produces SE very close to ideal output expected as shown in example in Section 3.

```
  ``In case of transactions carried out by a
walk-in customer, where the amount of transac- tion
is equal to or exceeds rupees fifty thousand,
whether conducted as a single transaction or several
transactions that appear to be connected, the
customer's identity and address should be verified.''
```

Figure 4: Example of a complex sentence having multiple clauses

5.3 Limitations

The NL sentences pose a big challenge as a sentence can be expressed in several ways. The NL sentences can be simple (with only one clause), or complex (having multiple clauses) as shown in Fig 4. The variations in way of interleaving clauses elicit our heuristic rules set to be enriched enough to take care of such sentences. Stronger benchmarks or metrics shall enable us to increase our accuracy in the future.

6 CONCLUSION

The larger objective of our work is to extract formal business rules (and processes) by analyzing requirements document, guidelines and do's and don't documents. To achieve the objective, we split the problem into multiple parts. We first decide on presence or absence of business rules in a sentence, and if present, we then extract the rule intents in the sentence followed by extracting pair-wise relations among them. We have successfully used the tri-gram language model to identify whether the English sentence contains a business rule or not. Depending upon the heuristics that are independent of the business domain, we extract the business intents from the sentences. We aim to experiment with many more documents to establish the adequacy of the heuristics, which we may parametrize in the long run. We have used maximum entropy classifier for detecting relation between rule intents. Our experiments with openly available documents and subsequent improvements in the methods yielded promising results in terms of the measures of precision, recall and accuracy. The above success gives us enough motivation to move further towards expressing the rules using a formal notation such SBVR.

REFERENCES

[1] [n. d.]. EU-RentACar case study. http://www.businessrulesgroup.org/first_paper/br01ad.htm. ([n. d.]). [Online; accessed 1-Oct-2017].

[2] [n. d.]. Object Management Group(OMG). http://www.omg.org. ([n. d.]). [Online; accessed 1-Oct-2017].

[3] [n. d.]. Reserve Bank of India (RBI), Master Circulars. https://rbi.org.in/scripts/BS_ViewMasCirculardetails.aspx/?id=9031. ([n. d.]). [Online; accessed 1-Oct-2017].

[4] [n. d.]. Semantics Of Business Vocabulary And Rules (SBVR). http://www.omg.org/spec/SBVR/. ([n. d.]). [Online; accessed 1-Oct-2017].

[5] J. Atkinson-Abutridy, C. Mellish, and S. Aitken. 2004. Combining Information Extraction with Genetic Algorithms for Text Mining. *IEEE Intelligent Systems* 19, 3 (May 2004), 22–30. https://doi.org/10.1109/MIS.2004.4

[6] Imran Sarwar Bajwa, Mark G. Lee, and Behzad Bordbar. 2011. SBVR Business Rules Generation from Natural Language Specification. In *AI for Business Agility, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-03, Stanford, California, USA, March 21-23, 2011*.

[7] Jason Baldridge, Tom Morton, and Gann Bierner. [n. d.]. The OpenNLP Maxent. http://maxent.sourceforge.net/about.html. ([n. d.]). [Online; accessed 1-Oct-2017].

[8] Abhidip Bhattacharyya, Pavan Kumar Chittimalli, and Ravindra Naik. 2017. An Approach to Mine Business Rule Intents from Domain-specific Documents. In *Innovations in Software Engineering Conference (ISEC)*. (to appear).

[9] Branimir K Boguraev. 2000. Towards finite-state analysis of lexical cohesion. In *Proceedings of the 3rd international conference on finite-state methods for NLP*.

[10] Pavan Kumar Chittimalli and Kritika Anand. 2016. Domain-independent method of detecting inconsistencies in SBVR-based business rules. In *Proceedings of the International Workshop on Formal Methods for Analysis of Business Systems@ASE 2016*. ACM, 9–16.

[11] Fabio Ciravegna. 2001. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2 (IJCAI’01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1251–1256.

[12] Marie-Catherine De Marneffe and Christopher D Manning. 2008. *Stanford typed dependencies manual*. Technical Report. Technical report, Stanford University.

[13] S. Della Pietra, V. Della Pietra, and J. Lafferty. 1997. Inducing features of random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19, 4 (Apr 1997), 380–393. https://doi.org/10.1109/34.588021

[14] S. Ghaisas, M. Motwani, and P.R. Anish. 2013. Detecting system use cases and validations from documents. In *Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on*. 568–573. https://doi.org/10.1109/ASE.2013.6693114

[15] Shalini Ghosh, Daniel Elenius, Wenchao Li, Patrick Lincoln, Natarajan Shankar, and Wilfried Steiner. 2014. ARSENAL: Automatically Extracting Requirements Specifications from Natural Language. *CoRR, abs/1403.3142* (2014).

[16] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL ’05)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 427–434. https://doi.org/10.3115/1219840.1219893

[17] Hai Huang. 1996. Business Rule Extraction from Legacy Code. In *Proceedings of the 20th Conference on Computer Software and Applications (COMPSAC ’96)*. IEEE Computer Society, Washington, DC, USA, 162–. http://dl.acm.org/citation.cfm?id=872750.873408

[18] Nanda Kambhatla. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions (ACLdemo ’04)*. Association for Computational Linguistics, Stroudsburg, PA, USA, Article 22. https://doi.org/10.3115/1219044.1219066

[19] Ion Muslea et al. 1999. Extraction patterns for information extraction tasks: A survey. In *The AAAI-99 Workshop on Machine Learning for Information Extraction*, Vol. 2.

[20] T. Nasukawa and T. Nagano. 2001. Text analysis and knowledge mining system. *IBM Systems Journal* 40, 4 (2001), 967–984. https://doi.org/10.1147/sj.404.0967

[21] Rahul Pandita, Xusheng Xiao, Hao Zhong, Tao Xie, Stephen Oney, and Amit Paradkar. 2012. Inferring Method Specifications from Natural Language API Descriptions. In *Proceedings of the 34th International Conference on Software Engineering (ICSE ’12)*. IEEE Press, Piscataway, NJ, USA, 815–825.

[22] Sachin Pawar, Pushpak Bhattacharyya, and Girish Keshav Palshikar. 2014. Semi-supervised Relation Extraction using EM Algorithm. https://www.cse.iitb.ac.in/~pb/papers/icon13-ie-em.pdf. (2014).

[23] Colette Rolland and Camille Ben Achour. 1998. Guiding the Construction of Textual Use Case Specifications. *Data Knowl. Eng.* 25, 1-2 (March 1998), 125–160. https://doi.org/10.1016/S0169-023X(97)86223-4

[24] A. Sinha, A. Paradkar, P. Kumanan, and B. Boguraev. 2009. A linguistic analysis engine for natural language use case description and its application to dependability analysis in industrial use cases. In *Dependable Systems Networks, 2009. DSN ’09. IEEE/IFIP International Conference on*. 327–336. https://doi.org/10.1109/DSN.2009.5270320

[25] H. M. Sneed. 2001. Extracting business logic from existing COBOL programs as a basis for redevelopment. In *Program Comprehension, 2001. IWPC 2001. Proceedings. 9th International Workshop on*. 167–175. https://doi.org/10.1109/WPC.2001.921728

[26] C. Wang, Y. Zhou, and J. Chen. 2008. Extracting Prime Business Rules from Large Legacy System. In *Computer Science and Software Engineering, 2008 International Conference on*, Vol. 2. 19–23. https://doi.org/10.1109/CSSE.2008.497

[27] Xusheng Xiao, Amit Paradkar, Suresh Thummalapenta, and Tao Xie. 2012. Automated Extraction of Security Policies from Natural-language Software Documents. In *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (FSE ’12)*. ACM, New York, NY, USA, Article 12, 11 pages. https://doi.org/10.1145/2393596.2393608

[28] Hao Zhong, Lu Zhang, Tao Xie, and Hong Mei. 2009. Inferring Resource Specifications from Natural Language API Documentation. In *Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering (ASE ’09)*. IEEE Computer Society, Washington, DC, USA, 307–318. https://doi.org/10.1109/ASE.2009.94