A Deep Learning Approach for Norm Conflict Identification (Extended Abstract)

João Paulo Aires
Potifical Catholic University of Rio Grande do Sul
Av. Ipiranga, 6681
Porto Alegre, Brazil
joao.aires.001@acad.pucrs.br

Felipe Meneguzzi
Potifical Catholic University of Rio Grande do Sul
Av. Ipiranga, 6681
Porto Alegre, Brazil
felipe.meneguzzi@pucrs.br

CCS Concepts

•**Computing methodologies** → *Natural language processing; Information extraction;*

Keywords

norm conflict, deep learning, contract reasoning

1. INTRODUCTION

Regulations are often applied to social members in a society in order to minimize conflicting behaviors [9]. Such regulations, also known as social norms, define expected behaviors for society members [7] and help ensure that individuals act in 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 sets of norms applied to a certain agreement is through contracts [8]. 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 contracts may have a large number norms, the identification of norm conflicts by human beings takes substantial effort and tends to be error-prone.

We address the problem of identifying and quantifying potential normative conflicts between natural language contract clauses [1]. Our main contributions consist of an approach based on deep learning to address the problem of identifying potential normative conflicts between natural language contract clauses, as well as the corpus containing normative conflicts [2] using to train the classifiers involved. 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 of 95% in conflict identification.

2. CONFLICT DETECTION APPROACH

Our approach to identify 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 machines using a manually annotated dataset. In the second part, we classify norm pairs as conflict or non-conflict using a convolutional neural networks (CNN). Figure 1 illustrates the architecture of our approach.

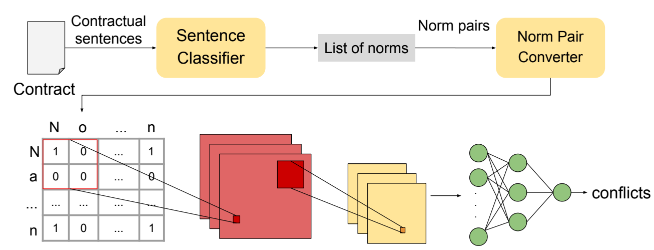

Figure 1: Architecture of the norm conflict identifier

2.1 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. 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[1], specifically contracts of the manufacturing domain[2]. 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, described in Section 3.1.

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

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

2.2 Norm Conflict Identification

In order to identify norm conflicts, we use the concepts introduced by Sadat-Akhavi [11]. Sadat-Akhavi identifies three main types of conflicts, they are: Permission $\times$ Obligation; Permission $\times$ Prohibition; and Obligation $\times$ Prohibition. We base our conflict identification following these three conflict types in addition to the first and second causes of norm conflict defined by Sadat-Akhavi. Thus, we consider norm conflicts to be: 1) a pair of norms with different deontic concepts applied to the same actions and same parties; and 2) a Pair of norms where the obliged action of one is either prohibited or permitted in another.

The key challenge in processing text using CNNs is to generate a representation suitable for the matrix-format input required for the convolutional layers. We propose a binary matrix to represent pair of norms based on recent work from Zhang and LeCun [12] and Kim [6]. Conflicting norms tend to be very similar in that usually both norms refer to the same party with similar actions, and only the modal tone of the sentence differs. 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 create 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 characters from α in its lines and the characters from β in its columns. 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. We limit the lengths of both norms to 200 characters, which is the mean length of norms from our dataset and truncate overlong sentences. Truncation has no noticeable effect in accuracy in our experiments. Using this representation, we train a CNN to generate a model to classify norm pairs as norm conflicts and non-norm conflicts.

Given a lack of corpora with real contract conflicts, we created a dataset with semi-automatically generated norm conflicts from a set of real norms as a basis. We developed a system for human users to insert conflicts randomly in a contract, while still maintaining language syntactic correctness. To create conflicts, we relied on the assistance of two volunteers each of which was responsible for inserting two different types 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. 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.

3. RESULTS

3.1 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 [10]. 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 [3] is liner 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 has the best result for the task with an accuracy of 90%. The passive aggressive algorithm has similar accuracy and the best precision in comparison to the others. However, since SVM obtains a better F-Score result, we use it as our sentence classifier.

3.2 Norm Conflict Identifier

To evaluate the norm conflict identifier, we divided our dataset into train, validation, and test set. Since we have a total of 104 norm pairs with conflicting norms and 204,443 norm pairs conflict-free, the first step is to create a balanced dataset. Thus, we reduce the number of non-conflict norm pairs to 104, which gives us a total of 208 samples. We divide this dataset by separating 80% to train, 10% to validation, and 10% to test. Each set has a balanced number of elements from each class.

We perform a 6-epoch training, testing each epoch using the validation set. After training, we use the model with the best result during training to classify the test set. As result, we got 95% of accuracy in the test set, which, in absolute terms, means we correctly classified 19 out of 20 norm pairs.

4. CONCLUSION AND FUTURE WORK

In this work, we propose a two-phase approach to identify potential conflicts between norms in contracts. Our main contributions are: a dataset with manually annotated norms and non-norm sentences from real contracts [2], a machine learning model to classify contractual sentences as norm and non-norm; a manually annotated dataset with contracts containing conflicts between norms; and a deep learning model to classify norm pairs as conflict and non-conflict. We evaluate both models and we get an accuracy of 90% to the sentence classifier and 95% to the norm conflict identifier.

As future work, we aim to implement two approaches. First, we aim to make 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.* [5], we aim to create an approach to translate natural language to $\mathcal{CL}$ (contract language) [4] and use a logical verification approach such as $CLAN$ to find conflicts.

5. ACKNOWLEDGEMENTS

We would like to thank Google Research Awards Latin America for funding our research.

REFERENCES

[1] J. P. Aires, V. L. S. d. Lima, and F. Meneguzzi. Identifying potential conflicts between norms in contracts. In *International Workshop on Coordination, Organisation, Institutions and Norms in Multi-Agent Systens*, 2015.

[2] J. P. Aires, D. Pinheiro, and F. Meneguzzi. JoaoPauloAires/norm-dataset: Dataset with Norms and Norm Conflicts, Mar. 2017. https://doi.org/10.5281/zenodo.345411.

[3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

[4] S. Fenech, G. J. Pace, and G. Schneider. Automatic conflict detection on contracts. In *International Colloquium on Theoretical Aspects of Computing*, pages 200–214. Springer, 2009.

[5] S. Fenech, G. J. Pace, and G. Schneider. 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*, pages 90–96, 2009.

[6] Y. Kim. Convolutional neural networks for sentence classification. In A. Moschitti, B. Pang, and W. Daelemans, editors, *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*, pages 1746–1751. ACL, 2014.

[7] M. Luck, S. Mahmoud, F. Meneguzzi, M. Kollingbaum, T. J. Norman, N. Criado, and M. S. Fagundes. Normative agents. In S. Ossowski, editor, *Agreement Technologies*, volume 8 of *Law, Governance and Technology Series*, pages 209–220. Springer Netherlands, 2013.

[8] F. Meneguzzi, S. Modgil, N. Oren, S. Miles, M. Luck, and N. Faci. Applying electronic contracting to the aerospace aftercare domain. *Engineering Applications of Artificial Intelligence*, 25(7):1471–1487, 2012.

[9] F. Meneguzzi, O. Rodrigues, N. Oren, W. W. Vasconcelos, and M. Luck. {BDI} reasoning with normative considerations. *Engineering Applications of Artificial Intelligence*, 43(0):127 – 146, 2015.

[10] M. Minsky and S. Papert. Neurocomputing: Foundations of research. In J. A. Anderson and E. Rosenfeld, editors, *Neurocomputing: foundations of research*, chapter Perceptrons, pages 157–169. MIT Press, Cambridge, MA, USA, 1988.

[11] A. Sadat-Akhavi. *Methods of Resolving Conflicts Between Treaties*. Graduate Institute of International Studies (Series), V. 3. M. Nijhoff, 2003.

[12] X. Zhang and L. Cun. Text understanding from scratch. *CoRR*, abs/1502.01710, 2015.