Norm Conflict Identification using Vector Space Offsets

João Paulo Aires*, Juarez Monteiro*, Roger Granada*, Felipe Meneguzzi†
School of Technology
Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681, 90619-900, Porto Alegre, RS, Brazil
* Email: {joao.aires.001, juarez.santos, roger.granada}@acad.pucrs.br
† Email: {felipe.meneguzzi}@pucrs.br

***Abstract*—Contracts formally represent agreements between parties and often involve the exchange of goods and services. In contracts, norms define the expected behaviors from parties using deontic statements, such as obligations, permissions, and prohibitions. However, norms may conflict when two or more apply to the same context but have different deontic statements, such as a permission to delay payment being present in the same contract as an obligation to pay in a fixed deadline. Contracts with conflicting norms may be invalidated in whole or in part, making conflict identification a major concern in contract writing. Conflict identification in contracts by humans is a time-consuming and error-prone task that would greatly benefit from automated aid. In order to automate such identification, we introduce an approach to identify potential conflicts between norms in contracts written in natural language that compares a latent (vector) representation of norms using an implicit offset that encodes normative conflicts. Experimental evaluation shows that our approach is substantially more accurate than the existing state of the art in an open dataset.**

I. Introduction

Living in society often involves complying with a set of constraints that both limits and structures the individual behaviors. This force that guides the way we interact with each other consists of mostly informal rules commonly known as social norms. Being part of a social group means accepting the specific rules created for it. In a small context, rules are applied to manage agreements between two or more individuals [1]. Agreements are commonly used to define the exchange of goods and services between individuals. Rules in an agreement describe the way each individual is expected to act during the agreement. A more formal way to define agreements is using contracts. Contracts are documents that define the parties, the agreement subject, what each party must comply with, and what is expected in case of violations. Each statement in a contract is defined by a clause and the rules are commonly known as norm clauses [2].

We use contracts in most agreements between people and enterprises [3]. Thus, a large number of contracts need to be created, reviewed, read, and signed. When creating a contract, conflicts between norm clauses may arise without a careful review from contract creators. Norm conflicts often occur when a norm invalidates one or more norms by its definitions [4]. For example, if *norm 1* states that "Company X must pay the taxes for all products." and *norm 2* states that "Company X shall not pay the taxes for product Y", *norm 1* invalidates *norm 2*, since paying all taxes for all products consists of paying the taxes for product Y too. When it happens, complying with one norm means violating the other, it creates a contract inconsistency that can invalidate the whole contract. The key aspect when looking for norm conflicts is the deontic meaning in norms, which states whether a norm refers to a permission, prohibition, or obligation. Most norm conflicts arise when two norms have different deontic meanings and deal with the same context, *e.g.*, norms applied to the same party dealing with the same action, but one being a prohibition and the other a permission.

As contracts tend to be long and complex (*e.g.*, our dataset has an average of 40,000 characters per contract), finding conflicts in such contracts is hard, laborious, and error-prone even for human experts. To avoid it, we need to be able to reason over contracts and detect such norm conflicts. Recent efforts to reason over contracts have generated multiple approaches that deal with parts of the problem of detecting such conflicts [5]–[7], while approaches that reason directly with natural-language contractual norms have limited accuracy [8]. We address the problem of identifying potential norm conflicts in contracts written in natural language by reasoning over a representation of text in latent space and comparing sentence embeddings [9] (which we review in Section II). By learning the average distance between the embeddings of pairs of conflicting norm sentences against that of non-conflicting ones, we arrive at a notion of normative conflict in latent space in Section III. In Section IV, we show that our results are substantially more accurate than competing approaches in the literature of both traditional [10] and deep learning approaches [8]. Our approach addresses most limitations of previous attempts at reasoning about contract clauses (Section V) and inspires further research into contractual norm reasoning, as we discuss in Section VI.

II. Background

In this section, we introduce concepts related to our approach to identify norm conflicts in contracts, such as norms, contracts, conflicts, word embeddings and sentence embeddings.

A. Norms and Contracts

Deontic logic is a type of modal logic developed to reason about "ideal" worlds from the point of view of compliance with a body of stipulations [11]. Modal logic studies different "modes" of truth, such as possibility, necessity, obligation, knowledge, belief, and perception [12]. These modes are divided into multiple types of modal logic, such as alethic, temporal, epistemic, deontic, among others. Two important modes of truth are what **must be** (necessity) and what **may be** (possibility), respectively represented by the unary operators $\Box$ and $\Diamond$. The assertion $necessarily$ P ($\Box P$, where P is some propositional variable) is true if P is true in all possible worlds. On the other hand, an assertion of $possibly$ P ($\Diamond P$) is true if at least one world exists where P is true.

Deontic logic is a modal logic that allows one to reason about normative expressions [13], comprising notions such as 'obligation' (O), 'prohibition' (F), and 'permission' (P). Thus, deontic logic deals with normative notions of permission, prohibition, and obligation [11]. Consider an action ϕ, permission is the act of allowing a certain behavior by granting some agent to do it without sanctions ($P\phi$). This type of permission is often described in natural language by "it is permitted" [4]. A prohibition is a constraint that indicates an act that must not be performed ($F\phi \equiv O\neg\phi \vee \neg P\phi$). The definition of a prohibition often has a definition of a sanction in case of violation. Finally, an obligation enforces a certain act to be performed by an agent ($O\phi \equiv F\neg\phi \vee \neg P\neg\phi$). Like prohibitions, obligations also have a definition of sanctions for cases when the agent fails to fulfill the obligation.

We use deontic logic to express norm definitions by means of permission, prohibition, and obligation. Norms are mechanisms that regulate expected behaviors from individuals in a specific society or group. They can manage a large number of situations, such as property rights, forms of communication, contracts, and concepts of justice [14].

Sadat-Akhavi [4] introduces two types of norms, namely, mandatory and permissive. A mandatory norm imposes an obligation to the addressee to do or not to do a given act. Mandatory norms that impose an obligation are called obligatory norms and often describe an act that the agent must do. An example of such norm is: "Company X must buy product Y from Company Q in the next three days.". When mandatory norms impose prohibitions, they are called 'prohibitive' norms and often describe a certain behavior that the agent must not perform. For example, "Company X shall not use product W in the formulation of solution Y.". In contrast, permissive norms address to the agent the freedom to do or not to do a certain behavior, which is different from both obligation and prohibition. For example, "Company X may require the expiration date of product Y from Company Q". In this case, Company X may require or not the expiration date and this freedom of choice is given by the permissive norm.

In order to enforce norms, sanctions emerge as a mechanism of social control [1]. The enforcement of social norms is often accomplished through sanctions of various types, so that violating a social norm entails the application of a sanction to the violator to dissuade further violations.

Contracts are semi-structured documents that define and use norms to ensure expected behaviors. A contract is the formalization of a voluntary agreement between two or more parties that is enforceable by law. Contracts have three main components, which define the content and the contracts purpose, namely, *promise*, *payment*, and *acceptance* [15]. Promise in contracts represent a communication of a commitment related to a future intent. The key element in communicating a promise in a contract is a behavioral event, which means a commitment to do (or not do) something. Given a promise, the payment is an element that offers something of value in exchange for the one promised. Finally, the acceptance is the voluntary participation that reflects each party's willingness to make commitments to the other. The transactions described in contracts often involve a seller (whether of goods or services) and a buyer. They occur between individuals and firms in four different categories: firm to firm, individual to firm, firm to individual, and individual to firm [16].

B. Norm Conflicts

As contracts comprise series of norm statements specifying what each party is expected to fulfill, it is important that these statements are logically consistent. Any mistake on specifying the statements of the norms of the contract may lead to conflicts between them. This is particularly true to contracts in natural language since language may be ambiguous and writers of such contracts may overlook subtle logical conflicts. Therefore, it is fundamental to understand how these conflicts arise and what are their configurations. Norm conflicts are the result of a collision between two or more norms due to their stipulations of what ought to be done. As norms describe what is expected during a contract, they use deontic meanings (permission, prohibition, and obligation) to state how parties must behave in each situation. When we have two norms that make impossible to comply with all their requirements, a norm conflict arises. In such case, norms are mutually exclusive since complying with one implies in non-complying with the other, and thus, they cannot exist in a legal order [4].

Sadat-Akhavi [4] 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, a conflict of norms arises "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". Example 2.1 illustrates a norm conflict between an obligation and a prohibition.

Example 2.1:

- The receiving State shall exempt diplomatic agents from indirect taxes.
- The receiving State shall not exempt diplomatic agents from indirect taxes.

The second cause occurs when one norm requires an act, while another norm requires or permits a 'contrary' act. In such case, a norm conflict occurs if "two contrary acts, or if one norm permits an act while the other norm requires a contrary act" [4]. Example 2.2 illustrates the conflict, where both norms indicate different places in which a prisoner of war must be treated. Norm 1 states that it must be done in the prisoner camps, whereas norm 2 states that it must happen in civilian hospitals. The conflict arises when one tries to comply with one norm and is not complying with the other.

Example 2.2:

1) Prisoners of war suffering from disease may be treated in their camps.
2) Prisoners of war suffering from disease shall be treated in civilian hospitals.

The third cause for a conflict between norms is when one norm prohibits a 'necessary precondition' of another norm. Suppose two actions A and B, in the case B cannot be performed without A been performed before. A norm conflict arises when one norm prohibits A and another norm allows B, as Example 2.3 illustrates. In the example, we consider action A as "enter area X" and action B as "render assistance to any person in danger in area X".

Example 2.3:

1) Ships flying the flag of State A shall/may render assistance to any person in danger in area X.
2) Ships flying the flag of State A shall not enter area X.

The fourth cause of norm conflict arises when one norm prohibits a 'necessary consequence' of another norm. Suppose that one cannot perform action B without producing A as result. The conflict arises when one norm obliges B and another norm prohibits A, as Example 2.4 illustrates. If we consider action B as "replace existing rails in area X" and A as the period of time that the line in area X will be hampered, one cannot comply with both norms 1 and 2 in Example 2.4.

Example 2.4:

- State A shall replace existing rails with new ones in area X.
- State A shall not hamper the transport of goods on the existing line in area X.

In this work, we focus on the first and second causes of conflict due to the dataset limitation. This leads us to cases where we have different deontic meanings or different norm structures with the same meaning but conflicting definitions.

C. Word Embedding

There has been much effort in recent years by researchers on Natural Language Processing to map words into low dimensional space in order to capture their lexical and semantic properties [17]–[19]. In order to obtain word embedding methods that convert words into n-dimensional vector representations, or Word2Vec, we can use the internal representations of neural network models, such as feed-forward models [20], recurrent neural network (RNN) models [17], or by low-rank approximation of co-occurrence statistics [19]. In fact, both methods are known to be closely related [21]. Although much effort has been put towards training neural models for word embeddings before [20], the idea of creating shallow methods that are cheaper to train and take the advantage of much larger datasets has become ubiquitous building blocks of a majority of current state of the art NLP applications.

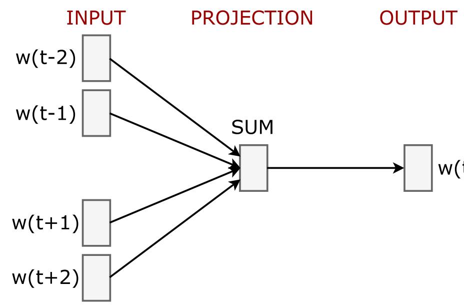

Fig. 1: The CBOW architecture to predict the target word based on the context.

Continuous Bag-of-Words (CBOW) [18] is an efficient algorithm to estimate distributed word representations (Word2Vec). The main assumption of this algorithm is that words used in similar contexts have similar meaning [22]. Thus, CBOW is a neural network architecture composed of three layers (input, hidden layer and output) that learns word representations by predicting a word according to its context. In order to predict the target word, the algorithm scans the corpus sequentially with a fixed-sized window, collecting a central target word and a few neighboring words called the context. The context of the target word is the average of the vectors associated with the target word inside the window, as illustrated in Figure 1. As each word w is represented by a vector v_w and context is represented by the average of word vectors $u_{w'}$ of the context words w', the scoring function is then computed as:

$$s(w, C) = \frac{1}{|C|} \sum_{w' \in C} u_{w'}^{\top} v_w \tag{1}$$

More precisely, given a sequence containing K words, w_1, w_2, ..., w_K, the algorithm maximizes the log probability of a target word given the vectors of the context words as follows:

$$\sum_{t=1}^{K} log\ p(w_t | C_t) \tag{2}$$

In practice, this means minimizing the dot product between the target embedding and the context centroid. Such optimization is often performed via backpropagation [23] using Stochastic Gradient Descent (SGD) where each sample is a window and a loss function is defined between the target and the context vectors. Due to the size of the vocabulary, the softmax function over the scores of contexts and words should not be used for calculating the conditional probability in Equation 2, and instead compute $p(w_t|C_t)$ using independent binary classifiers over words [24]. In order to improve generality, CBOW performs random word sub-sampling as a regularization, deciding to discard each token with certain probability.

Although this model has no knowledge of morphology, syntax or semantics, it can induce word representations with syntactic and semantic properties. As demonstrated by Mikolov *et al.* [25], word embeddings generated by RNN encode not only attributional similarity between words, but also linguistic similarities between pairs of words. Linguistic similarities, also referred to as relational similarities by Turney [26], can capture gender relations such as *man:woman*, *king:queen*, past tense relations such as *capture:captured* and singular/plural relation such as *car:cars*. As such relations are observed as vector offsets between pairs of words sharing a particular relationship (*man - woman* $\approx$ *king - queen*), one could answer the analogy question "*Man is to Woman as King is to* $\langle$*word*$\rangle$", where $\langle$*word*$\rangle$ is unknown, by simply performing vector arithmetic. Thus, one should find the embedding vectors v_{King}, v_{Man}, v_{Woman} and compute $y = v_{\text{King}} - v_{\text{Man}} + v_{\text{Woman}}$, where y is expected to be the continuous space representation of the best answer. As no word might exist in the exact position y, one then has to calculate the distance between the existent embedding vectors to find the closest one. As result, the embedding vector that is closest to y should represent the word *Queen*.

III. Norm Conflict Detection

Since we are dealing with contracts written in natural language, we need to obtain syntactic and semantic information from norm sentences in order to identify norm conflicts. To gather such information, we convert norm sentences into embeddings. This allows us to generate an offset vector that represents norm conflicts and use it to compare embeddings in order to identify conflicts.

Our approach to detect norm conflict consists of three major steps. First, we produce a vector representation to each norm within contracts generating a vector that represents the semantic content of the sentence. Having the embedding for pairs of norms representing a conflict, we create an offset model characterizing all conflicts of our dataset. Finally, given two norms, we can perform the identification of norm conflicts by generating an offset vector of the pair and comparing it with our offset model.

A. Sentence Representation

Although word embeddings work very well to capture linguistic regularities, it still fails when trying to capture the semantic between long sequences of words, such as paragraphs, sentences or documents [24]. Learning such longer representation is central to many natural language applications. Many recent approaches try to capture the semantic and syntactic properties of sentences by creating embeddings [9,24,27]. In this work, we represent norms as embeddings generated by a state of the art sentence embedding algorithm called Sent2Vec [9], see Figure 2(A). This algorithm uses an efficient unsupervised objective to train distributed representations of sentences. It can be understood as an extension of the CBOW [18] that learns the representation of sentences instead of words. According to Pagliardini *et al.* [9], the method can be interpreted as a natural extension of the word contexts from CBOW to a larger sentence context, with the sentence words being specifically optimized towards additive combination over the sentence, by means of the unsupervised objective function.

More precisely, the algorithm learns source v_w and target u_w embeddings for each word w in the vocabulary ($|\mathcal{V}|$) and embedding dimension h. The embedding generated for a sentence is defined as the average of the source word embeddings of its constituent words. Unlike CBOW, Sent2Vec does not consider only unigrams and uses n-grams to create sentence embeddings. The embedding generated for a sentence is defined as the average of the source word embeddings of its constituent words. Thus, the sentence embedding v_S for the sentence S is computed using Equation 3, where $R(S)$ is the list of n-grams (including unigrams) in the sentence.

$$v_S = \frac{1}{|R(S)|} \sum_{w \in R(S)} v_w \tag{3}$$

Similarly to Word2Vec, Sent2Vec improves generality by performing random sub-sampling by deleting random words once all the n-grams have been extracted. Finally, following Mikolov *et al.* [17], missing words are predicted from the context by using a softmax output approximated by negative sampling.

B. Offset Model

As observed by Milokov *et al.* [25] and Levy and Goldberg [21], word embeddings are surprisingly good at capturing syntactic and semantic regularities in language. These regularities can be represented as vector offsets, so that in the embedding space, all pairs of words that share a certain relation are related by the same offset. Using the previous example, the "monarch" relation can be learned by performing the vector operation $v_{\text{Monarch}} = v_{\text{King}} - v_{\text{Man}}$. Adding the embedding of the word *Woman* to the "monarch" embedding $y = v_{\text{Monarch}} + v_{\text{Woman}}$ should result in a vector close to the embedding representing *Queen*. Hence, given two pairs of words that share the same semantic relation $v_a : v_a^*, v_b : v_b^*$, the relation between those two words can be represented as

$$v_a^* - v_a \approx v_b^* - v_b \tag{4}$$

We can see from Equation 4 that the relation between $v_a : v_a^*$ and between $v_b : v_b^*$ generate approximately the same vector offset. Therefore, we learn a $v_{\textit{conflict}}$ vector offset that corresponds to a relation between pairs of embeddings, allowing us to use matrix operations to find similar vectors. Assuming that the linguistic regularities of word embeddings can be extended to sentence embeddings, a vector offset containing the relation between two sentences should hold. Thus, a vector representing the conflict between two sentences (norms) can be learned as a vector offset:

$$v_{\text{conflict}} = \frac{1}{|\mathcal{P}|} \sum_{(n_1,n_2) \in \mathcal{P}} v_{n_1} - v_{n_2} \tag{5}$$

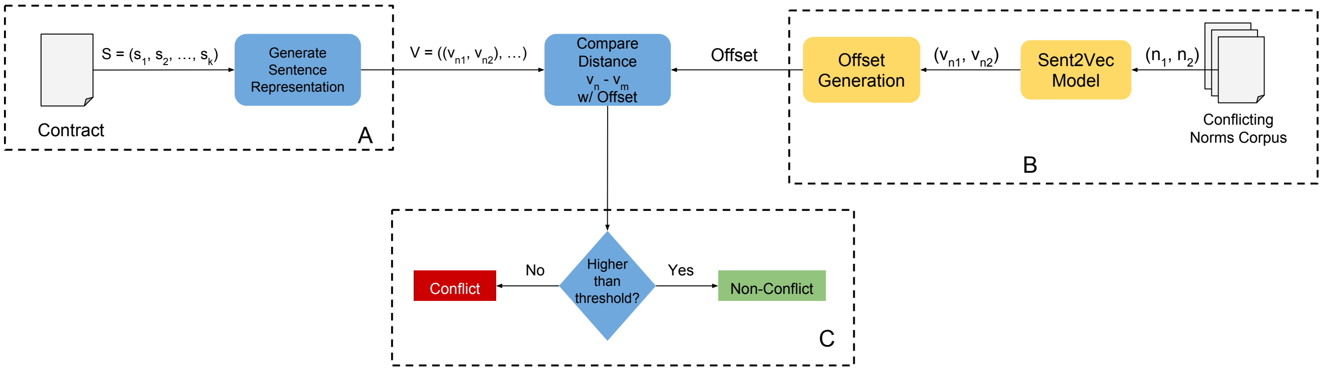

Fig. 2: Pipeline of the norm conflict identifier divided into two steps: the generation of the offset and the identification of norm conflicts using the offset.

where $\mathcal{P}$ represents the set of all norm pairs that contain conflicts, and v_{n_1} and v_{n_2} are the vector embeddings of each norm of the pair.

In order to learn the vector offset (v_{conflict}), we extract pairs of norms (n_1, n_2) from a dataset with known conflicts [28], where n_1 conflicts with n_2. We compute the embedding (v_{n_1}, v_{n_2}) of each norm pair using the Sent2Vec library[1] and a pre-trained model from the Wikipedia English corpus, and use these embeddings to compute a conflict vector (v_{conflict}) as the average offset of all pairs of conflicts in training data, as illustrated in Figure 2(B).

C. Norm Conflict Identification

Having a trained vector offset representing conflicts in contracts, our third step consists of using it and the produced Sent2Vec model to identify norm conflicts in contracts. Our pipeline to identify norm conflicts is illustrated in Figure 2(C) and consists of the following steps for each contract: filter norm sentences from the contract, convert norm sentences into a vector representation using Sent2Vec, compute the offset between the vector representation of each pair of norms, and identify conflicts by computing the distance between the computed offset and the conflict offset v_{conflict} learned with Equation 5. To filter norm sentences from a contract, we use the norm identifier proposed by Aires and Meneguzzi [8], which contains a Support Vector Machine (SVM) model trained with a set of 699 norm sentences and 494 non-norm sentences that obtains an accuracy of 90%. Having the selected norms, we use the Sent2Vec model to convert each norm into its embedding representation. Using the norm representations, we identify a conflict between two norms by comparing their distance in the latent representation to the learned offset against a threshold following Equation 6.

$$\mathcal{C}(v_{n1}, v_{n2}, v_{\textit{conflict}}, t) \begin{cases} \top & \textit{if } d(v_{\textit{conflict}}, v_{n2} - v_{n1}) < \lambda \\ \bot & \textit{otherwise} \end{cases} \tag{6}$$

[1]https://github.com/epfml/sent2vec

Our conflict identifier $\mathcal{C}$ takes four parameters: a pair of norm embeddings (v_{n1} and v_{n2}); an offset vector representation ($v_{\textit{conflict}}$) containing the average value of differences between conflicting norms; and a threshold (λ) that we use to classify the distance of a vector to the $v_{\textit{conflict}}$ as a conflict or not. We consider a conflict ($\top$) when the difference between $v_{\textit{conflict}}$ and the subtraction between v_{n1} and v_{n2} is smaller than λ.

IV. Experiments and Results

In this section, we describe the dataset we use to empirically validate our approach and the experiments performed to compute the accuracy of our approach. We also detail the trained models used to generate our Sent2Vec model as well as the metrics used to evaluate our approach. Finally, we report the results obtained by our approach.

A. Norm Conflict Dataset and Sent2Vec Models

We use the dataset created by Aires *et al.* [28] to evaluate our approach of norm conflict detection in contracts, since it includes real conflicts between norms. This dataset comprises 111 manually annotated norm conflicts from 16 contracts. The annotation process consisted of modifying a copy from an existing norm in a contract such that the resulting pair of norms contains a conflict between the new modified copy and its original version, and then inserting the modified copy back into the contract. The dataset divides norm conflicts into two types, one modifying only the modal verb in the norm and the other modifying both modal verb and norm structure. In the modal verb modification, given a norm with a prohibition deontic meaning, the modified norm turns the verb into one with obligation or permission deontic meaning, thus, creating a conflict. The same occurs for the other types of deontic meanings. The modification of norm structure consists of modifying the norm (words and word order) maintaining the same meaning. Hence, the norm meaning keeps the same while the structure has differences. This is important to identify conflicts that have semantic similarities but use different words to represent it. The non-conflicting norms were obtained by just selecting the norms not used during conflict creation. The resulting set has a total of 204,443 norm pairs.

Each norm in the dataset is transformed into its embedding representation. In order to do that, we use two pre-trained models of Sent2Vec. The Sent2Vec authors made available a series of pre-trained models from different sources, among them there is an unigram and a bigram models trained on Wikipedia. The unigram model produces embeddings with 600 positions, while the bigram model produces embeddings with 700 positions. Both models were trained on English.

B. Approach Results

To measure the performance of our approach, we use a 10-fold cross validation over the norm conflict dataset. Since the generation of the offset uses only the conflicting norms, we divide both sets of conflicting and non-conflicting norms into 10 folds. From the conflicting set, we use 9 folds to generate the offset and the remaining one to test. The test set is then a concatenation between the conflicting fold with its corresponding non-conflicting. We use the same size of conflicts and non-conflicts, so the total number of samples is 222 (111 conflicts + 111 non-conflicts). Each fold has a total of 11 samples and the generation of the offset is made using 99 conflicting samples. Our test set consists of 22 balanced samples. We use the accuracy to measure our results.

In order to compare the results using different distance metrics, we decided to use euclidean distance and cosine similarity. The Euclidean distance (d_{euc}) computes the similarity between two vectors, as shown in Equation 7, where y' is the offset vector representation generated by the pair of sentences from contract. The cosine similarity (d_{cos}) measures the cosine of the angle between two vectors from the same space. Thus, vectors with the same orientation have a similarity of 1, whereas opposite vectors have a -1 similarity. Equation 8 illustrates how we calculate the cosine similarity. In order to obtain positive values (from 0 to 2), we use the cosine distance, which can be obtained by applying $1 - d_{cos}$.

$$d_{euc}(v_{conflict}, y') = ||v_{conflict} - y'||^2 \tag{7}$$

$$d_{cos}(v_{conflict}, y') = \frac{v_{conflict} \cdot y'}{||v_{conflict}|| \; ||y'||} \tag{8}$$

We evaluate our approach using difference distance metrics and pre-trained models. We use euclidean and cosine distance in both unigram and bigram models. To see what is the best threshold to divide conflicts from non-conflicts, we vary its value according to the distance metric applied. Table I shows the accuracy results using euclidean distance in both unigram and bigram. We vary the threshold (λ) from 0 to 3 based on the maximum distance non-conflict cases obtained during tests. As we can see, we get the best results for all folds using the unigram model and a threshold of $\lambda = 2$.

To generate a better visualization of the results considering the variation of the threshold, we created the heat map shown in Figure 3 with the accuracy for both unigram and bigram models using the cosine similarity. In this case, we vary the threshold value (λ) from 0 to 2 as they are, respectively, the minimum and maximum values when using the cosine similarity. As Figure 3 illustrates, we obtain the best results using a threshold between $\lambda = 0.4$ and $\lambda = 0.8$. This range of threshold shows the best separation between conflicting and non-conflicting cases. The boundaries (0 and 2) obtain poor accuracy since can only correctly classify half of the test samples. Overall, we obtain the best result using a bigram model with a threshold of $\lambda = 0.8$. When comparing the distances used, we can see that both have similar results with opposite models. The euclidean distance obtains better results using the unigram model, whereas the cosine distance obtains better results using the bigram model.

Table II compares the results of our approach with related work on the dataset we used in our experiments. We use the best result obtained by the mean of folds, which is the euclidean distance with unigram model and a threshold of $\lambda = 2$. Although using just the distance between embedding vectors, our approach overcomes approaches using rules [10] and deep learning models [8].

	Threshold Unigram				Threshold Bigram			
Fold	0	1	2	3	0	1	2	3
0	0.5	0.77	**0.95**	**0.95**	0.5	0.86	0.82	0.50
1	0.5	0.77	**0.95**	0.91	0.5	0.91	0.86	0.59
2	0.5	0.91	**0.95**	0.82	0.5	0.91	0.91	0.64
3	0.5	0.73	**0.91**	**0.91**	0.5	0.86	0.82	0.50
4	0.5	0.85	**0.95**	0.85	0.5	**0.95**	0.85	0.50
5	0.5	0.90	**1.00**	**1.00**	0.5	0.90	0.95	0.50
6	0.5	0.85	**0.95**	0.90	0.5	**0.95**	0.85	0.55
7	0.5	0.70	**0.95**	0.80	0.5	0.90	0.85	0.55
8	0.5	0.75	**0.90**	0.85	0.5	**0.90**	0.70	0.55
9	0.5	0.90	**0.95**	0.90	0.5	**0.95**	0.85	0.50
Mean	0.5	0.81	**0.95**	0.89	0.50	0.91	0.85	0.54

TABLE I: Results obtained using both unigram and bigram models with euclidean distance

Approaches	Accuracy
Aires *et al.* [10]	0.78
Aires and Meneguzzi [8]	0.84
Offset Conflict	**0.95**

TABLE II: Comparison between accuracies from existing work and our approach

V. Related Work

Among the existing work involving the identification of normative conflicts in natural language, most of them involve the use of a controlled language. Rosso *et al.* [5] manually translate norms in contracts to a formal contract language in order to perform conflict detection. Using such contract language, they use a series of predefined rules to identify the norm conflicts. They test the approach in a controlled example

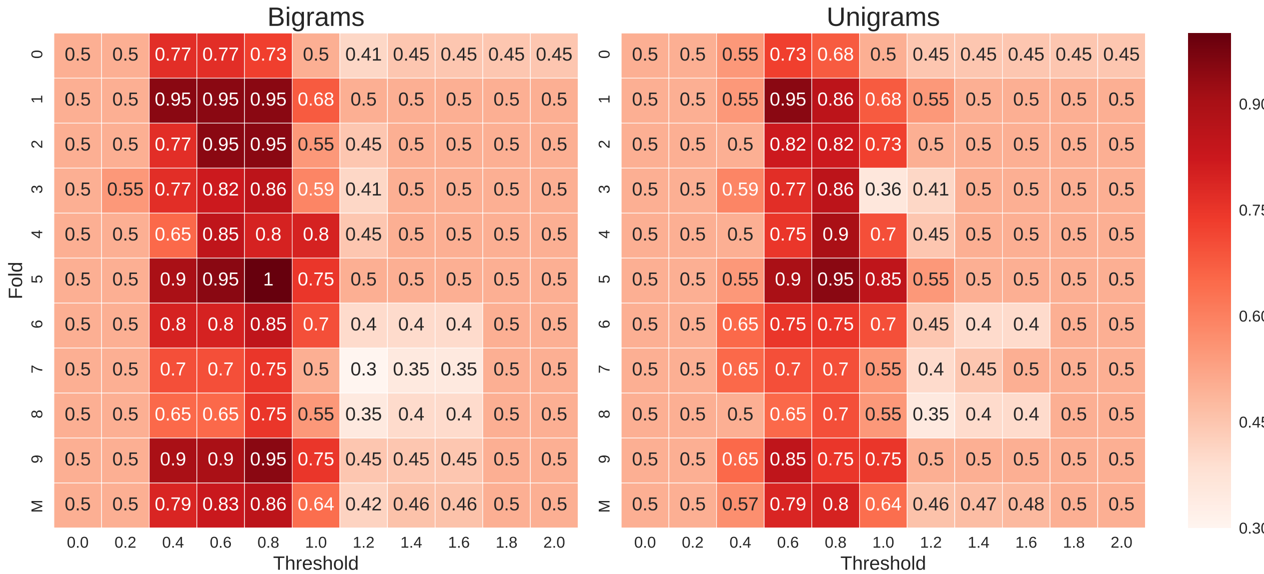

Fig. 3: Accuracy obtained using the cosine distance for different folds and thresholds. "M" represents the Mean value for the specific threshold values for the Mean.

generated by them and identify the conflicts on it. Using the same logic, Azzopardi *et al.* [7] propose an automatic approach to translate natural language to a deontic logic representation that allows to identify certain types of conflicts. To detect norm conflicts, they compare the following conflict cases: permission and prohibition of the same action; obligation and prohibition of the same action; obligation of two mutually exclusive actions; and obligation and permission of mutually exclusive actions. They tested their automatic translation over a manually annotated set of 5 real contracts from the Australian Contract Corpus [29] obtaining an accuracy of 0.62. Our proposed approach differs from these works by identifying conflicts directly from natural language. Instead of using a logical representation that demands rules to adapt from natural language, we use sentence embeddings that preserve syntactic and semantic information from text. We did not compare our results to theirs because we do not have access to the contract that Rosso *et al.*used, and Azzopardi *et al.*do not use the contracts from the Australian Contract Corpus to identify conflicts, instead, they use it to test their approach to translate norms into a deontic logic representation.

Previous work by Aires *et al.* [10] propose a rule-based approach to identify norm conflicts in contracts written in natural language. They extract specific information from norms dividing them into three parts: party, modal verb, and norm action. The identification of conflicts consists of comparing such information in order to identify conflicting elements. Using the same dataset we use in this work, they obtained an accuracy of 78%. Aires and Meneguzzi [8] use a convolution neural network (CNN) to identify conflicts between norms. The authors convert norm pairs into binary matrices that are used as input to the CNN. The CNN output classifies a pair of norm as conflict or not. Using the same dataset we use in this work, they obtained an accuracy of 84%. Aires *et al.* and Aires and Meneguzzi have similar work to ours, the main difference to these approaches is that we do not rely on rules or on a model training using CNN. Instead, our sentence embeddings allow us to create an offset vector containing the conflict embedding, which yields better results over the same dataset.

VI. Conclusion

We developed an approach to detect conflicts between norms in contracts using sentence embeddings. Our main contribution is the creation of an offset vector that contains the average distance between embeddings from conflicting norms. We use the offset vector to identify norm conflicts by calculating the distance between the offset vector and the resulting vector from the difference between two norm embeddings. Our experiments show that the use of an offset vectors surpass existing results for the task of norm identification. Using a dataset containing norm conflicts and non-conflicting norms, our approach obtains an accuracy of 95%, surpassing the previous state of the art by 11%.

As future work, we aim to perform three main experiments. First, we intend to use the offset vector to generate conflicting norms by adding it to a norm embedding. To do so, we aim to train a recurrent neural network able to generate natural language from embeddings. Second, we aim to use the offset to identify the specific parts of a norm sentence that causes a conflict. It allows us to identify conflicting cases using only segments of norm embeddings. Finally, we aim to extend the current available dataset to apply our approach on more complex conflict cases involving the third and fourth cases defined by Sadat-Akhavi.

ACKNOWLEDGEMENT

The authors would like to thank CNPq and NVIDIA for funding this research.

REFERENCES

[1] M. L. Andersen and H. F. Taylor, *Sociology: Understanding a Diverse Society, Updated*, 4th ed. Cengage Learning, 2007. [Online]. Available: https://books.google.com.br/books?id=UILcQZS6Bi4C

[2] O. D. Hart and B. Holmstrm, "The theory of contracts," in *Advances in Economic Theory: Fifth World Congress*, T. F. Bewley, Ed. Cambridge University Press, 1989, pp. 71–156. [Online]. Available: http://assets.cambridge.org/97805213/89259/frontmatter/9780521389259_frontmatter.pdf

[3] F. Meneguzzi, S. Miles, M. Luck, C. Holt, and M. Smith, "Electronic contracting in aircraft aftercare: A case study," in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track*, ser. AAMAS'08. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 63–70. [Online]. Available: http://dl.acm.org/citation.cfm?id=1402795.1402807

[4] S.-A. Sadat-Akhavi, *Methods of Resolving Conflicts Between Treaties*, ser. Graduate Institute of International and Development Studies - Volume: 3. Martinus Nijhoff, 2003. [Online]. Available: https://books.google.com.br/books?id=Dv0QZWUSGJcC

[5] P. Rosso, S. Correa, and D. Buscaldi, "Passage retrieval in legal texts," *The Journal of Logic and Algebraic Programming*, vol. 80, no. 3, pp. 139–153, Jul. 2011. [Online]. Available: https://doi.org/10.1016/j.jlap.2011.02.001

[6] X. Gao and M. P. Singh, "Mining contracts for business events and temporal constraints in service engagements," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 427–439, Jul. 2014. [Online]. Available: http://dx.doi.org/10.1109/TSC.2013.21

[7] S. Azzopardi, A. Gatt, and G. J. Pace, "Integrating natural language and formal analysis for legal documents," in *Proceedings of the Conference on Language Technologies and Digital Humanities*, ser. JTDH 2016. Academic Publishing Division of the Faculty of Arts, 2016, pp. 32–35. [Online]. Available: http://www.sdjt.si/jtdh-2016/en/

[8] J. P. Aires and F. Meneguzzi, "Norm conflict identification using deep learning," in *AAMAS'17 Workshop on Coordination, Organisation, Institutions and Norms in Multi-Agent Systems*, ser. COIN 2017. Springer International Publishing, 2017, pp. 194–207. [Online]. Available: https://doi.org/10.1007/978-3-319-71679-4_13

[9] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," *arXiv:1703.02507*, 2017. [Online]. Available: http://arxiv.org/abs/1703.02507

[10] J. P. Aires, D. Pinheiro, V. S. de Lima, and F. Meneguzzi, "Norm conflict identification in contracts," *Artificial Intelligence and Law*, vol. 25, no. 4, pp. 397–428, Dec. 2017. [Online]. Available: https://doi.org/10.1007/s10506-017-9205-x

[11] J. Carmo and A. J. Jones, "Deontic logic and contrary-to-duties," in *Handbook of Philosophical Logic: Volume 8*, D. M. Gabbay and F. Guenthner, Eds. Springer, 2002, pp. 265–343. [Online]. Available: http://dx.doi.org/10.1007/978-94-010-0387-2_4

[12] E. A. Emerson, "Temporal and modal logic," in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, J. Van Leeuwen, Ed. MIT Press, 1990, pp. 995–1072. [Online]. Available: http://dl.acm.org/citation.cfm?id=114891.114907

[13] R. Hilpinen, *Deontic Logic: Introductory and Systematic Readings*, ser. Synthese Library. Springer, 1971. [Online]. Available: http://dx.doi.org/10.1007/978-94-010-3146-2

[14] M. A. Mahmoud, M. S. Ahmad, M. Z. Yusoff, and A. Mustapha, "A review of norms and normative multiagent systems," *The Scientific World Journal*, vol. 2014, p. p. 23, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1155/2014/684587

[15] D. M. Rousseau and J. M. Parks, "The contracts of individuals and organizations," *Research in Organizational Behavior*, vol. 15, pp. 1–43, Jan. 1993. [Online]. Available: http://web.mit.edu/curhan/www/docs/Articles/15341_Readings/Justice/TheContractsOfIndividualsAndOrganizations_RosseauParks.pdf

[16] A. Schwartz and R. E. Scott, "Contract theory and the limits of contract law," *The Yale Law Journal*, vol. 113, no. 3, pp. 541–619, Dec. 2003. [Online]. Available: http://www.jstor.org/stable/3657531

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, ser. NIPS 2013. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999792.2999959

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781, 2013a.*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[19] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP'14. Association for Computational Linguistics, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[20] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, March 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944966

[21] O. Levy and Y. Goldberg, "Linguistic regularities in sparse and explicit word representations," in *Proceedings of the 18th Conference on Computational Natural Language Learning*, ser. CoNLL 2014. Association for Computational Linguistics, 2014, pp. 171–180. [Online]. Available: http://www.aclweb.org/anthology/W14-1618

[22] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 23, pp. 146–162, 1954. [Online]. Available: http://dx.doi.org/10.1080/00437956.1954.11659520

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–538, October 1986. [Online]. Available: http://dx.doi.org/10.1038/323533a0

[24] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, ser. ICML'14. JMLR.org, 2014, pp. II–1188–II–1196. [Online]. Available: http://dl.acm.org/citation.cfm?id=3044805.3045025

[25] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ser. NAACL-HLT 2013. Association for Computational Linguistics, 2013, pp. 746–751. [Online]. Available: http://www.aclweb.org/anthology/N13-1090

[26] P. D. Turney, "Similarity of semantic relations," *Computational Linguistics*, vol. 32, no. 3, pp. 379–416, September 2006. [Online]. Available: http://dx.doi.org/10.1162/coli.2006.32.3.379

[27] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016. [Online]. Available: https://arxiv.org/abs/1612.03651

[28] J. P. Aires, D. Pinheiro, and F. Meneguzzi, "Norm dataset: Dataset with norms and norm conflicts," March 2017. [Online]. Available: https://doi.org/10.5281/zenodo.345411

[29] M. Curtotti and E. C. McCreath, "A corpus of australian contract language: Description, profiling and analysis," in *Proceedings of the 13th International Conference on Artificial Intelligence and Law*, ser. ICAIL '11. ACM, 2011, pp. 199–208. [Online]. Available: http://dx.doi.org/10.1145/2018358.2018387