ConCon: A Contract Conflict Identifier

Demonstration

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

Roger Granada  
Pontifical Catholic University of Rio Grande do Sul  
Porto Alegre, Brazil  
roger.granada@acad.pucrs.br

Felipe Meneguzzi  
Pontifical Catholic University of Rio Grande do Sul  
Porto Alegre, Brazil  
felipe.meneguzzi@pucrs.br

ABSTRACT

Contracts are the main medium through which people and legal entities formalise their trade relations, be they the exchange of goods or the specification of mutual obligations. While electronic contracts allow automated processes to verify their correctness, most agreements in the real world are still encoded in contracts written in natural language, necessitating substantial human revision effort to eliminate possible conflicting statements, especially for long and complex contracts. We demonstrate the ConCon (Contract Conflicts) tool, to automatically read natural language contracts and indicate potential conflicts among their clauses. Using our tool, legal professionals and the general public can benefit from a ranking of potential conflicts between the clauses in a contract, saving time and effort from legal experts in contract proof-reading.

KEYWORDS

Natural Language Processing; Norms; Norm Conflicts; Semantic Representation

ACM Reference Format:

1 INTRODUCTION

Most societies use contracts as a central tool to formalise agreements [7]. Contracts are semi-structured documents that describe the agreement subject, its parties, and a series of norm clauses of what is expected from each party during the agreement validation. Using deontic logic concepts, norms describe prohibitions, permissions, and obligations. During contract creation, one needs to be aware of how norms relate to each other. When the definition of one norm interferes with the definition of another one, we have a norm conflict. Norm conflicts occur for many reasons [8], such as different deontic meanings (e.g., prohibition × obligation) over the same action, conditional effects cancelling effects from other norms, among others. To avoid such conflicts, contract writers need to read all of them and compare all clauses to each other, demanding significant time and effort. We introduce ConCon1, a web-tool that allows users to upload their contracts and verify the existence of potential conflicts between norms. ConCon implements three conflict identification models, which the public can use to check their own contracts. Each conflict identifier uses a different approach, which ranges from rule-based mechanisms [3] to deep learning [2] and embedding approaches [1]. The ConCon web-tool facilitates proof reading for conflicts as people can prioritise potential conflicts indicated by the tool, and that could create contract inconsistencies. The video accompanying our demonstration is available at https://goo.gl/1JPHXL.

2 MOTIVATION AND BACKGROUND

The identification of norm conflicts is a challenging task that, when performed by humans, involves reading norms multiple times to ensure none is conflicting. As contracts are usually long and complex, i.e., consist of hundreds of norms with conditions and exceptions, defining conflicting cases becomes time consuming and requires multiple verification.

Norms regulate autonomous agents by specifying expected behaviours from members of a specific group [4] using deontic logic to describe prohibitions, obligations, and permissions. In contracts, norms govern how the agreement must occur between the parties. A conflict between norms arises when the definition of one norm cancels the effects of another one. Conflict cases can be divided in levels of complexity [8]. The most common and yet simple case is when we have only differences between the deontic meanings of two norms [6], e.g., prohibition and obligation of the same action. For example, consider the following norm pair: (1) Creditor must print receipts for all purchasers.; (2) Creditor shall not print receipts for purchasers. Here, both norms refer to the same action to the same party with opposite deontic verbs. The existence of both norms in the same contract may invalidate them as one cannot comply with one without defying the other.

More complex conflict cases may involve the existence of a condition that contradicts the definition of an existing norm. For example, consider the following norm pair: (1) In case of payment delay, Debtor must pay double the price.; (2) Debtor shall not pay more than the agreed price for the product. In this case, the condition in (1) allows the party to perform a prohibited action by (2).

2.1 Approach

ConCon consists of a set of contract conflict detection approaches with a Django2 graphical web interface. We host it using our own server and make our web-tool publicly available online3. We use three different approaches for conflict identification. The first one [3]
is a rule-based approach that selects norms from contracts and breaks them into: party, modal verb, and action. Using such division, it identifies conflicts by comparing norm structures and selecting pairs where both norms have the same party, different deontic meanings, and semantic similar actions. The key element in this approach is the semantic similarity algorithm we proposed, which measures the similarity between norm actions. This algorithm allows us to select conflicts that have similar meanings but different words. The second approach [2] uses a convolution neural network to classify norm pairs as conflicts or non-conflicts. To classify norm pairs, we convert them into binary matrices by setting 1 when characters from both norms are the same, and 0 otherwise. The convolution neural network receives the binary matrix and process it as an image classifying it as conflict or not. Finally, the third approach [1] compares embedding representations of norms to identify conflicts. In this approach, we convert all norms into embedding representations using Sent2Vec [5]. Then, we create a conflict offset using a set of conflicting norm pairs. By subtracting conflicting norm embeddings from the set and calculating an average of these subtractions, we obtain a conflict embedding. Using the conflict offset, we measure the distance between it and each embedding resulting from the subtraction of two norm embeddings and use a threshold to classify the norm pair as conflicting. Using these three approaches, we allow a user to compare the outputs and choose the best one.

In all three approaches, we use the same dataset for both training and testing. Our manually annotated dataset is available online. It consists of 228 conflicting norms and 11,329 non-conflicting norm pairs. The conflicts vary in levels of complexity that includes simple cases with only modal verb modifications (modifying the deontic modality); and complex cases with modal verb and sentence structure modifications, which include the use of synonymous that preserve the meaning but with different words. In order to compare approaches fairly, we performed a 10-fold cross-validation and report the results in Table 1, which shows the mean accuracy for each conflict identifier over the dataset.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-based [3]</td>
<td>0.78</td>
</tr>
<tr>
<td>CNN-based [2]</td>
<td>0.84</td>
</tr>
<tr>
<td>Embedding-based [1]</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Table 1: Accuracy for conflict identifiers.

3 APPLICATION

We created ConCon with two main goals. First, we aim to make it available for everyone as a web tool to preprocess contracts and help on identifying potential conflicts in them. Second, we want to use the tool as data annotator where users can remove misclassified conflicts and annotate unclassified ones. In order to use ConCon, we ask users to register and accept a term of use since we aim to use uploaded contracts as future data for training conflict identifier models. In this term of use, we ask permission to users to use their uploaded contracts and their annotations in our research. We make sure that this information will be used only to train new approaches that aim to enhance our classifiers. Once the user is registered, we provide a profile section where all uploaded contracts can be accessed and processed. In the Conflict section, users can upload a contract in natural language and choose one of our three conflict identifiers to process the contract. As result, we provide a table with two main columns, in each column, we have the norms of the contract as rows. They are the same in both columns, we use this structure to show conflicting norms in different parts of the same contract. Using a dropdown button, we list all potential conflicts found with by our conflict identifier with a degree of confidence. When clicking in one of the potential conflicts listed, rows in both columns focus on the selected norm pair so the user can check the conflict. Figure 1 illustrates an example of highlighted potential conflict between two norms. Below the table we have two interactive buttons. The first one allows the user to indicate if the highlighted conflict is a false positive one. The second asks the user to select a norm pair and indicate it as a conflict.

![Figure 1: Conflict identification example.](image)

4 DISCUSSION AND FUTURE WORK

So far ConCon has 26 registered users, 87 uploaded contracts, and has helped identify 9980 potential conflicts (considering conflicts found by all conflict identifiers). We aim to obtain more users as we publish our web-tool to the target public. Our goal is to make ConCon a daily tool for contract makers and general public. ConCon has a great potential to serve as an assistant to avoid contract conflicts.

As future work, we aim to improve our tool in at least three ways. First, we will implement ways to make annotated contracts available for download so other researchers can develop new models for contract processing. Second, we aim to make a fine-grained classification of conflicts by defining types of conflicts. This classification will allow us to suggest ways for the user to solve conflicts. As part of this enhancement, we aim to allow users to edit their contracts in the web tool. Finally, we aim to incorporate a contract structure identifier, that will help users to automatically discover contractual elements. Thus, users can select norms applied to a certain party, check norm deadlines, and browse norms easily.

Acknowledgement. This study was financed in part by the CAPES - Finance Code 001, and CAPES/FAPERGS agreement (DOCTEX 04/2018). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. We thank Google Research Awards for Latin America for funding part of our project.
REFERENCES


