Norm Conflict Identification using Deep Learning

João Paulo Aires and Felipe Meneguzzi

School of Computer Science
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil

joao.aires.001@acad.pucrs.br
felipe.meneguzzi@pucrs.br

Abstract Contracts represent agreements between two or more parties formally in the form of deontic statements or norms within their clauses. If not carefully designed, such conflicts may invalidate an entire contract, and thus human reviewers invest great effort to write conflict-free contracts that, for complex and long contracts, can be time consuming and error-prone. In this work, we develop an approach to automate the identification of potential conflicts between norms in contracts. We build a two-phase approach that uses traditional machine learning together with deep learning to extract and compare norms in order to identify conflicts between them. Using a manually annotated set of conflicts as train and test set, our approach obtains 85% accuracy, establishing a new state-of-the art.

Keywords: norms, contracts, deep learning, natural language

1 Introduction

Regulations are often applied to social members in a society in order to minimize conflicting behaviors [18]. Such regulations also known as social norms, define expected behaviors accepted for society members and that ensure that individuals act according to a socially acceptable behavior. Besides regulating entire societies, social norms are also used to regulate interactions in smaller groups, and are often present in social relationships involving agreements over products and services. A common way to formalize a set of norms applied to an agreement is through contracts. In human societies, contracts are semi-structured documents written in natural language, which are used in almost every existing formal agreement. Contracts define the parties involved in the agreement, their relations, and the behavior expected of each party within clauses. When written in natural language, contracts may use imprecise and possibly vague language to define parties, obligations and objects of its clauses, leading to inconsistencies. Such inconsistencies may create, in the long run, unforeseen legal problems for one or more of the involved parties. To identify and solve such conflicts and inconsistencies, the contract maker needs to read the entire contract and identify each conflicting pair of norms. As conflicts tend to have a large number of norms, the task of identifying norm conflicts is quite difficult for human beings, which makes it error-prone and takes substantial human effort.

Our main contributions in this work are two: first, an approach to address the problem of identifying and quantifying potential normative conflicts between natural language contract clauses; and second, a corpus containing normative conflicts [1]. We process raw text from contracts and identify their norms. Then, we train a convolutional neural network to classify norm pairs as conflict or non-conflict. We evaluate our approach using a dataset of contracts in which conflicts have been deliberately but randomly introduced between the norms, obtaining an accuracy around 85% in conflict identification for a 10-fold cross validation.

2 Norms and Contracts

Norms ensure that individuals act according to a defined set of behaviors and are punished when they are perceived not to be complying with them given a social setting [1]. Norms provide a powerful mechanism for regulating conflict in groups, governing much of our political and social lives. They are often represented using deontic logic, which has its origins in philosophical logic, applied modal logic, and ethical and legal theory. The aim of deontic logic is to describe ideal worlds, allowing the representation of deviations from the ideal (i.e. violations) [27]. Thus, deontic logic and the theory of normative positions are very relevant to legal knowledge representation, and consequently they are applied to the analysis and representation of normative systems [16]. Norms often use deontic concepts to describe permissions, obligations, and prohibitions. A prohibition indicates an action that must not be performed, and, if such action is carried out, a violation occurs. Conversely, a permission indicates an action that can either be performed or not, and no violation occurs in either case. In most deontic systems, a prohibition is considered to be equivalent to the negation of a permission, thus, an action that is not permitted comprises a prohibition. Although these two modalities are sufficient to represent most norms, obligations are also commonly employed in norm representation. An obligation represents an action that must be performed, and it is equivalent either to the negation of a permission not to act or a prohibition not to act.

In contracts, norms are defined within clauses and are often directed to one or more parties of the contract. A contract is an agreement that two or more parties enter voluntarily when it is useful to formalize that a certain duty comes into existence by a promise made by at least one of the parties. The creation of a contract formalizes what each party expects from the other, creating a warranty that each party will fulfill their duties [22] and legally enforceable obligations between these parties. These enforceable obligations are defined by a set of norms, which are responsible for describing any expected behavior from the parties.

[1] https://goo.gl/3Hbl1r

With the use of the Internet, electronic contracts arise as a new way to represent formal agreements and are increasingly explored for commercial services. An electronic contract is very similar to a traditional paper-based commercial contract, following the same rules and structure [20]. Almost all types of contract can be represented electronically, leading to the need of managing such contracts, dealing with the representation and evaluation of agreements. In this work, we deal with contracts written in natural language, thus, the task of analyzing and evaluating norms is traditionally done by human readers. As more contracts are required to codify an increasing number of online services which span over multiple countries and different legal systems, the tasks of writing and verifying contracts by humans become more laborious, taking substantial time [10].

2.1 Norm Conflicts

Sadat-Akhavi [23] describes four causes for a norm conflict to arise. The first cause is when the same act is subject to different types of norms. Thus, two norms are in conflict "if two different types of norms regulate the same act, i.e., if the same act is both obligatory and prohibited, permitted and prohibited, or permitted and obligatory". For example, consider a norm n1 that states that company X must pay product Z taxes, and a norm n2 that states that company X may pay product Z taxes. The second cause is when one norm requires an act, while another norm requires or permits a 'contrary' act. In this case, there is a normative conflict if "two contrary acts, or if one norm permits an act while the other norm requires a contrary act" [23]. For example, consider a norm n1 that states that Company X shall deliver product Z on location W, whereas norm n2 states that company X must deliver product Z on location Q. The conflict arises in the moment that one tries to comply with one norm and, at the same time, is non-complying with the other. The third case defines a cause of conflict when a norm prohibits a precondition of another norm. For example, norm 1 obliges company X to perform α in location θ, whereas norm 2 prohibits company X to be in location θ. In this case, company X cannot comply with norm 1 since been in location θ implies in a violation of norm 2. Finally, Sadat-Akhavi defines a cause of conflict when one norm prohibits a necessary consequence from another norm. For example, norm 1 states that company X shall/may replace its material supplier each year and the process shall not last more than two weeks, whereas norm 2 states that company X cannot be without a material supplier. In this case, the process of replacing the material supplier (norm 1) implies to company X an amount of time without a material supplier, complying with such norm makes company X violate norm 2.

3 Deep Learning

Deep learning is a branch of machine learning that tries to solve problems by automatically finding an internal representation based on hierarchical layers [12]. Such layers can extract complex features from data as they get deeper, which makes feature design from human engineers unnecessary [3]. There are multiple architectures of deep neural networks that achieve this type of learning, such as, convolutional neural networks (CNN) [4], recurrent neural networks (RNN) [15], and autoencoders [26].

3.1 Convolutional Neural Networks

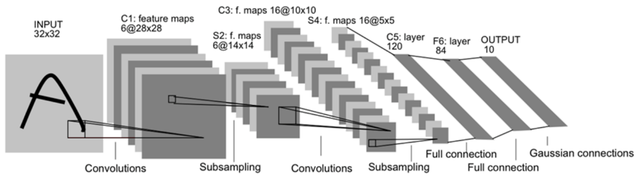

Figure 1: Abstract representation of a CNN (extracted from LeCun *et al.* [4])

Convolutional neural networks were first introduced by LeCun *et al.* [4]. They modify the usual neural network by adding successive convolutional layers before the fully connected neural network output layer, as illustrated in Figure 1. A convolutional layer uses the convolution mathematical operator to modify specific regions of input data using a set of kernels, substantially diminishing the number of neural connection weighs a learning algorithm must adjust close to the input features. A convolution can be viewed as an operation between two functions that produces a third one. Each kernel of a convolutional layer has a defined size and contains a value for each cell; these values, called weights, multiply the values from the input features resulting in a new feature map. The kernel goes through the input multiplying every matrix cell, as illustrated in Figure 2. The result of applying multiple convolutions to an input is a set of feature maps with specific information from the input.

In order to reduce the dimensionality of features resulting from convolutions, convolutional networks often contain pooling layers between successive convolutional layers. These layers have a single kernel without weights that goes through the input aiming to down-sample the size of the image, much in the same way resizing an image reduces its dimensions, as illustrated in Figure 3. They can be either a max pooling or a mean pooling, the former outputs the highest value among the ones in the kernel size and the later outputs the mean value among the ones in the kernel. LeCun *et al.* use this type of neural network to identify handwritten numbers from zip codes in real U.S. mail. From then on, convolutional neural networks have been used extensively to solve image processing problems. More recently, researchers have used CNNs to solve classical natural language processing problems ([28], [11]), such as part-of-speech tagging, named entity recognition, and sentiment analysis. In most cases, approaches using CNNs have matched and surpassed previous approaches using rule-based and probabilistic approaches. The key challenge in applying CNNs to text processing is finding a suitable matrix representation for the input text.

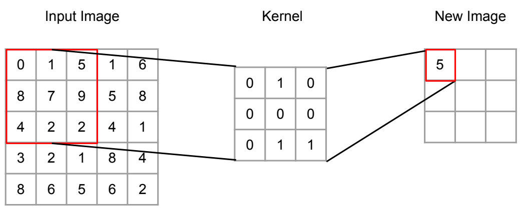

Figure 2: Convolution example

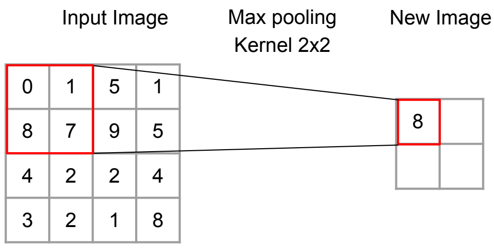

Figure 3: Pooling example

4 Conflict Detection Approach

Our approach to identifying potential conflicts between norms in contracts is divided into two phases. In the first one, we identify norms within contractual sentences by training a Support Vector classifier using a manually annotated dataset. In the second part, we classify norm pairs as conflicting or non-conflicting using a CNN. Figure 4 illustrates the architecture of our approach.

4.1 Norm Identification

The first step towards norm conflict identification is to identify which sentences in a contract contain deontic statements (norms). For this task we consider contract sentences to be of two exclusive types: norm sentences and non-norm sentences.

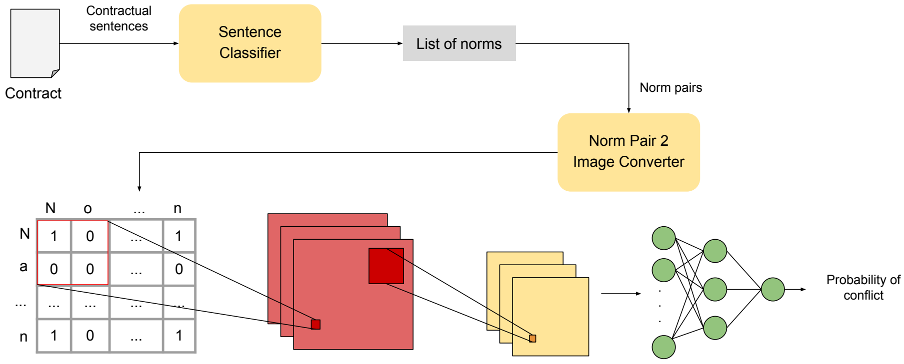

Figure 4: Architecture of the norm conflict identifier

In order to separate norm sentences from the rest of the contract text, we train a classifier based on Support Vector Machines (SVM) using a manually annotated dataset. We created the dataset using real contracts extracted from the *onecle* website[2], specifically contracts of the manufacturing domain[3]. We manually annotated the sentences in each contract as being either norm or non-norm, resulting in a set of 699 norm sentences and 494 non-norm sentences from a total of 22 contracts, which we use as both train and test sets.

4.2 Norm Conflict Identification

In order to identify norm conflicts, we use the concepts introduced by Sadat-Akhavi [23]. Unlike the four causes for conflicts, Sadat-Akhavi identifies three main types of conflicts, which are:

- Permission x Obligation;
- Permission x Prohibition; and
- Obligation x Prohibition.

We base our conflict identification on these three conflict types in addition to the first and second causes of norm conflict defined by Sadat-Akhavi. Thus, in this work, we consider norm conflicts to be:

- Pairs of norms with different deontic concepts applied to the same actions and the same parties; and
- Pairs of norms where the obliged action of one clause is either prohibited or permitted in another clause.

[2] http://contracts.onecle.com/
[3] http://contracts.onecle.com/type/47.shtml

The key challenge in processing text using CNNs is to generate a representation suitable for the matrix-format input required for the convolutional layers. Here, we take inspiration from recent work that deals with natural language. The first sentence representation, created by Zhang and LeCun [28], in which they use a CNN to deal with natural language processing problems. Their approach aims to, among other tasks, classify the sentiment (positive, negative, and neutral) of product reviews from Amazon. Since CNNs are designed to process images, the solution they propose to translate a sentence into an image is to create a matrix representation with the review characters as lines and the alphabet as columns. Thus, given a cell $\{i, j\}$, they assign 1 when the *ith* character is equal to the *jth*, otherwise, they assign 0. Figure 5 illustrates their sentence representation using as example a sentence that begins with 'above'. The resulting matrix has 1 where letters are equal (such as cell $1, 1$ and $2, 2$) and 0 otherwise.

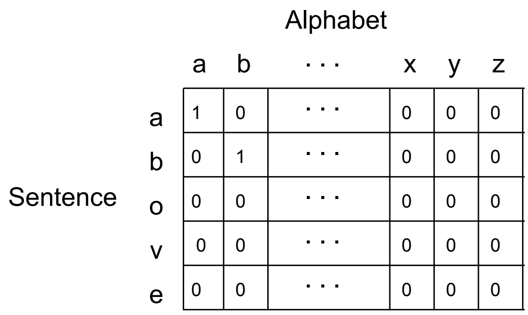

Figure 5: Sentence representation by Zhang and LeCun

The second work is from Kim [17], which uses a sentence representation to classify sentences in different natural language processing problems. Here, the representation is a matrix in which the lines are the words of a sentence and columns are the word embedding of each word. An embedding is a representation that turns words into vectors of floating point numbers. Such representation may have a variable size and carries semantic information from each word. In Kim's approach, the resulting matrix is a group of word embedding lines. Figure 6 illustrates this sentence representation.

One of the key aspects in norm conflicts is that both norms tend to be very similar in that usually both norms refer to the same party/parties with similar actions, and only the modality of the sentence differs. Thus, the similarity distance between two sentences often indicates how norm pairs are likely to conflict. Consequently, we rely on training examples that consist of binary images created from each pair of norms denoting the distance between these norms. Thus, we created a pair-of-norms representation using a matrix to denote similar characters in each norm. Given two norms α and β, our matrix consists of the

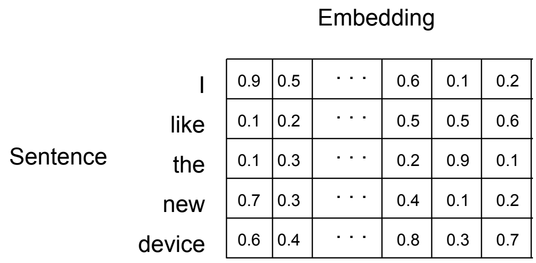

Figure 6: Sentence representation by Kim

characters from α in its lines and the characters from β in its columns, as Figure 7 illustrates. Given a cell {i, j}, we assign 1 to it when the i^{th} character of α is equal to the j^{th} character of β and 0 otherwise. For this work, we limit the lengths of both norms to 200 characters, which is the mean length of norms from our dataset and truncate overlong sentences (which, as we see in the experiments, seems to have no effect in accuracy). Using this representation we train a CNN to generate a model to classify norm pairs as conflicting and non-conflicting.

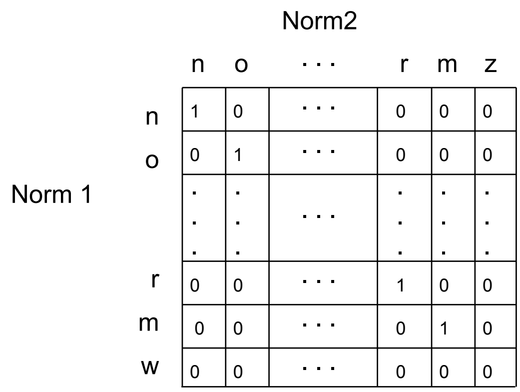

Figure 7: Norm pair representation in our approach

5 Experiments

5.1 Norm Conflict Dataset Annotation

To evaluate our approach to detect potential conflicts between norms, we required a corpus with contracts containing real conflicts. However, since we found no such corpus available, we created a dataset with semi-automatically generated norm conflicts using a set of real non-conflicting norms as a basis. To assist in the creation of conflicts, we developed a system to assist human users to insert conflicts randomly in a contract, while still maintaining language syntactic correctness. In order to create such conflicts, we relied on the assistance of two volunteers each of which was responsible for inserting two different types of conflict. Each volunteer was asked to create one of the two causes of conflict. We asked the first volunteer to insert conflicts that have only differences in the modal verb, e.g. changing an obligation modal verb ('must') for a permission one ('may'). This volunteer created 94 conflicts in 10 different contracts, totaling 13 conflicts between Permission x Prohibition, 36 conflicts between Permission x Obligation, and 46 conflicts between Obligation x Prohibition. We asked the second volunteer to insert conflicts that contain deontic conflicts and modifications in the norm actions. This volunteer created 17 conflicts in 6 different contracts, totaling 2 conflicts between Permission x Prohibition, 6 conflicts between Permission x Obligation, and 4 conflicts between Obligation x Prohibition.

We developed a semi-automatic process conflict creation within a system that, when prompted, selects a random norm from a random contract, makes a copy of it, and asks the user to modify it. After user modification, the system creates a new contract containing both the original norm and the modified copy, ensuring that a semantically similar, but conflicting, clause is present in the resulting contract. Thus, we use these new contracts to identify the inserted conflicts.

From the contracts we used to create conflicts, we selected all sentences not used in the conflict creation to produce a set for the non-conflicting norm class. This set has a total of 204,443 norm pairs.

5.2 SVM

To create the sentence classifier, we trained a support vector machine (SVM) classifier using the dataset described in the Norm Identification section. SVM is often used to classify datasets with few training examples with multiple features and a binary classification task since it creates a hyperplane that tries to find the best division between two classes [14]. In order to train the SVM, we turn each sentence into a bag-of-words representation [13], which represents the frequency of words from a fixed dictionary in sentence. Using this representation, the SVM learns from the frequency each word appears in a class.

5.3 CNN

To create the norm conflict identifier, we train a CNN using norm pairs from the dataset described in Norm Conflict Dataset Annotation section. In this work, we use the classical *LeNet* CNN, developed by LeCun *et al.* [4]. The network architecture consists of two convolutional layers followed by a max pooling layer and two fully-connected neural networks. Each convolutional layer has 32 kernels that are responsible for extracting features from the input image. The network receives as input an image representation of each norm pair.

6 Results

6.1 Sentence Classifier

To evaluate our sentence classifier, we divided our manually annotated dataset into train and test set. We use a 80/20 division, which results in 954 sentences in the train set and 238 sentences in the test set. Both sets are balanced according to the number of elements in each class, i.e., 559 norm sentences and 395 non-norm sentences in the train set, and 139 norm sentences and 98 non-norm sentences in the test set. To compare the SVM with other linear models, we test the same dataset with two other classifiers: Perceptron and Passive Aggressive. Perceptron is a well-known linear model, which can be better explained as a neuron in a neural network [19]. It processes the input by multiplying it using a set of weights. The result goes to an activation function, which defines the input class. Passive Aggressive [2] is a linear model that has its name based on its weight update rule that, in each round, can be passive, when the hinge-loss result of its update is zero, and aggressive, when it is a positive number. Table 1 shows the results for each classifier.

Classifier	**Prec.**	**Rec.**	**F-Score**	**Acc.**
Perceptron	0.89	0.88	0.88	0.87
Pass. Agr.	**0.92**	0.88	0.90	0.89
SVM	0.88	**0.94**	**0.91**	**0.90**

Table 1: Results for sentence classifier

As we can see, SVM obtains the best result for the task with an accuracy of 90%. The passive aggressive algorithm has a similarly good accuracy and has the best precision in comparison to the others. However, since SVM obtains a better overall result, we use it as our sentence classifier.

6.2 Norm Conflict Identifier

To evaluate the norm conflict identifier, we used a 10-fold cross-validation step dividing our dataset into train, validation, and test. Since we have a total of 104 norm pairs with conflicting norms and 204,443 conflict-free norm pairs, the first step is to create a balanced dataset. Thus, we reduced the number of non-conflicting norm pairs to 104, which gives us a total of 208 samples. Each fold has 10% of the data, which is around twenty samples, ten of each class. In each round, we use eight folds to train, one to validate, and one to test. To prevent overfitting, we use the early stopping technique that monitors the accuracy in the train and validation set. When the accuracy in the validation set starts to decrease and the train accuracy keeps increasing, an overfitting is detected, resulting in the termination of the training phase. We show the accuracy results for each fold and the mean accuracy overall in Table 2.

Fold	0	1	2	3	4	5	6	7	8	9	**Mean**
Accuracy	0.85	0.85	0.76	0.95	0.85	0.76	0.71	0.95	0.95	0.80	0.84

Table 2: Results for the norm conflict identifier

7 Related Work

Since our approach merges information retrieval, which is the extraction of information from unstructured data, and contract reasoning, which is manipulation and reasoning over contract elements, in this section we compare our approach to recent work that deals with similar concepts applied to contracts.

Rosso *et al.* [21] propose an approach to retrieve information from legal texts. Their approach uses JIRS[4] (Java Information Retrieval System), a system that measures distances between sentences using n-grams, to develop a solution for three problems: passage retrieval in treaties, patents, and contracts. In the first problem, they want to answer questions from treaty documents. Given a question about the content of the treaty, they use JIRS to measure the distance between the question and the text in the treaty, thus, they can rank the best answers to each question by their similarity. To the second problem, they develop an approach to help patent creators identify similar patents. As in the first problem, given a set of patents and a new one, they use JIRS to measure how similar the new patent is to existing ones. To the third problem, they develop an approach to identify conflicts between norms in contracts. To do so, they create a contract example between an airline and a ground operations company with a defined set of norms applied to both parties. They divide the process of conflict identification into three steps, first, they translate every norm in contract to a formal contract language ($\mathcal{CL}$ [7]), which they call Contract Language clauses. Second, they analyse the clauses using a model checker performed by the contract analysis tool *CLAN* [8]. From the identified conflicts, they use JIRS to translate the sentences from $\mathcal{CL}$ to natural language. Although this work also tries to identify norm conflicts, it differs from ours in two points. First, our work tries to identify normative conflicts dealing directly with natural language, whereas in their work they use the approach proposed by Fenech *et al.* [8], which uses a single contract that has its norms *manually translated* into the controlled language $\mathcal{CL}$. Second, to identify norm conflicts, *CLAN* uses a series of predefined rules, whereas in our approach we rely on a convolution neural network that processes matrix of distances between pairs of norms automatically extracting the information needed to classify them.

Curtotti and McCreath [5] propose an approach to annotate contracts using machine learning and rule-based techniques. They aim to classify each component of contractual sentences based on their structure. To extract data for machine learning, they create a hand-coded tagger and manually correct its outcome. As data, they use the Australian Contract Corpus [6] with 256 contracts, containing

[4] http://sourceforge.net/projects/jirs/

42910 sentences and a vocabulary of 14217 words. In their experiments, they randomly select 30 contracts and divide them into three sets, one for train and two for test. Using different classifiers to compare the results, they obtain 0.86 of F-score. Instead of classifying each clause structure with a different class, in this work we want to identify norm clauses. However, we can use Curtotti and McCreath annotation for a further work with a deeper contract analysis.

Gao and Singh propose two different solutions for problems concerning information extraction from contracts. In the first one, they propose an approach to extract exceptions within norms in contracts [10]. They use a corpus with 2,647 contracts from the Onecle repository [5] as data for processing. As result, Enlil obtains an F-score of 0.9 in classifying contracts using a manually annotated corpus. Although Gao and Singh work is similar to ours by dealing with contractual norms, we have different ends. In our work, we use norms to find potential conflicts, whereas they use them to identify exceptions within a contract. However, we can use their concept of exception in a new approach to identify conflicts with a high-level of detail, since exceptions in norms may induce to new types of conflicts.

In their second work, Gao and Singh [9] develop a hybrid approach for extracting business events and their temporal constraints from contracts. Using different machine learning algorithms they obtain an F-score of 0.89 for event extraction and 0.9 for temporal constraints. This, similar to the first work, is an approach to extract information from contracts. The main difference between their work and ours is that they try to identify temporal elements from norms. This is also an improvement we can apply to the norm conflict identification process.

Vasconcelos *et al.* [25] propose an approach to deal with normative conflicts in multi-agent systems. They develop mechanisms for detection and resolution of normative conflicts. To resolve conflicts they manipulate the constraints associated to the norms' variables, removing any overlap in their values. In norm adoption, they use a set of auxiliary norms to exchange by the ones applied to the agent. In norm removal, they remove a certain norm and all curtailments it caused, bringing back a previous form of the normative state. Figueiredo and Silva's work [24] consist of an algorithm for normative conflict detection using first-order logic. They use the Z language to formalize the conflict types and then identify them between norms. Both approaches from Vasconcelos *et al.* and Figueiredo and Silva propose a solution for norm conflicts applied to normative multi-agent systems. The main difference between their work and ours is that we make the identification of potential conflicts between norms from contracts written in natural language. It creates the need for a different approach since natural language is not structured. However, an alternative approach would be the translation of natural language to first-order-logic and use one of these approaches to identify conflicts.

[5] http://contracts.onecle.com

8 Conclusion and Future Work

In this work, we developed a two-phase approach to identify potential conflicts between norms in contracts. Our main contributions are: (1)a dataset with manually annotated normative and non-normative sentences from real contracts; (2) a machine learning model to classify contractual sentences as normative and non-normative; (3) a manually annotated dataset with contracts containing conflicts between norms; (4) and a deep learning model to classify norm pairs as conflicting and non-conflicting. We evaluate both models and we obtain an accuracy of 90% for the sentence classifier and around 85% for the norm conflict identifier.

As future work, we aim to develop two different approaches. First, we aim to develop a pre-processing step in the norm conflict identification to identify elements that may improve the detection of conflicts, such as temporal information. Second, to fairly compare our results with the work proposed by Fenech *et al.* [8], we aim to create an approach to translate natural language to $\mathcal{CL}$ (contract language) and use *CLAN* to discover conflicts.

Acknowledgements: We gratefully thank Google Research Awards for Latin America for funding our project.

References

1. Axelrod, R.: An evolutionary approach to norms. The American Political Science Review 80(4), pp. 1095–1111 (1986)
2. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research 7(Mar), 551–585 (2006)
3. Cun, L., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
4. Cun, L., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Handwritten digit recognition with a back-propagation network. In: Advances in Neural Information Processing Systems. pp. 396–404. Morgan Kaufmann (1990)
5. Curtotti, M., Mccreath, E.: Corpus based classification of text in australian contracts. In: Proceedings of the Australasian Language Technology Association Workshop, Melbourne, Australia. pp. 18–26 (2010)
6. Curtotti, M., McCreath, E.C.: A corpus of australian contract language: Description, profiling and analysis. In: Proceedings of the 13th International Conference on Artificial Intelligence and Law. pp. 199–208. ICAIL '11, ACM, New York, NY, USA (2011)
7. Fenech, S., Pace, G.J., Schneider, G.: Automatic conflict detection on contracts. In: International Colloquium on Theoretical Aspects of Computing. pp. 200–214. Springer (2009)
8. Fenech, S., Pace, G.J., Schneider, G.: CLAN: A tool for contract analysis and conflict discovery. In: Automated Technology for Verification and Analysis, 7th International Symposium, ATVA 2009, Macao, China, October 14-16, 2009. Proceedings. pp. 90–96 (2009)
9. Gao, X., Singh, M.P.: Mining contracts for business events and temporal constraints in service engagements. Services Computing, IEEE Transactions on PP(99), 1–1 (2013)

10. Gao, X., Singh, M.P., Mehra, P.: Mining business contracts for service exceptions. IEEE Transactions on Services Computing 5(3), 333–344 (2012)
11. Gillick, D., Brunk, C., Vinyals, O., Subramanya, A.: Multilingual language processing from bytes. arXiv preprint arXiv:1512.00103 (2015)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning (2016), book in preparation for MIT Press
13. Harris, Z.S.: Distributional Structure, pp. 775–794. Springer Netherlands, Dordrecht (1970)
14. Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their Applications 13(4), 18–28 (1998)
15. Jain, L.C., Medsker, L.R.: Recurrent Neural Networks: Design and Applications. CRC Press, Inc., Boca Raton, FL, USA, 1st edn. (1999)
16. Jones, A.J.I., Sergot, M.J.: Deontic logic in the representation of law: Towards a methodology. Artificial Intelligence and Law 1(1), 45–64 (1992)
17. Kim, Y.: Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 1746–1751. ACL (2014)
18. Meneguzzi, F., Rodrigues, O., Oren, N., Vasconcelos, W.W., Luck, M.: {BDI} reasoning with normative considerations. Engineering Applications of Artificial Intelligence 43(0), 127 – 146 (2015)
19. Minsky, M., Papert, S.: Neurocomputing: Foundations of research. In: Anderson, J.A., Rosenfeld, E. (eds.) Neurocomputing: foundations of research, chap. Perceptrons, pp. 157–169. MIT Press, Cambridge, MA, USA (1988)
20. Prisacariu, C., Schneider, G.: A formal language for electronic contracts. In: International Conference on Formal Methods for Open Object-Based Distributed Systems. pp. 174–189. Springer (2007)
21. Rosso, P., Correa, S., Buscaldi, D.: Passage retrieval in legal texts. The Journal of Logic and Algebraic Programming 80(3-5), 139–153 (2011)
22. Rousseau, D.M., McLean Parks, J.: The contracts of individuals and organizations, vol. 15. JAI Press Ltd. (1993)
23. Sadat-Akhavi, A.: Methods of Resolving Conflicts Between Treaties. Graduate Institute of International Studies (Series), V. 3, M. Nijhoff (2003)
24. da Silva Figueiredo, K., da Silva, V.T.: An algorithm to identify conflicts between norms and values. In: Coordination, Organisations,Institutions and Norms in Multi-Agent Systems. pp. 259–274 (2013)
25. Vasconcelos, W.W., Kollingbaum, M.J., Norman, T.J.: Normative conflict resolution in multi-agent systems. Autonomous Agents and Multi-Agent Systems 19(2), 124–152 (2009)
26. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. pp. 1096–1103. ICML '08, ACM, New York, NY, USA (2008)
27. von Wright, G.H.: Deontic Logic, New Series, vol. 60. Oxford University Press on behalf of the Mind Association (1951)
28. Zhang, X., Cun, L.: Text understanding from scratch. CoRR abs/1502.01710 (2015)