Comparison and Synergy Between Fact-Orientation and Relation Extraction for Domain Model Generation in Regulatory Compliance

Sagar Sunkle(✉), Deepali Kholkar, and Vinay Kulkarni

Tata Consultancy Services Research, 54B Hadapsar Industrial Estate, Pune 411013, India
{sagar.sunkle,deepali.kholkar,vinay.vkulkarni}@tcs.com
http://www.tcs.com/

Abstract. Modern enterprises need to treat regulatory compliance in a holistic and maximally automated manner, given the stakes and complexity involved. The ability to derive the models of regulations in a given domain from natural language texts is vital in such a treatment. Existing approaches automate regulatory rule extraction with a restricted use of domain models counting on the knowledge and efforts of domain experts. We present a semi-automated treatment of regulatory texts by automating in unison, the key steps in fact-orientation and relation extraction. In addition, we utilize the domain models in learning to identify rules from the text. The key benefit of our approach is that it can be applied to any legal text with a considerably reduced burden on domain experts. Early results are encouraging and pave the way for further explorations.

Keywords: Regulatory compliance · Rule extraction · Fact-orientation · Relation extraction · Natural language processing · Machine learning

1 Introduction

Modern enterprises face an unprecedented regulatory regime. Non-compliance often results in personal liability and risk for top management and to shareholders. Compliance management needs to be holistic in nature, because the same regulations may vary based on geography and over time and different units of an enterprise may have to be compliant with different regulations [12]. Equally importantly, it needs to be automated to the extent possible, so that compliance can be proved quickly, reliably, and maintained through time- and geography-specific variations.

With a formal representation of regulatory rules, it becomes possible for enterprises to check compliance with more reliable and thorough proofs/evidence [22]. Significant literature exists focusing on formal compliance checking [13,20]. But these solutions presuppose existence of rules.

I. Comyn-Wattiau et al. (Eds.): ER 2016, LNCS 9974, pp. 381–395, 2016.
DOI: 10.1007/978-3-319-46397-1_29

Several approaches use natural language processing (NLP) and machine learning (ML) techniques to extract the rules from legal NL texts in a semi-automated manner. Even so, complexity of legal texts leads most of these approaches to formulate targeted solutions. For instance, these approaches require the domain experts to identify structural arrangements like chapters, sections, paragraphs, etc., specific to legal texts [7], to simplify complex legal sentences and making them amenable to analyses [3,14,25], and to annotate legal texts to identify rules and various other aspects specific to given approaches [24].

Interestingly, many of these approaches either do not use a conceptual modeling method or it is done in a way that restricts its applicability to rule extraction. We believe that the lack of a conceptual modeling method targeted at obtaining domain-specific regulation model in a generic manner results in most of these approaches being (a) specific to a regulation and specific to a given natural language [2,16], and (b) not being able to scale due to continued reliance on the domain experts in various activities.

We argue in this paper that a more generic approach that uses a conceptual modeling method should drive the legal rule extraction. We present an automation of fact-orientation enhanced with relation extraction aimed at domain model generation. We consider the automated rule extraction to be a 3-step process consisting of (1) domain model generation, (2) rule identification using the domain model, and (3) rule authoring based on identified rules. Our specific contributions with respect to the first 2 steps are as follows:

1. We compare fact-orientation and relation extraction for their suitability toward regulatory domain model generation. Focusing on the commonality between them that both utilize *examples/instances* of core concepts, we provide an interactive synergistic treatment for the same.
2. We use the domain model and the dictionary obtained thus to identify rules in an automated manner.

We begin in Sect. 2 by reviewing related work and presenting the technical overview of our approach. The focus of this paper is on the first 2 steps, which are detailed in Sects. 3 and 4 respectively. The third step is similar to existing approaches but less reliant on domain experts. Results of an ongoing case study are discussed along with key issues in Sect. 5. We present future work and conclude the paper in Sect. 6.

Departing from most of the work in ontology learning as well as legal rule extraction, we aim to obtain a simple list of domain concepts and as many mentions of these concepts as possible. Once this list is available, we also use open information extraction techniques to obtain relations. We do not focus on any legal text-specific aspects such as segmentation, cross-referencing, identification of modalities, types of provisions and so on. Our idea is to only obtain a domain model and a dictionary with which to identify rules and defer the consideration of legal text-specific aspects till we obtain the logical specifications which provide appropriate level of abstraction at which to treat the aspects. Both the generation of logical specification and treatment of legal text-specific aspects are out of the scope of this paper.

2 Related Work and Technical Overview

Legal texts are unique from other NL texts mainly because legal texts are *prescriptive* in nature [25] and present details of modalities like permissions, obligations, and prohibitions.

2.1 Complexity of Legal Texts

Legal texts are different from other NL texts in the following ways [24]:

- Legal NL texts contain long sentences with complex clauses with a number of lists representing characteristics of norms and their applicability in specific conditions.
- They use cross references such that various details of a norm may be found in different chapters/sections/subsections.
- The changes to the definitions of norms over time in terms of exceptions, and variety of repeals and amendments are often placed in supplementary annexes.

At the same time, some studies have found that specific kinds of provisions follow typical sentential forms, at the least in a given regulation [15]. This peculiarity can be exploited as in some ML-driven approaches which use patterns of sentence structures in their learning techniques [16].

2.2 Current Approaches to Rule Extraction/Authoring

The complexity of legal NL texts has compelled the existing approaches in rule extraction research to come up with targeted solutions. In particular the NL-driven approaches use steps that include (a) identifying language patterns like *juridical natural language constructs* as in [7] and coming up with specialized parsing mechanisms for the same, (b) manually transforming statements in legal NL texts to simplified form such as *restricted natural language statements* as in [3] for easier processing, and (c) utilizing structural characteristics of legal NL texts by identifying sections of text at varying granularity from phrases to chapters and annotating cross references as in [25].

The approaches that do refer to a conceptual model use it in a restricted sense. For instance, the approach in [25] uses a *conceptual (meta) model of deontic concepts*. This model represents legally oriented concepts such as an actor, a right, an obligation, an exception, and so on. It is used as a basis of the semantic annotations, but is not a core artifact that drives the rule extraction process. The approach in [11] is similar to the approach in [25] in that it uses what it refers to as a *governance extraction model*, which again focuses on legal concepts alone rather than business domain concepts.

Similar to NL-driven approaches, ML-driven approaches too focus on classifying the sentences/paragraphs from the legal texts into different kinds of provisions without informing the features of the classifiers with a representation of domain model. For instance, approaches like [2] implemented in the context of

Norme in Rete and [16] from the project E-POWER use classifiers based on word frequency with a training set labeled by the domain expert. When such features are used to train classifiers, the reason of classification remains hard to understand and improve upon as demonstrated in [15,18].

Below, we describe how we enable generating a domain model of regulations and a dictionary and how both are used in learning rules from the legal text. We choose fact-orientation as a domain modeling method. This choice is influenced to some extent by our previous work. We used a realization of fact-orientation known as Semantics of Business Vocabulary and Rules (SBVR) to manually create vocabulary of regulations which we use to generate NL explanations of proofs of (non-) compliance [22]. Our previous work was in the context of Indian Know Your Customer (KYC)[1] regulations. KYC regulations aim to prevent money laundering (ML) and financing of terrorism (FT). They require the financial institutions like banks to take new customers following strict identity and address checks while transactions of existing customers need to be monitored based on their risk profiles. We use running examples from KYC henceforth. Note that we use the principles of fact-orientation without aiming specifically to generate SBVR formulations of the regulations.

2.3 Technical Overview

Our approach is illustrated in Fig. 1. As mentioned in Sect. 1, we consider the semi-automated rule extraction to be a 3-step process.

We make no assumption about the legal text like NL-driven approaches as described above, rather we use one peculiarity of legal texts that they contain definitions of key concepts referred in the text. We use fact-orientation as an overall modeling method. In step (1), we implement relation extraction (RE) as an adapted version of an RE technique called *Dual Iterative Pattern Relation*

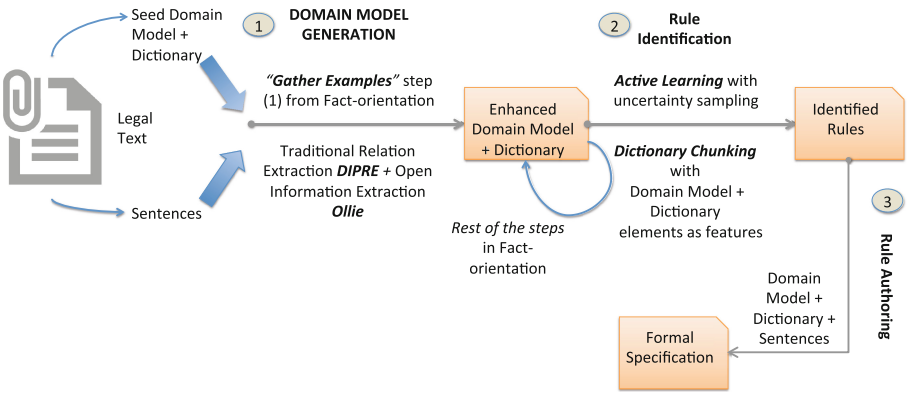

Fig. 1. Technical overview

[1] https://rbi.org.in/scripts/BS_ViewMasCirculardetails.aspx?id=9848.

Extraction (DIPRE) [4]. We also use a recent cousin of RE called open information extraction (open IE) implemented using Ollie tool [17]. Step ① uses sentences from legal NL text along with a seed domain model created from the definitions and an initial dictionary.

Once we obtain an enhanced domain model and dictionary, in step ② we use them to inform the features of *active learning*, a semi-supervised ML technique to classify legal NL text sentences into rules and non-rules rather than variety of provisions. Once the classifier is trained with the required precision and recall, in step ③ we can use a rule authoring environment along with all the artifacts so far obtained to author the rules.

In the next section, we first review each of fact-orientation and relation extraction and then describe how they are used in step ① of our approach.

3 Fact-Orientation and Relation Extraction

While our familiarity with fact-orientation (FO) is one reason, the other more substantial reason is the focus in FO on enabling the domain-expert to partake in domain modeling.

3.1 Role of Fact-Orientation in Domain Model Generation

FO bears many advantages over entity relationship and object-orientation when it comes to modeling the domain at conceptual level as enlisted below:

1. All ground assertions of interest are non-decomposable *facts* which are instances of *fact types*. Fact types can be unary to n-ary. This attribute free approach facilitates advantages such as semantic stability, an analysis of which the interested reader is invited to refer in [9].
2. FO models are validated by domain experts in two ways: verbalization [6] and population. It means that verbalization of fact types has to be agreed upon by the domain expert using populations of the same from the NL texts. This makes FO apt in the context of legal NL texts.
3. Being more generic, FO models can be transformed to other modeling formats if required. For instance, the freeware NORMA tool enables exporting the models in object role model specification to many other formats including relational views and even Datalog [5].

The first step of the conceptual schema design procedure (CSDP) prevalent in FO is that of *(Gather and) Transform familiar information examples into elementary facts (and apply quality checks)*. The second step applies population check, meaning that the fact types indeed are valid with respect to examples from NL texts. Step 3 to 7 refine the concept types and add constraints of various kinds.

The first step essentially abstracts from examples to create fact types. In general, the process of collecting population examples and abstracting from them is manual. This is where we make use of relation extraction as described next.

3.2 Role of Relation Extraction in Domain Model Generation

Relation extraction (RE) or traditional information extraction (IE) is the task of discovering assertions of a particular relation between two or more concepts in NL texts [1]. Supervised approaches to RE require completely labeled training sets of sentences and unsupervised methods use trained named entity taggers to identify concepts thereby being able to identify relations between more prevalent pairs of concepts like persons and locations [1].

Since we wish to be able to identify relations specific to a domain, we are interested in semi-supervised approaches which learn from a small set of tagged seed instances or few hand-crafted extraction patterns. Given a known pair of concepts and their handful of mentions, semi-supervised RE techniques like Dual Iterative Pattern Relation Extraction (DIPRE) [4] enable finding the rest of the mentions.

At this point, we bring to notice that FO and RE operate in opposite directions. Figure 2 illustrates this with concept types Bank, Customer, and Document and population examples containing mentions of these types from KYC text. While the first step in FO uses population examples to abstract to concept types, RE enables finding all mentions of given related concept types and their seed instances. For instance, given the known concept types Customer and Document related through *submits* relation and a handful of mentions so related as in Fig. 2, DIPRE can find other mentions with patterns induced from sentences containing the known mentions.

To automate the first step of FO, we still need to take care of two more aspects:

1. We need a way to find unknown concept types.
2. We need a way to find relations between all concept types found so far.

We describe in the following how we automate these two aspects in sync with RE so that by the end of the processing the text for mentions and new concept types, we generate a basic domain model and a dictionary that maps all concepts to their mentions throughout the text thus automating the first step of FO.

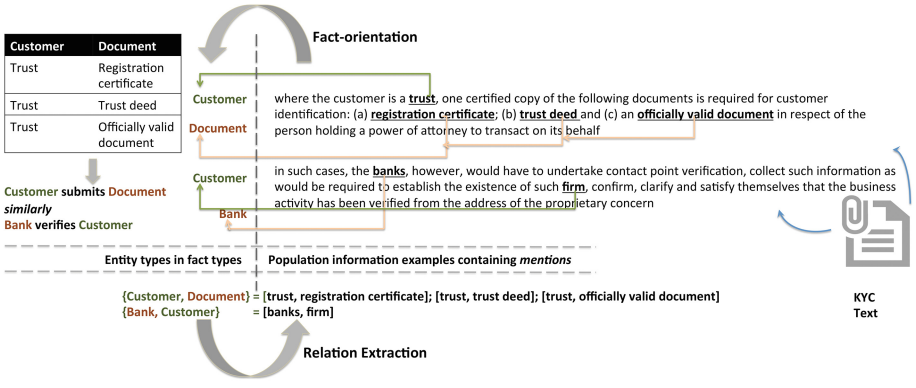

Fig. 2. Comparative view of fact-orientation and relation extraction

3.3 Our Approach for Domain Model Generation

We use the LingPipe[2] toolkit for processing text, which implements several algorithms from computational linguistics.

Finding Unknown Mentions of Known Concept Types. In order to identify known mentions in the text, we use an implementation of approximate dictionary chunker from [23] implemented in LingPipe. This chunker produces chunks based on weighted edit distance of strings from entries of a dictionary in which we store known pairs of concept types and their mentions. In order to seed this dictionary, we refer to the *definitions* section of the legal text, in our case KYC. This chunker forms an important module of our system, since it is used in both finding unknown concept types and finding relations between all concept types. It is also used in creating a specialized feature representation in learning to identify rules in legal text as explained later in Sect. 4.

Finding Unknown Mentions of Unknown Concept Types. In order to find concept types that could be part of the domain model but not yet known, we again use mentions of concepts that we have so far found. We use a hypothesis known as *distributional semantics* [10], which suggests that counting the contexts that two words share improves the chance of correctly guessing whether they express the same meaning, in other words, semantically similar expressions occur in similar contexts (Fig. 3).

We cluster the contexts, i.e., n characters to the left and right of mentions of each concept type so far known and then cluster these to suggest to the domain expert, what looks like other possible mentions. This is illustrated in Fig. 4.

The domain expert either adds to the dictionary, a new mention of a known concept type as in the case of (A) in Fig. 4 or as in the case of (B) has the option to add a new concept type along with the mention(s), if she recognizes that the mention(s) refers to different concept type not in the current set of

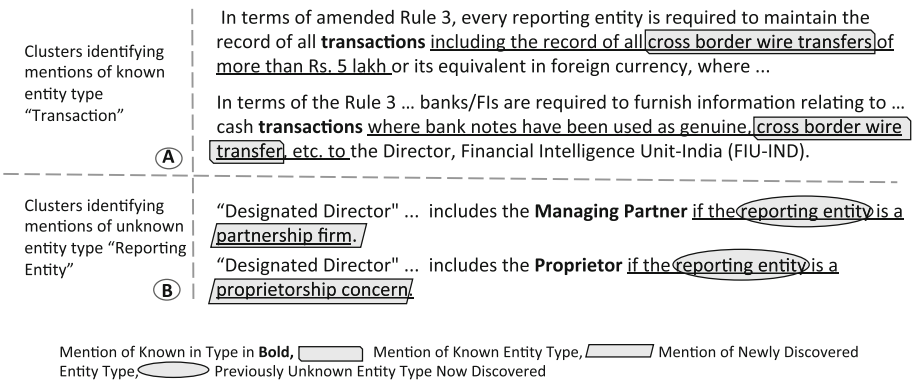

Fig. 3. Clustering of contexts around mentions

[2] http://alias-i.com/lingpipe/index.html.

known concept types. In (A), clustering the contexts of mentions of the concept type Transaction reveals a mention *cross border wire transfer* which the domain expert deems to be of the same concept type Transaction. In (B), clustering the contexts of mentions of the concept type Designated Officer reveals the mentions *partnership firm* and *proprietorship concern*, which the domain expert adds along with previously unknown type Reporting Entity of which they are mentions.

Finding Relations Between Known Concept Types. In order to find relations between concept types identified in the domain model, we use Ollie[3], an open IE implementation. Open IE differs from traditional RE in that open IE systems extract a diverse set of relational tuples without requiring any relation-specific human input. Whereas traditional RE like DIPRE requires concept type pairs representing a relation as shown in Fig. 2, open IE works where target relations are not known in advance. Open IE systems identify relation phrases, i.e., phrases that denote relations in English sentences.

The implementation we use, Ollie, is able to avoid uninformative and incoherent extractions. As an example of the former, whereas other open IE systems incorrectly extract *made(Faust,deal)* from the sentence "Faust made a deal with the devil.", Ollie (as successor of Reverb [8]) correctly extracts *made a deal with(Faust, the devil)*. Ollie is better than other open IE systems also because it extracts relations mediated by nouns, adjectives, and other verbal structures and a context-analysis step increases precision of relations extracted by including contextual information from the sentence in the extractions [17].

We input the concept types and mentions found so far to Ollie along with all the sentences in the legal text. Whenever existing mentions match with phrases that Ollie has found to be in relation, that relation is considered to exist between the concept types of the mentions.

Overall Approach to Domain Model Generation. We present the use of RE, context clustering, and open IE in Algorithm 1. The seed domain model and dictionary are obtained from *definitions* section of the regulation. In most of financial services regulations that we have encountered apart from KYC, we have found that definitions of key concept types are provided along with their subtypes and terms with which they are referred to in the text.

The procedure searchMentions implements semi-supervised RE. We implemented an adaptation of DIPRE wherein instead of looking for mentions on the web, we search the sentences for mentions to induce patterns using induce-Patterns. Since compared to web, legal text is very small, the number of patterns that can be induced is also small. Whenever searchMentions finds possible mentions in the same relation via apply-re-patterns, we ask the domain expert to verify if the mentions are indeed in relation.

The procedure contextClustering implements clustering of contexts via apply-Clustering around mentions of all known concepts. In our experiments, we use length of 80 characters when capturing contexts via computeContexts to make single link clusters. Examples shown in Fig. 4 show results where the underlined

[3] https://github.com/knowitall/ollie.

text is a context under consideration. We involve the domain expert to verify and add to domain model and dictionary only the concepts and mentions she knows to be relevant.

Finally, the procedure searchRelations implements open IE over sentences of the text via runOpenIE. In our case, this is a call to Java wrapper around Ollie. If mentions of two different concepts are found in the subject and object of IE relation, then that relation is taken to exist between the two concepts.

Algorithm 1. Domain Model Generation

Input: Text, Seed Domain Model (DM), Dictionary of Mentions (DoM)
Output: DM, DoM

```
1  sentences ← sentenceDetection(text)
2  procedure searchMentions(Sentences sents, DM dm, DOM dom)
3      for each conceptPair cp in dm do
4          mentionPair ← mentionsOfConcept(cp, dom)
5          while apply-re-pattern(sents, re-Patterns) > 0 do
6              re-Patterns ← inducePatterns(sents, mentionPair)
7              de-Input ← apply-re-pattern(sents, re-Patterns)
8              dom ← dom + de-Input
10     return dom ;
11 procedure contextClustering(Sentences sents, DM dm, DOM dom)
12     for each concept cn in dm do
13         for each conceptMention cm of cn in dom do
14             mentionContextList ← computeContexts(sents, cm)
15             de-Input ← applyClustering(mentionContextList)
16             dm ← dm + de-Input
17             dom ← dom + de-Input
19     return dm, dom ;
20 procedure searchRelations(Sentences sents, DM dm, DOM dom)
21     for each sent in sents do
22         open-IE-Relation ← runOpenIE(sent)
23         for each conceptPair cp(cn1, cn2) in dm do
24             for each mentionPair mp of cp in dom do
25                 if open-IE-Relation.subject contains mp.mention1 and
                    open-IE-Relation.object contains mp.mention2 then
26                     dm ← open-IE-Relation.relation(cn1, cn2)
28     return dm ;
29 while dm.hasChanged() or dom.hasChanged() do
30     dom ← searchMentions(sents, dm, dom)
31     dm, dom ← contextClustering(sents, dm, dom)
32 dm ← searchRelations(sents, dm, dom)
33 return dm, dom
```

At this juncture, the domain model does not contain constraints or sub-types. We revert back to fact-orientation and follow step 2 to 7. Step 2 of FO applies population check. Since we take domain experts' input on each of RE, clustering, and IE stages in terms of mentions and concepts, step 2 of FO is implicitly supported in our approach. We provide a view to the domain expert into the sentences of the legal text, where a pair of concept is under consideration for combining or sub-typing. Similarly, the domain expert refers to the occurrences of concepts and their mentions in the text via specialized view to add and refine constraints.

4 Regulatory Rule Identification

Active Learning. To automate manual rule identification, we use semi-supervised active learning. Active learning techniques can learn from very less number of labeled sentences, by querying the domain expert on possible classes of a sentence. In our case, the classes are rule sentences and non-rule sentences.

The process of active learning involves taking a small set of labeled examples (sentences) as input, as well as a larger set of unlabeled examples, and generating a classifier and a relatively small set of newly labeled data. The learning process aims at keeping the domain expert annotation effort to a minimum, only asking for advice where the training utility of the result of such a query is high [21].

Representing Features based on Domain Model and Dictionary. We intend to make the use of the domain model and the dictionary mimic the way a domain expert actually identifies regulations in the text. We use a specialized FeatureExtractor[4] from LingPipe called ChunkerFeatureExtractor. A feature extractor provides a method of converting generic input objects into feature vectors. A ChunkerFeatureExtractor implements a feature extractor for character sequences based on a specified chunker. Here, we utilize the same approximate dictionary chunker we referred to in Sect. 3.3. This arrangement helps us in uniquely representing features in terms of concepts and their mentions from the domain model and the dictionary respectively.

To implement an active learner for rule identification, we use LogisticRegressionClassifier from LingPipe. It is a scored classifier that provides conditional probability classifications of input objects. It uses an underlying logistic regression model and feature extractor which in our case is the ChunkerFeatureExtractor. We implement the prototypical active learning algorithm from [19].

5 Results and Discussion

We present the results of applying our approach from Algorithm 1 to KYC text as well as applying active learning to the task of identifying rules in KYC below along with the discussion of key pointers.

[4] http://alias-i.com/lingpipe/docs/api/com/aliasi/util/FeatureExtractor.html.

Applying RE, Clustering, and IE to KYC Text. We copy pasted the text of KYC from the link shared earlier. We use LingPipe's IndoEuropeanSentenceModel to split the text into sentences. We obtained 525 sentences. From the *definition* section, we obtained 4 concepts.

We get 4 more concepts and their mentions through contextual clustering. Table 1 shows the mentions from definitions (#2) and from the application of RE and clustering (#3). We only specify 5 mentions of concepts in the table for the want of space. The column #4 indicates no. of sentences out of 525 where mentions of concepts were found.

Figure 4 shows some of the relations discovered between concepts based on the mentions that actually occurred in the corresponding sentences using Ollie.

Applying FO to Domain Model and Dictionary. Fig. 5 shows a fact-oriented model of KYC regulations. We used NORMA[5] tool to draw the object role model displayed on the right in Fig. 5.

Figure 6 shows the verbalization of concept Bank as well as fact types verifies and submitsForVerification generated automatically from the model. The relations or the fact types were adapted in consultation with the domain expert from initial set of relations from definitions sections and relations obtained from IE, a few of which were shown in Fig. 4.

Using the Dictionary with the Active Learner. Out of 525 sentences, we use 300 sentences to teach the active learner in a 10-fold cross validation setup with 225 sentences to test the learner. We annotated 10 sentences as denoting rules and 5 sentences as denoting non-rules before starting the learning sessions.

To identify how the use of domain model and dictionary affect recall and precision, we show the feature representations (a) when dictionary is used, i.e., when the learner is informed, (b) when instead of the dictionary, only a feature extractor based on n-gram tokenizer is used, i.e., the learner is uninformed, and (c) when dictionary is used along with a feature extractor based on n-gram tokenizer, i.e., the learner is semi-informed.

[individuals] **CUSTOMER** < may be categorised as > **RISK CATEGORY** [low]
[individuals/entities] **CUSTOMER** < be regulated by > **REPORTING ENTITY** [insurance companies]
[officially valid document in respect of] **DOCUMENT** < may be required by > **CUSTOMER** [fi]
[document] **DOCUMENT** < would be submitted to > **BANK** [bank]
[banks] **BANK** < should closely monitor especially accounts of multi-level marketing (mlm) companies in > **CUSTOMER**[firms]
[banks] **BANK** < should obtain ovd for > **DOCUMENT** [proof of address and identity of the relative]
[banks] **BANK** < to submit feedback on > **TRANSACTION** [deposits]
[banks] **BANK** < should categorise their customers into > **RISK CATEGORY** [low]
[exchanges] **TRANSACTION** < be reported by > **BANK** [banks]

Fig. 4. Relations found with IE; mentions in [ ] brackets, concepts in **bold**, relations in <>

We used InteractionFeatureExtractor from LingPipe which produces interaction features between two feature extractors to create the combined extractor

[5] https://www.ormfoundation.org/files/folders/norma_the_software/default.aspx.

Table 1. KYC concepts and mentions; #1: concept present in seed domain model, #2: no. of seed mentions, #3: no. of total mentions found with RE and IE, #4: no. of sentences where concept mention occurs

Sr.	Concepts	#1	#2	#3	Mentions	#4
1	Reporting entity	N	0	9	All India financial institutions, local area banks, primary (urban) co-operative banks, scheduled commercial banks, state and central co-operative banks	14
2	Bank	N	0	1	Bank	257
3	Account	N	0	2	Client accounts, small accounts	5
4	Customer	Y	12	30	Foreign portfolio investors, politically exposed persons, artificial juridical person, association of persons, body of individuals	123
5	Document	Y	33	51	Certificate of incorporation, certificate/licence issued by the municipal authorities under shop and establishment act, complete income tax return, licence/certificate of practice issued in the name of the proprietary concern by any professional body incorporated under a statute	128
6	Transaction	Y	15	17	Creating a legal person, cross-border wire transfer, deposits, withdrawal, fiduciary relationship	111
7	Risk category	N	0	3	High, low, medium	23
8	Designated director	Y	4	4	Managing partner, managing director, managing trustee, whole-time director	2

for case c. The value of an interaction feature is the product of the values of the individual features.

We found that when we used domain model and dictionary exclusively to represent features, we obtained consistently higher recall than the other two extractors. On the other hand, using n-grams of lengths 3 to 5 exclusively, we obtained higher precision than the other two extractors. Recall represents retrieval coverage. Because the dictionary captures mentions of concepts, the recall or coverage of dictionary extractor is comprehensive.

In our case, the extractor based on n-grams consistently has higher precision, which measures retrieval specificity but has correspondingly lower recall than the dictionary extractor. These results may be attributed to capturing concepts via dictionary of mentions against n-grams which do not make sense (*ther, nci,*

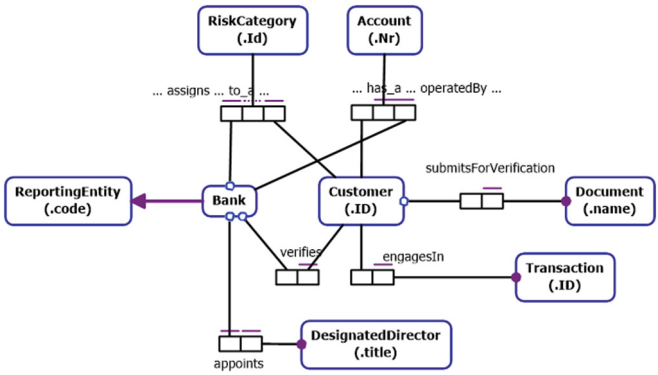

Fig. 5. KYC domain model using fact-orientation

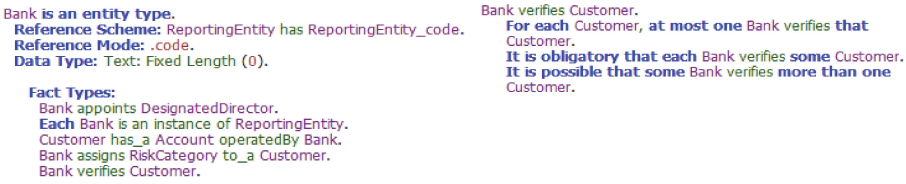

Fig. 6. KYC verbalization using fact-orientation

nanci, cash, and so on). The combined extractor achieves recall of dictionary extractor and reaches the precision of n-gram extractor. Our results indicate that for rule identification, a semi-informed approach performs better.

Summary. Our initial foray using FO, RE, context clustering, and IE indicates that we can do away with simplified paraphrasing of legal NL texts and other annotations used in existing approaches while reliably generating a domain model and a dictionary. The domain model and the dictionary can be used in unison with extractors focused on precision to achieve a better combination of precision and recall.

6 Future Work and Conclusion

We presented an approach and an algorithm to domain model generation using fact-orientation (FO) and flavors of relation extraction (RE) based on how each treats mentions of concepts in NL texts. In our ongoing experiments, we are trying to create an integrated development environment that shows views of FO, RE, context clustering, and open information extraction (IE) to the domain expert. Further work also includes giving more immersive treatment to rule authoring whereby knowledge latched so far can be utilized by the domain expert.

We are also experimenting with MiFID II[6] regulations, which presents more than 5000 sentences.

Our approach has shown to be generic in the sense that no regulation-specific structuring, simplification, or annotation is needed to capture the domain model and the rules. Also compared to existing approaches, it has the potential to scale well.

References

1. Bach, N., Badaskar, S.: A review of relation extraction. Lit. Rev. Lang. Stat. II (2007)
2. Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C.: Automatic semantics extraction in law documents. In: Sartor, G. (ed.) ICAIL, Italy, 6–11 June 2005, pp. 133–140. ACM (2015). http://doi.acm.org/10.1145/1165485
3. Breaux, T.D., Antón, A.I.: Deriving semantic models from privacy policies. In: 6th Policy Workshop, Sweden, pp. 67–76. IEEE Computer Society (2005)
4. Brin, S.: Extracting patterns and relations from the world wide web. In: Atzeni, P., Mendelzon, A., Mecca, G. (eds.) WebDB 1998. LNCS, vol. 1590, pp. 172–183. Springer, Heidelberg (1999). doi:10.1007/10704656_11
5. Curland, M., Halpin, T.: The NORMA software tool for ORM 2. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. LNBIP, vol. 72, pp. 190–204. Springer, Heidelberg (2011). doi:10.1007/978-3-642-17722-4_14
6. Curland, M., Halpin, T.: Enhanced verbalization of ORM models. In: Herrero, P., Panetto, H., Meersman, R., Dillon, T. (eds.) OTM 2012. LNCS, vol. 7567, pp. 399–408. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33618-8_54
7. van Engers, T.M., van Gog, R., Sayah, K.: A case study on automated norm extraction. In: Gordon, T. (ed.) The Seventeenth Annual Conference on Legal Knowledge and Information Systems, JURIX 2004, pp. 49–58. Frontiers in Artificial Intelligence and Applications. IOS Press, Amsterdam (2004)
8. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 1535–1545. ACL, Stroudsburg (2011)
9. Halpin, T.A.: Fact-orientation and conceptual logic. In: Proceedings EDOC 2011, Finland, pp. 14–19. IEEE Computer Society (2011)
10. Harris, Z.S.: Mathematical Structures of Language. Wiley, New York (1968)
11. Hassan, W., Logrippo, L.: Governance requirements extraction model for legal compliance validation. In: RELAW 2009, USA, pp. 7–12 (2009)
12. Kaminski, P., Robu, K.: Compliance and control 2.0: emerging best practice model. McKinsey Working Papers on Risk 33, October 2015
13. Kharbili, M.E., de Medeiros, A.K.A., Stein, S., van der Aalst, W.M.P.: Business process compliance checking: current state and future challenges. In: MobIS. LNI, vol. 141, pp. 107–113. GI (2008)
14. Kiyavitskaya, N., Zeni, N., Breaux, T.D., Antón, A.I., Cordy, J.R., Mich, L., Mylopoulos, J.: Automating the extraction of rights and obligations for regulatory compliance. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 154–168. Springer, Heidelberg (2008). doi:10.1007/978-3-540-87877-3_13

[6] http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32014L0065&from=EN.

15. de Maat, E., Krabben, K., Winkels, R.: Machine learning versus knowledge based classification of legal texts. In: Proceedings of JURIX 2010, pp. 87–96. IOS Press, Amsterdam (2010)
16. de Maat, E., Winkels, R.: Automatic classification of sentences in Dutch laws. In: Proceedings JURIX 2008, pp. 207–216. IOS Press, Amsterdam (2008)
17. Mausam, S., M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: Proceedings of EMNLP-CONLL (2012)
18. Moens, M.F., Boiy, E., Palau, R.M., Reed, C.: Automatic detection of arguments in legal texts. In: ICAIL 2007, pp. 225–230. ACM, New York (2007)
19. Olsson, F.: A literature survey of active machine learning in the context of natural language processing. Technical report, Kista, Sweden, April 2009
20. Racz, N., Weippl, E.R., Bonazzi, R.: IT governance, risk & compliance (GRC) status quo and integration: an explorative industry case study. In: SERVICES 2011, USA, 4–9 July 2011, pp. 429–436. IEEE Computer Society (2011)
21. Settles, B.: Active learning literature survey. Computer Sciences Technical report 1648, University of Wisconsin-Madison (2009)
22. Sunkle, S., Kholkar, D., Kulkarni, V.: Explanation of proofs of regulatory (non-)compliance using semantic vocabularies. In: Bassiliades, N., Gottlob, G., Sadri, F., Paschke, A., Roman, D. (eds.) RuleML 2015. LNCS, vol. 9202, pp. 388–403. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21542-6_25
23. Tsuruoka, Y., Tsujii, J.: Boosting precision and recall of dictionary-based protein name recognition. In: Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine, BioMed 2003, vol. 13, pp. 41–48. ACL, Stroudsburg (2003)
24. Wyner, A., Peters, W.: On rule extraction from regulations. In: Atkinson, K. (ed.) Legal Knowledge and Information Systems - JURIX, Vienna, Austria. Frontiers in Artificial Intelligence and Applications, vol. 235, pp. 113–122. IOS Press (2011). http://www.booksonline.iospress.nl/Content/View.aspx?piid=26386
25. Zeni, N., Kiyavitskaya, N., Mich, L., Cordy, J.R., Mylopoulos, J.: GaiusT supporting the extraction of rights and obligations for regulatory compliance. Requir. Eng. **20**(1), 1–22 (2015)