PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

MULTI-MODEL APPROACH TO IDENTIFY POTENTIAL PROBLEMS IN A CONTRACT.

ALEXANDRE YUKIO ICHIDA

Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Prof. Felipe Rech Meneguzzi

Porto Alegre
2020

REPLACE THIS PAGE WITH THE LIBRARY CATALOG PAGE

REPLACE THIS PAGE WITH
THE COMMITTEE FORMS

I dedicate this work to my family and my wife.

“Once you know what failure feels like, determination chases success.”
(Kobe Bryant)

ACKNOWLEDGMENTS

I would like to acknowledge my wife Jessica Rodrigues da Silva for all her support in my life during the entire master program. I would like to acknowledge my manager and work colleagues for their support and comprehension during my work absences.

I would like to thank my advisor Felipe Meneguzzi for all guidance, scientific writing lessons, insights, and ideas. To João Paulo de Souza Aires for sharing his thoughts and his works about norm processing. To anonymous reviewers of AAAI-2020 and ECAI-2020 for providing feedback on our papers allowing us to improve this work.

ABORDAGEM MULTI-MODELO PARA IDENTIFICAÇÃO DE POTENCIAIS PROBLEMAS CONTRATUAIS.

RESUMO

Os contratos sustentam a maioria das transações comerciais modernas, definindo os deveres e obrigações das partes relacionadas em um contrato, e garantir que esses contratos estejam livres de erros é crucial para a sociedade moderna. A análise de um contrato requer a compreensão das relações lógicas entre cláusulas e a identificação de possíveis contradições, que, por sua vez, dependem de esforços humanos para entender cada cláusula no qual são suscetíveis a erro. Neste trabalho, desenvolvemos uma abordagem para automatizar essas análises, identificando relações lógicas e detectando possíveis conflitos nas cláusulas contratuais. A abordagem resultante deve ajudar os autores do contrato a detectar possíveis conflitos lógicos entre as cláusulas.

Palavras-Chave: Contratos, Aprendizado de Máquina, Redes Neurais Artificiais, Sistemas Normativos.

MULTI-MODEL APPROACH TO IDENTIFY POTENTIAL PROBLEMS IN A CONTRACT.

ABSTRACT

Contracts underlie most modern commercial transactions defining the duties and obligations of the related parties in an agreement, and ensuring such contracts are error-free is crucial for modern society. The analysis of a contract requires understanding the logical relations between clauses and identifying potential contradictions, which, in turn, depends on error-prone human effort to understand each clause. In this work, we develop an approach to automate such analyses identifying logical relations and detecting potential conflicts in contract clauses. The resulting approach should help contract authors detecting potential logical conflicts between clauses.

Keywords: Contracts, Machine Learning, Neural Networks, Normative Systems.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

NLI – Natural Language Inference

SNLI – Stanford Natural Language Inference

MCP – McCullock-Pitts

MLP – Multi-layer Perceptron

MSE – Mean Square Error

LSTM – Long Short Term Memory

GRU – Gated Recurrent Unit

NCC – Norm Conflict Classification

BPE – Byte-Pair Encoding

OOV – Out-of-Vocabulary

GPU – Graphic Processing Units

PMI – Point-wise Mutual Information

DS – deontic-structure

DM – deontic-modality

DO – deontic-object

OC – object-conditional

CONTENTS

1. INTRODUCTION

Understanding existing logical relations between sentences is a difficult task that requires an accurate understanding of meaning of the underlying natural language. The ambiguity and variability of linguistic expression in natural language complicates the recognition of these relations such as entailment and contradiction contained in texts. The ability to classify these logical inferences among different text is a significant feature of an intelligent system [5]. Detecting these logical relations can be useful to help humans to interpret a more complex text, where entailment and contradiction are crucial aspects to fully understanding such as norms and contracts.

Contracts are documents that contain normative sentences formalizing agreements among the related parties, which involve people and companies. The normative sentences describe the duties that the related parties are subject to and the penalties in case of rule violation. In a contract, the norms may contain logical relation between them such as entailment, contradiction or a neutrality of obligations [12].

For instance, in a contract that contains the following norms “All companies must pay the Y tax” and “The company X must pay the Y tax”, it is logically not possible to satisfy the first norm without satisfying the second norm. In the case of company X not paying the tax Y, automatically violates both norms due to the conditions of compliance. Since both norms are logically linked and are in the same context, we have an entailment relation between them.

By contrast, conflicts in a contract may emerge through problems related to a logical contradiction between norm clauses. Taking the example above, we have a contradiction relation if we change the second norm to “The company X must not pay the tax Y” due to their contradictory compliance condition. Analyzing these conflicts demands a careful analysis of all parties involved in a contract. An automated way to detect a conflict between contract clauses addresses these reviews of contract clauses, which is a long and complex issue even for human experts.

The problem of classifying the logical relation between norms is analogous to Natural Language Inference (NLI), which is the task of determining whether a natural language hypothesis h can be inferred from a natural language premise p [22]. In an entailment relation, if p is true then h cannot be false, otherwise is a contradiction relation. Natural Language Inference is a broader task than conflict identification, and thus, good models to classify logical relations will naturally be applicable to detect contract conflicts. Importantly, since NLI has seen a surge in research, including new machine learning models and dataset curation [6, 37], it offers substantial labelled training data in much larger quantities than purely contract conflict datasets [4].

In this work, we provide an automated way to detect normative conflicts in contracts by analyzing the inferential relations between contract clauses. We detect such conflicts using a multi-model approach, developed in Section 3 composed of models that deal with two specific tasks: natural language inference and norm conflict classification. For the first task, we develop an NLI model that identifies the logical relation between normative sentences, whereas for the second task we develop a state-of-the-art norm conflict classifier that detects the conflict type between two normative sentences. We show that the new model surpasses the old state of the art by a substantial margin in Section 4. We perform an experimental analysis in Chapter 5, we show that the combination of such models can help to further understand potential contractual problems, for example, identifying transitive conflicts.

2. BACKGROUND

In this chapter, we detail the concepts that we use in this work and the problem formulation. First, we report a brief introduction about norms and contracts detailing their basic concepts. Second, we explain the main concepts of natural logic and natural language inference detailing which inferential relations we can extract in sentences written in natural language. Finally, to understand our developed approach, we explain concepts of machine learning methods reporting novel neural network methods to deal with natural language.

2.1 Norms and Contracts

Norms are statements that indicate a judgment about an expected behavior imposing rules and ought in a society. The main objective of norms is to determine what an agent (human or artificial) should and should not do, according to established rules [7]. A normative sentence is characterized by modal verbs to make explicit the ought of each agent involved.According to Griffiths [16], there are two types of norm: Informal norms and Formal norms.

Informal norms do not need to be clearly specified and can be grounded by behavior observations and reproduction [16]. The violation of an informal norm may not have a penalty due to these norms are implicit and subject to ambiguity. Dress code and other personal interaction rules are examples of informal norms, which are commonly related to human behavior.

By contrast, formal norms are explicit and must be clearly detailed. Formal norms have penalties attached to the cases of violation to ensure its fulfillment. Contract clauses and laws are examples of formal norms, which are most strictly enforced and should be clearly stated. For instance, consider the formal normative statement defined by the government "The company X should pay 10 percent of his incoming for the government". Regardless of the reason, in the case of *Company X* violates this norm, the government can apply sanctions and fines for the company X. This example enforces that formal norms should be clear to avoid any possible misunderstanding.

2.1.1 Contract

Contracts are documents that define the duties and obligations between two or more people in an agreement. Organizations use contracts with themselves to make an economic exchange, which may involve goods, services or money. A contract contains a set of formal norms, known as clauses, which defines the constraints and the penalties in case of a clause violation. According to Rousseau [29], contracts contains the following components: *promise*, *payment*, and *acceptance*.

In a contract, a promise is the communication of a commitment to do something. The key element of contract premise is the communication of a future intent through statements that describe obligations of each organization. The payment of a contract represents the offer made by the other party, which is considered a promise as well [29]. Given the fact that the organizations sign the contract after the consensus regarding their obligations, the acceptance represents the voluntary participation of each party, which symbolizes their willingness to make commitments to the other.

Understanding all norms in a contract is essential to all involved parties in order to comply with commitments defined. Besides that, all norms must be clear and well-defined to avoid potential conflicts in a contract. In this work, we focus on logical relations between normative sentences in a contract dealing with aspects of natural language processing such as ambiguity.

2.1.2 Norm Conflict Types

To further help contract writers to understand the nature of conflicts, Aires *et al* [1] introduce a typology of norm conflicts that specifies their causes. Given a norm pair, this typology relies on inconsistency in their deontic modalities, their normative action, and their conditions. This typology contains four types of norm conflicts: deontic-modality, deontic-structure, deontic-object and object-conditional.

The *deontic-modality* conflict type indicates conflicts originated by the deontic statement of each clause, i.e., prohibition × obligation, obligation × permission, and permission × prohibition. *Deontic-structure* conflict types involves different deontic meaning but with different sentence structure. *Deontic-object* conflict occurs when the action or specification of the two norms are different, with the same deontic meaning. The *object-conditional* conflict occurs when the condition to perform a norm is conflicting with another. Table 2.1 shows examples of norm pairs contained in norm conflict dataset with their respective conflict types.

In the first row, although both norms are similar, they contain different deontic meanings for the same action and the same subject. Similar to the first row, the second norm pair contains different deontic meaning with a different sentence structure. The third row shows an example when the conflict arises from the norm object, which indicates two different dates in their specification for the same subject and the same modality. The last row shows an example where the conflict arises from the condition since if the condition imposed by the second norm is not satisfied, the first norm is conflicting with the second.

Table 2.1 – Examples of norm pairs with the respective conflict type.

Norm Pair	Conflict Type
- The Specifications may be amended by the NCR design release process. - The Specifications shall not be amended by the NCR design release process.	deontic modality
- All inquiries that Seller receives on a worldwide basis relative to Buyer's air chamber "Products" as specified in Exhibit III, shall be directed to Buyer. - Seller may not redirect inquiries concerning Buyer's air chamber "Products".	deontic structure
- Autotote shall make available to Sisal one (1) working prototype of the Terminal by May 1, 1998. - Autotote shall make available to Sisal one (1) working prototype of the Terminal by June 12, 1998.	deontic object
- The Facility shall meet all legal and administrative code standards applicable to the conduct of the Principal Activity thereat. - Only if previously agreed, the Facility ought to follow legal and administrative code standards.	object conditional

2.2 Natural Logic and Inference

2.2.1 Natural Logic

Natural Logic is a model that describes logical inferences over natural language representation. Given that most of the reasoning is done in natural language by humans and most uses of natural language express reasoning of some sort, this model aims to create a correspondence between logical and grammatical structure[20]. Although Lakoff developed the main concepts of Natural Logic [20], Aristotle uses a similar approach to introduce syllogisms through logical arguments represented by natural language. Recently, Natural Logic model was revisited by Valencia [34] that uses monotonic properties to explain inferences and MacCartney *et al.* [23] introduce semantic exclusion relations.

Valencia work [34] develops a Natural Logic model based on a mechanism that applies monotonicity calculus. In the Natural Logic context, monotonicity calculus describes entailment as semantic containment relation between natural language texts, which is analogous to the set containment relation. Therefore, this analogy allows classifying bi-directional relation distinguishing forward entailment and reverse entailment. For example, the word pair ("crow", "bird") is a forward entailment whereas the pair ("European", "French") is a reverse entailment. However, the approach of monotonicity calculus cannot represent semantic exclusion, which represents contradiction meaning. Thus, using many simple inferences fails using monotonicity calculus such as "Stimpy is a cat" $\models$ "Stimpy is not a poodle".

MacCartney *et al.* [23] develop an extension of Natural Logic to incorporate semantic exclusion in monotonicity calculus. Following Valencia, this work formalizes an inventory of relations that represents semantic containment including two new relations to represent semantic exclusion: negation and alternation. Negation is analogous to set complement while alternation is analogous to exclusive disjunction. MacCartney *et al* [23] uses the $\wedge$ symbol to evoke the logically similar bitwise XOR operator of the C programming language family, although its is the same symbol used to represent a logical conjunction. This extension includes a cover relation to represent non-exhaustive exclusion (non-equivalence). Table 2.2 describes the symbol defined by MacCartney *et al.* [23] that represents each relation with an example.

Table 2.2 – Inventory of semantic relations of MacCartney *et al.* [23] extension of Natural Logic

Symbol	Name	Example
x $\equiv$ y	equivalence	sofa $\equiv$ couch
x $\sqsubset$ y	forward entailment	crow $\sqsubset$ bird
x $\sqsupset$ y	reverse entailment	European $\sqsupset$ French
x $\wedge$ y	negation	human $\wedge$ nonhuman
x $\mid$ y	cover	cat $\mid$ dog
x # y	independence	hungry # hippo

2.2.2 Natural Language Inference

Automated reasoning and inference are essential topics of artificial intelligence. Natural language inference (NLI) is a widely-studied natural language processing task that is concerned with determining the inferential relation between a premise *p* and a hypothesis *h* [6]. In NLI, the entailment relation inferred is formulated based on the following representations: two-way classification and three-way classification [22].

Two-way classification is the simplest representation of NLI, which describes the task as a binary decision. The objective of this NLI task is to classify whether the hypothesis follows the premise (entailment) or does not (non-entailment). This classification form were used in the RTE competition [9]. This task representation cannot detect a contradiction between *p* and *h* due to the inference of a non-entailment does not specify the relation that differs from an entailment.

Alternatively, the three-way classification form deals with contradiction creating an extra category of inferred relations. In three-way classification form, the relations are divided into three categories: entailment, contradiction and neutral. Given a pair of premise-hypothesis *p* and *h*, the *entailment* relation occurs when *h* can be inferred from *p* [22]. When *h* infers the negation of *p*, the pair results in a *contradiction*. Otherwise, if none of these relations can be inferred, the relation of *p* and *h* is *neutral*.

In NLI, both *p* and *h* are sentences written in natural language. The challenge of this task differs of formal deduction from logic due to deal with informal reasoning [22]. The emphasis of the NLI is on aspects of natural language such as lexical semantic knowledge and the deal with the variability of linguistic expression. Consider the following premise *p* and hypothesis *h* as an instance of an NLI scenario [22]:

- *p*: Several airlines polled saw costs grow more than expected, even after adjusting for inflation.
- *h*: Some of the companies in the poll reported cost increases.

In the NLI context, this example is considered a valid entailment inference because any person that interprets *p* would likely accept that *h* implies in the information of *p*. Although is a valid NLI classification, *h* is not a strict logical consequence of *p* due to the fact that *p* informs that airline companies **saw** the growth of the cost, not necessarily **reporting** the growth of the cost. This example reflects the informal reasoning of the task definition due to deal with ambiguity of natural language [22].

2.3 Machine Learning and Neural Networks

2.3.1 Machine Learning Overview

Machine learning is the field of study that covers computational models and algorithms that give computers the ability to improve through experience [25]. These methods focus on detecting patterns and using the discovered patterns to predict future information to help in decision making under uncertainty [26]. Mitchel [25] states the following about machine learning: “A computer program is said to learn from the experience E with respect to some class of task T and performance measure P if its performance at task in T, as measured by P, improves with experience E”. A machine learning model contains a set of parameters supplied by the input information, known as features. There are the following types of machine learning methods: supervised learning and unsupervised learning.

The objective of supervised learning is to learn a function using a training dataset that contains the expected function result given a set of input values. A supervised model uses the expected value of the training data to measure the error compared with the inferred value through a cost function. The model improves its performance considering the result of the cost function by minimizing the error. For example, given an house price prediction task and a dataset with enough data, we can create a supervised model using *house size* and *number of rooms* as feature and the *house price* as expected function value.

On the other hand, the unsupervised model objective is to learn a function using a training dataset that does not contains the expected function result. An unsupervised model relies only on features of the training dataset to predict the results. Clustering is an example task of unsupervised learning, which classify the input value based only on the input features and the classes are not known beforehand.

Machine learning problems are categorized based on the expected output of the model [30]. When the output value of the model is a finite set of values, the learning problem is called *classification*. The objective of a classification task is to predict a class given a input value. When the output of the model represents a continuous number value, the learning problem is called *regression*. House price prediction is an example of a regression task due to the predicted price is a real number.

2.3.2 Neural Networks

Neural network is a computational model for machine learning inspired by the biological neural networks of the animal brains. In a neural network, the input information flows through intermediate computations, which are called activation functions, and finally to the model output [15]. This model is associated with a directed acyclic graph describing how these activation functions are composed together[15]. The nodes of the graph are named *artificial neurons* and each edge contains an associated real number, which is called *weight* [15]. The number of artificial neurons measures the dimensionality of its input data, which is represented by a numeric vector. A neural network is composed of one input layer, one or more hidden layers and one output layer.

The input layer describes the input data, where the neurons represent each feature of the model. The hidden layer is an intermediary network tier that receives the values from the input layer or the others hidden layers and executes an activation function. Each hidden layer may contains an extra neuron called *bias*, which propagates a fixed value for the next layers. The output layer receives the result of hidden layer and produces the output of the neural network.

The number of layers of a neural network architecture represents the *depth* of the model [15]. The *feed forward neural network* is a neural network architecture that can have multiple layers with fully connected neurons, which a weight value associated with each neuron connection. The Figure 2.1 shows an example of feed forward neural networks, given the model feature set *x* and the model output *y*.

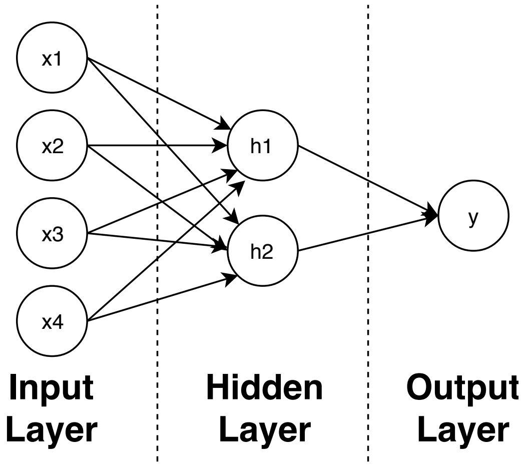

Figure 2.1 – A feed forward neural network with the input neurons *x*, hidden neurons *h* and the output neuron *y*.

2.3.3 Activation Functions

The activation function is the function that converts the input values of a layer to an output value to feed the next layers. In a neural network, we use the sum of products of the input values with their respective weights as the input of activation function. In this subsection, we cover two instances of activation functions: sigmoid and hyperbolic tangent (tanh). The Equation 2.1 shows an example of an activation function *a* that receives the sum of products *z* given the input value *x* and weights *w*.

$$\begin{aligned} z &= \sum_i w_i \cdot x_i \\ activation &= a(z) \end{aligned} \tag{2.1}$$

Sigmoid

Sigmoid is a mathematical function that produces values between 0 and 1, resulting in an S-shaped curve. A neural network uses the sigmoid as an activation function to squash their inputs into a value that represents a probability [15]. The Equation illustrates the sigmoid function σ receiving *z* that is the sum of products between the layer input and their weights.

$$\sigma = \frac{1}{1 + \exp(-z)} \tag{2.2}$$

The sigmoid function results in values between 0 and 1, which saturates when its argument approaches positive infinity or negative infinity in the limit. In the case when the sum is extremely high, the sigmoid saturates to 1 and when it is very low the function saturates to 0. Figure 2.2 shows the shape of sigmoid and the saturation regions.

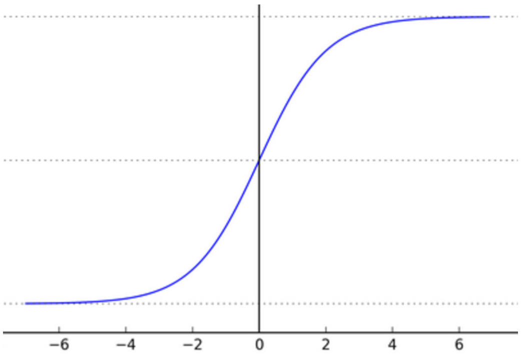

Figure 2.2 – S-shape formed by sigmoid function [15]

Hyperbolic Tangent

The Hyperbolic Tangent (*tanh*) is a non-linear function that forms an S-shaped curve similar to the sigmoid function. *Tanh* function results in values between -1 and 1, which saturates when its argument approaches positive infinity or negative infinity in the limit. A neural network can use *tanh* as the activation function that can control the increase or decrease of values in a hidden state [15]. The Equation 2.3 illustrates the *tanh* function receiving the argument *z* and the Figure 2.3 shows the shape and its saturation region.

$$tanh(z) = \frac{\epsilon^{2z} - 1}{\epsilon^{2z} + 1} \tag{2.3}$$

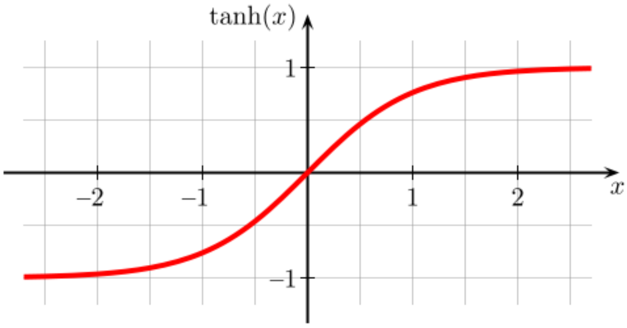

Figure 2.3 – S-shape formed by *tanh* function.

Softmax

Softmax is a numerical function that represents a normalized probability distribution [26]. Formally, given a set of values $x \in \mathbb{R}$ with length K, for each element of x, the softmax results in probability values normalized in the interval (0,1). The softmax function applies an exponential function of element x_i and normalizes it dividing by the sum of the exponential of all elements in x. Neural network models that deal with multi-class classification tasks use softmax as an activation function of its output layer to represent each K class probability [15]. Equation 2.4 shows the softmax function of the i-th element of set x.

$$softmax(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{2.4}$$

2.3.4 Backpropagation

The Backpropagation is an algorithm that adjusts the weights of a neural network through error minimization between the predicted and the expected value [25]. The algorithm uses the measured error to adjust weights using the *Chain Rule* of calculus to compute the derivatives of activation functions of the neural network layers [15]. This algorithm is divided in two steps: forward propagation and backward propagation.

In forward propagation, the algorithm executes a prediction over input values computing the activation functions of all the network layers. The input of activation function is the result of the matrix multiplication between the layer input values and his weights including the sum with bias value. Equations 2.5 and 2.6 show the operations processed in a neural network layer, where the variable j represents the neuron index of the layer l, the variable k represents the neuron index of the previous layer $(l-1)$ and the operation a_j^l represents the activation function σ.

$$z_j^l = \sum_k w_{j,k}^l \cdot a_k^{l-1} + b \tag{2.5}$$

$$a_j^l = \sigma(z_j^l) \tag{2.6}$$

After forward propagation reaches the output layer predicting the neural network output, then the backward propagation is executed. This step of algorithm relies on the Chain Rule to propagate the error measured from the output layer to all of neural network weights. Equation 2.7 shows the Chain Rule applied in the backward propagation with variable J representing the cost function and variable δ^L representing the error to be back propagated from the output layer L.

$$\begin{aligned} \delta^L &= \frac{\partial J}{\partial z^L} \\ &= \frac{\partial J}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \end{aligned} \tag{2.7}$$

2.4 Byte Pair Encoding for Subwords

Byte Pair Encoding (BPE) is an algorithm for data compression that merges frequent pair of bytes in a single and unused byte. Sennrich *et al* [32] adapt the BPE algorithm to encoding words via subword units to generate a reduced vocabulary. Instead of merge frequent pairs of bytes, Sennrich *et al* technique merges pairs of character sequences to generate subword fragments. This adaptation's objective is to create a vocabulary for neural machine translation that deals with rare words without requiring an extra encoding model.

Given a text corpus, the subword algorithm iterates over all character pairs and replaces each occurrence of the most frequent pair ("A", "B") with its merged representation ("AB") to create a symbol vocabulary. The merge operation creates a new symbol and, consequently, reduces the vocabulary size. The number of merge operations is a constant value that is defined by a hyperparameter of the algorithm. For example, given a text corpus ["low", "lowest", "newer", "wider"], the algorithm detects the following symbols using three merge operations of most frequent pairs: "lo" ("l"+"o"), "er" ("e"+"r"), and "low"("lo"+"w"). With these symbols, we can represent out-of-vocabulary(OOV) words that are composed of such symbols. For instance, we can represent the OOV word "lower" using the symbols "low" and "er".

2.5 Transformer

Transformer is a type of neural network architecture that processes sequences based solely on attention mechanisms instead of using recurrent connections in the network. Vaswani *et al.* [35] developed this architecture to deal with the machine translation task, achieving impressive performance on machine translation tasks. This architecture uses an Encoder-Decoder approach based on other machine translation neural networks such as Sequence to Sequence learning [33]. Approaches that use Transformer variations recently achieve state-of-the-art results on natural language understanding tasks, such as Question Answering [39] and Sentiment Classification [13]. The encoder and decoder of Transformer are composed of blocks that contain two types of neural network layers: a self-attention layer and a feed-forward layer. Image 2.4 shows the Transformer Architecture describing the blocks and its layers.

The self-attention mechanism learns the internal relations between elements of an input sequence, which is a significant part of this architecture. These relations allow Transformer to learn semantic representations that carry information about all elements in the same sequence and learning long term dependencies as well. Given that these neural

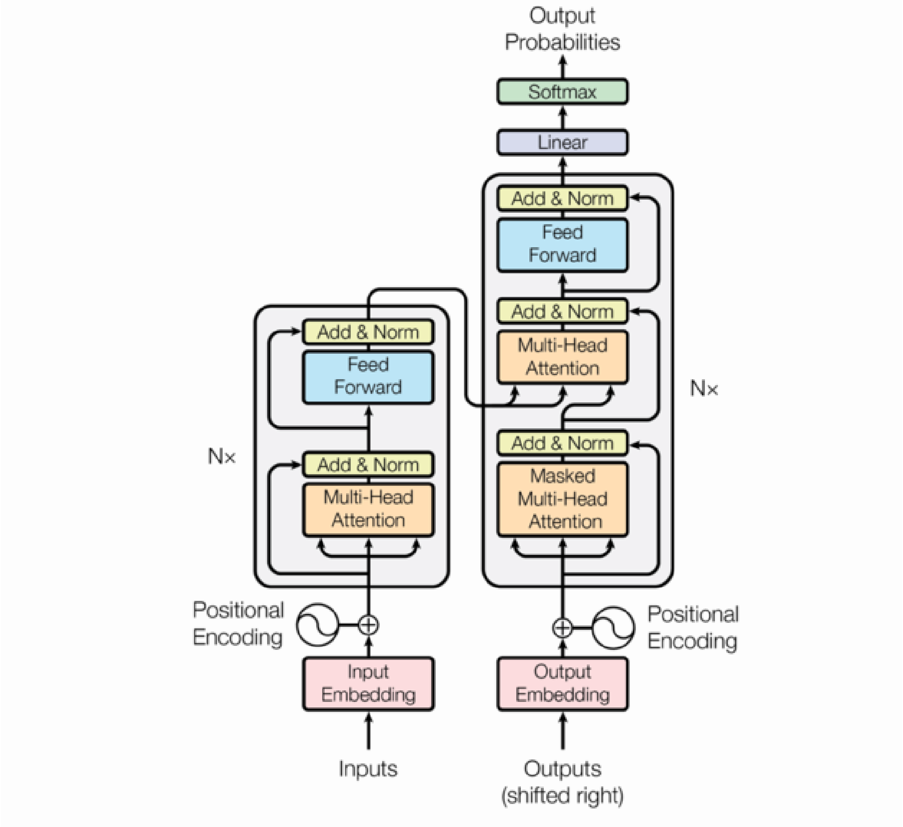

Figure 2.4 – Diagram of Transformer architecture describing the layer composition of blocks. [35]

networks learn term dependencies that rely only on self-attention approach, a recurrent neural network with memory gates is dispensable in this architecture.

On the other hand, since a recurrent mechanism is absent, the input of Transformer architecture does not have information about the order of elements. To deal with this issue, the authors of Transformer architecture introduce a representation that informs the element position in its representation, known as Positional Embedding. The input of Transformer architecture is the sum of input embedding with its Positional Embedding. Vaswani *et al* [35] use a sinusoidal function to represent Positional Embeddings of each position.

A self-attention layer is composed of three matrices: a query matrix, a key matrix, and a value matrix. Each element of a sequence is associated with a query matrix that maps an output given a pair of key/value matrix, which projects the magnitude of relations with other elements. Although encoder and decoder uses self-attention mechanism similarly, the decoder part of Transformers applies a mask to remove later element values to avoid computing the attention score to subsequent positions of a single word. Equation 2.8 shows the function of the self-attention mechanism that results in a representation Z given a query matrix Q, a key matrix K, a value matrix V and a normalization term d_k, which represents the dimensionality of matrix K. Figure 2.5 describes an example detailing each operation of the self-attention mechanism, which produces the attention weights to focus on relevant words and then multiply with its values.

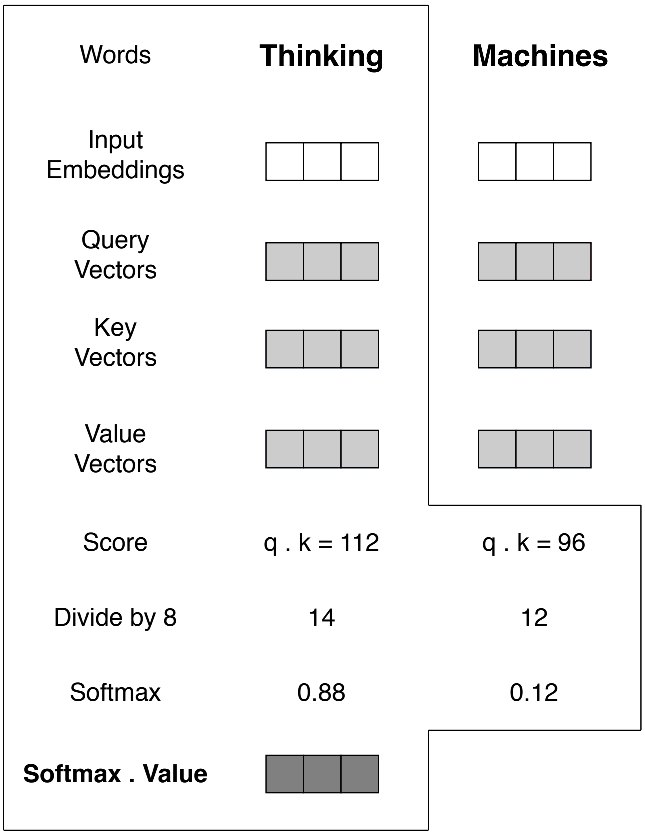

Figure 2.5 – Diagram of operations executed by self-attention mechanism, generating the representation of the word “thinking” given the input sentence “Thinking Machine”. The numbers are hypothetical.

$$Z = softmax(\frac{Q.K}{\sqrt{d_k}}).V \tag{2.8}$$

Besides the computation of relations between elements in a sequence, this architecture introduces the approach of computing multiple representations for each sequence element, known as Multi-Headed attention [35]. The objective of Multi-Headed attention is generating representation subspaces of an element to learn different aspects of an individual element, in which each subspace represents an attention mechanism head. Each head contains a distinct query, key and values matrices randomly initialized and computes the self-attention mechanism as well as explained previously. After computing each head, the neural network concatenates each head output into a single matrix and then transforms into a matrix with the same shape of input size using a fully-connected layer. Figure 2.6 shows how each subspace of multi-head attention computes the output representation.

After computing the self-attention layer, the Transformer block uses a feed-forward neural network to process each hidden state of the sequence separately and identically. The transformer block uses an extra layer applying the Batch Normalization method [18] on the output of each block layer.

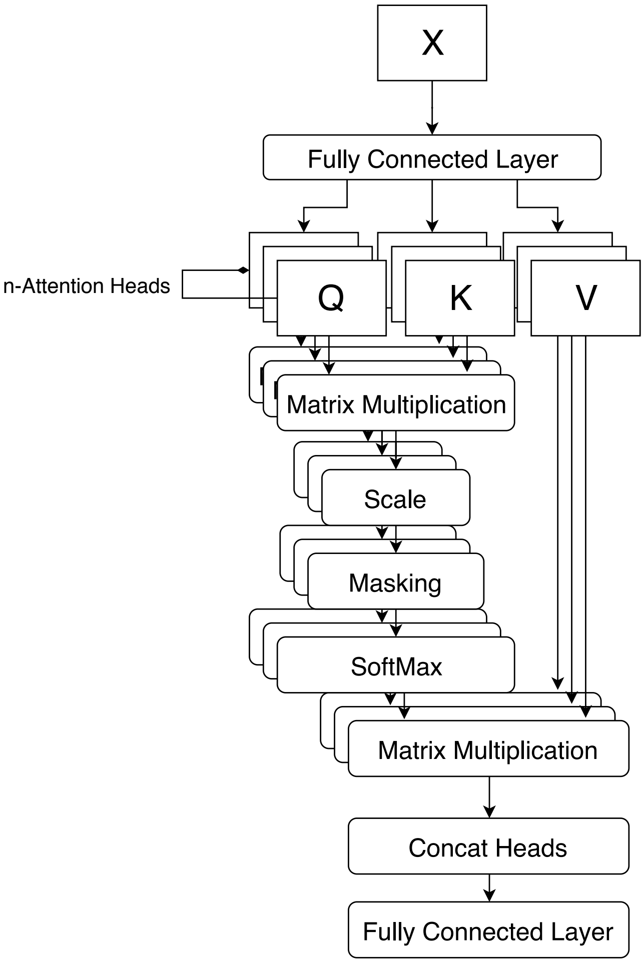

Figure 2.6 – Diagram of self-attention model using Multi-Head approach using 3 heads.

2.6 Transformer-XL

The default Transformer neural network processes a text corpus by splitting into segments and each segment is a separate model input. Given that, a specific segment cannot access the information of other segments of the same corpus, which results in a lack of contextual information. To deal with this issue, the Transformer-XL neural network uses a recurrent neural network to process the segments that compose the corpus as a continuous sequence [10]. With a recurrent mechanism, the Transformer-XL can memorize previous segments improving learned representations with better contextual information.

The Transformer-XL model stores the hidden states of the previous segment using as contextual information to learn term dependencies between segments. For instance, given an application of a default Transformer model for the task of next word prediction in a large corpus, during the training phase, we need to split the text into smaller segments and optimize the model using only the segment words. During the evaluation phase, we need to shift segments to the right to predict the next word of different segments to test considering all words in the corpus regardless of segments separated during training phase.

2.7 XLNet

XLNet is a pretraining method for the Transformer-XL neural network that uses the next-word prediction unsupervised task for language modeling. Using a large unlabeled corpus, this approach aims to improve a model through transfer learning for a supervised downstream task, which usually relies on small labeled datasets. In this section, we describe the main concepts of XLNet. First, we describe the concept of autoregressive language modeling used in XLNet. Second, we describe the unsupervised tasks used in XLNet pretraining known as permutation language modeling. Finally, we describe the new approach of computing self-attention representations introduced by XLNet: Two-Stream Self-Attention.

2.7.1 Autoregressive Language Model

Autoregressive language modeling is a task that aims to predict a token given only its predecessors [39]. Using a large corpus, an autoregressive language model tries to predict each corpus token processing a fixed-size window of its previous word. When processing this task using neural networks, the AL model creates a contextual word representation by learning hidden layer features. Using a large corpus, an AL model tries to predict each corpus token processing a fixed-size window of its previous word. Specifically, given a text sequence $X = [x_1, ..., x_T]$ of length T, autoregressive language modeling maximizes the likelihood using the following equation:

$$\max_{\theta} \log p_\theta = \sum_{t=1}^{T} \log p_\theta(x_t|x_{<t}) = \sum_{t=1}^{T} \log \frac{\exp(h_\theta(x_{1:t-1}).e(x_t))}{\sum_{x'} \exp(h_\theta(x_{1:t-1}).e(x'_t)))} \tag{2.9}$$

In Equation 2.9, $h_\theta(x_{1:t-1})$ corresponds to hidden representations learned by a model given previous the inputs of x_t and $e(x_t)$ corresponds to the embedding representation of x_t. The θ symbol corresponds to the model parameters. Regarding language representation, an autoregressive model brings a limitation about context information: it does not capture the bidirectional context since its inputs consists only of the previous elements (i.e. tokens to the left). To address this problem, XLNet uses all possible permutations of elements in a sequence for the unsupervised task of permutation language modeling [39].

2.7.2 Permutation Language Modeling

Based on autoregressive language modeling, Permutation Language Modeling is a task that predicts the next word considering the preceding context given a permutation of the original sequence. For instance, given a set of tokens $[x_1, x_2, x_3]$, we can use a possible permutation of this sequence $[x_1, x_3, x_2]$ to predict x_2. Since the model predicts x_2 conditioned by x_1 and x_3, using the permutation instead of original sequence allows the model to learn the bidirectional context of x_2. Formally, given a set Z that represents all possibles permutations of sequence X with length T, the PL modeling aims to maximize the likelihood using the following equation:

$$\max_{\theta} \mathbb{E}_{z \sim Z_T} \left[\sum_{t=1}^{T} \log p_\theta(x_{z_t} | x_{z<t}) \right] \tag{2.10}$$

The equation above is similar to the AR modeling equation, however, the difference is that x_{z_t} is conditioned by the previous elements of the permutation sequence $z \in Z$ [39]. The z_t variable denotes the t-th element of the permutation z. In expectation operator, all permutation sequences adjust the same parameter θ during the training execution. Given that, and the x_t symbol receives the information of all elements of sequence x including its successors, capturing then the bidirectional context.

2.7.3 Two-Stream Self-Attention

To deal with the Permutation Language Modelling task using the Transformer architecture, the XLNet uses a modified version of the self-attention mechanism. To predict the x_{z_t} token, the neural network should not see its content but only its position to avoid to become a trivial task. For example, given the sentence “permission does not imply obligation” for the next-word prediction task, to predict the word “obligation” we should only consider its predecessor tokens “permission does not imply”. On the other hand, to use other elements in $x_{z<t}$ to predict x_{z_t}, we need its representations already formulated by the neural network. The XLNet introduces the Two-Stream attention that formulates two types of hidden representation: the content representation and query representation.

The content representation h_{z_t} uses the standard self-attention mechanism as well as in the original Transformer implementation, which can access all tokens $x_{z<t}$ including x_{z_t}. Alternatively, the query representation g_{z_t} do not access the x_{z_t} information but only its context $x_{z<t}$ in the permutation z. Equation 2.11 shows the details of how to generate the g_{z_t} and h_{z_t} using the self-attention mechanism.

$$\begin{aligned} h_{z_t}^{m} &= Attention(Q = h_{z_t}^{m-1}, KV = h_{h_{z \leq t}}^{m-1}) \\ g_{z_t}^{m} &= Attention(Q = g_{z_t}^{m-1}, KV = h_{h_{z<t}}^{m-1}) \end{aligned} \tag{2.11}$$

In the m-th self-attention layer, the self-attention query vector Q uses the value of previous layer (m-1) in both representations. The single difference is that the self-attention Key and Value vectors (KV) of g_{z_t} does not include the $h_{z_t}^{m-1}$, which means that it only considers the previous token representations $h_{h_{z<t}}$. Both representations are related considering

that g_{z_t} uses the $h^{m-1}_{h_{z<t}}$ computed by content representation in the previous layer. Regarding the initialization of such representations in the first layer (m=1), the $h^1_{z_t}$ receives the embedding vector of x_{z_t} and $g^1_{z_t}$ receives a randomly initialized vector, which is updated in the backpropagation execution. Since the XLNet uses the query representation to deal with issues of the PL modeling task, it is not necessary during the finetuning process.

3. APPROACH

In this chapter, we describe our approach to detect potential problems in a contract, given the relation of its normative sentences. First, we develop our multi-model approach to recognize not only whether a pair of sentences contains a conflict but also its inferential relation. Second, we specify our neural network to deal with the natural language inference task and the norm conflicting detection task. Third, we report the dataset that we use for training our models. Finally, we describe the implementation details of our neural network and training details as well.

3.1 Multi-Model for Potential Contractual Problems

Our multi-model approach relies on a model to predict the inferential relation between a pair of norms and a model to detect potential conflicts in the same pair. Although the results of both tasks are different, we use the same model specification since both tasks deal with sentence pairs written in natural language. Our multi-model receives a pair of normative sentences and processes the norms by two different models simultaneously and apart. Given that there is no dependency between models, the two models do not share parameters and, hence, we train both models separately.

Given a normative sentence pair (n_1,n_2), the NLI Model predicts the inferential relation between both sentences resulting in the following classes: contradiction, entailment and neutral. Since the existence of conflict between clauses is independent of the order of the clauses (i.e. a conflict is a problem regardless of which clause comes first), we process pairs (n_1,n_2) and (n_2,n_1) to consider both sentences as the premise in the NLI Model.

The Norm Conflict Classification (NCC) model predicts whether a potential conflict exists between n_2 and n_1 or not. When conflict exists, the model predicts the conflict type following the typology introduced by Aires *et al* [1]. Figure 3.1 shows a diagram that represents our multi-model approach, illustrating how it processes a pair of norms and how we represent the likelihood of the norm pair belonging to a particular class.

3.2 Neural Network Specification

We implement a neural network model for each task as a multi-label classifier. Both models use the same specification and the same neural network architecture details except the number of predicted classes in the output layer. Since both tasks involve natural

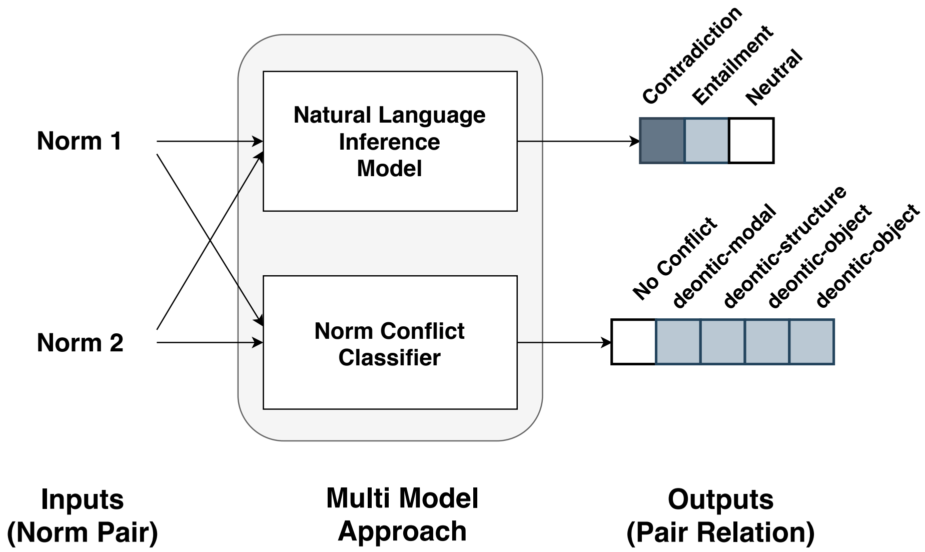

Figure 3.1 – Diagram illustrating our multi-model approach processing a norm pair, which results in two logit vector: one that represents the likelihood of the norm pair belonging to a particular conflict type conflict type (or nonconflicting) and another to represent its inferential relation (NLI classes).

language classification, we leverage a modified XLNet [39] model to learn sentence-pair representations.

As well as original Transformer architecture [35], we use subword units [32] to represent the sentence words using a pretrained token vocabulary made available by Yang *et al* [39]. The model input consists of the concatenation of both sentences n_1 and n_2 including special tokens *SEP* and *CLS*. We define this input structure following the input structure of XLNet pretraining task to avoid creating a discrepancy between the pretraining method and our tasks. The *SEP* token separates the sentences and *CLS* is token used by the XLNet to encode the whole input sequence.

Since XLNet input consists of text fragments that might contain one or more sentences, we separate each sentence in segments including a segment identifier for n_1 and n_2 to differentiate whether two tokens are within the same sentence [39]. Regarding the special tokens (*CLS*, *SEP*), we include a specific segment index to differentiate from sentence content considering that these tokens should not bias the segment representation. Equation 3.1 shows how we concatenate the (n_1,n_2) tokens and Equation 3.2 shows the segment definition, where x_{n_1,n_2} is the token sequence and s_{n_1,n_2} is the sequence of segment indexes.

$$x_{n_1,n_2} = [sep, n_1, sep, n_2, sep, cls] \tag{3.1}$$

$$s_{n_1,n_2} = [s_{sep}, s_{n_1}, s_{sep}, s_{n_2}, s_{sep}, s_{cls}] \tag{3.2}$$

To predict the output class, we include a feed-forward neural network on top of our classifier model using the representation produced by XLNet as the input of this network.

The XLNet neural network uses the self-attention mechanism [35] to represent all input tokens, which means that all representations carry a part of other token information measured by an attention score. We use only the *CLS* representation h_{cls} to predict the output class since it contains the information of all tokens in sentence pair. The feed-forward neural network computes a *tanh* activation function to generate a logit vector representing each class probability. Equation 3.3 shows how we predict the output class, where h denotes a vector that contains all token representations of (n_1,n_2) produced by XLNet, $tanh(h_{cls}W^T)$ denotes the *tanh* activation function of feed forward network and $\hat{y}$ is the predicted class given the highest logit value contained in logit vector l. Figure 3.2 shows how the XLNet neural network predicts the output class $\hat{y}$ given the tokens $x^{(1)}$ of n_1 and the tokens $x^{(2)}$ of n_2. Importantly, Figure 3.2 shows the neural network details represented by boxes in the diagram of Figure 3.1.

$$\begin{aligned} h &= XLNet(n_1, n_2) \\ l &= tanh(h_{cls}W) \\ \hat{y} &= \arg\max_c l_c \end{aligned} \tag{3.3}$$

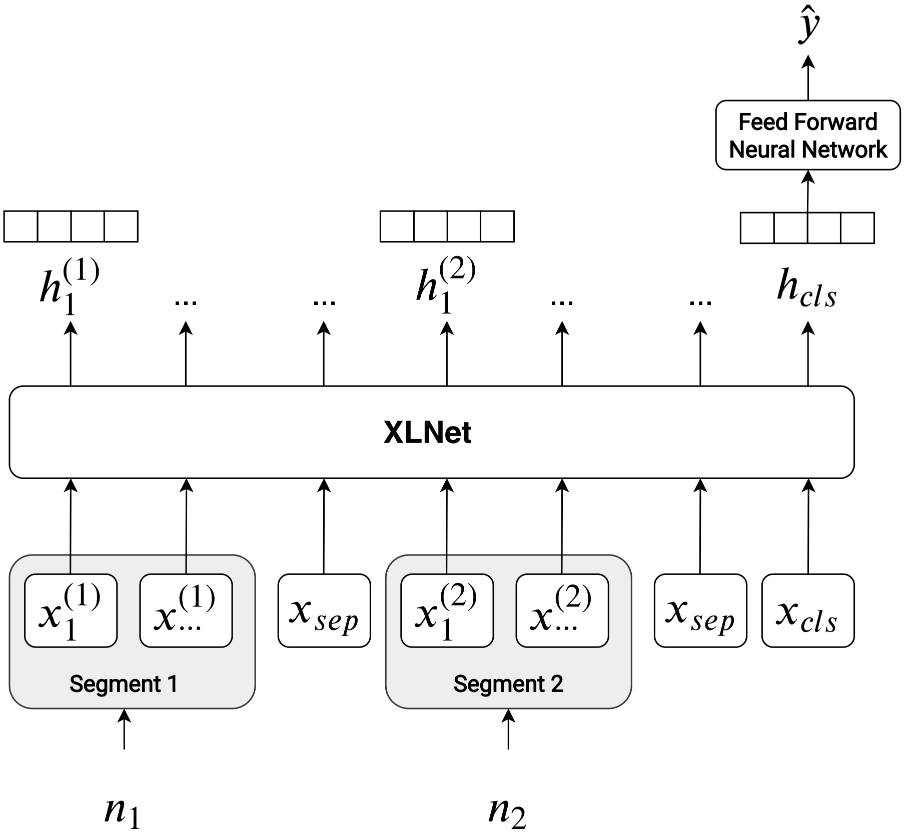

Figure 3.2 – Multi-label classifier that uses XLNet to generate tokens representation followed by a Feed Forward Neural Network to predict the output class.

3.3 Datasets

In this section, we report the datasets that we use for training our models. First, we describe the MultiNLI (MNLI) dataset composed by sentence pairs annotated with a Natural Language Inference label. Second, we describe the Norm Dataset for norm conflict classification that we use for NCC-Model training.

3.3.1 MultiNLI

The Multi-Genre Natural Language Inference (MultiNLI) corpus is a large dataset that contains 433k sentence pairs annotated with NLI classes (*contradiction, entailment, neutral*) [37]. This dataset is based on Stanford Natural Language Inference (SNLI) corpus [6] and the data has the same format, and was collected in a similar way. However, MultiNLI contains more diverse sentences including text and speech from ten different genres compared to SNLI, since it contains only image caption descriptions. The genres of MultiNLI are the following categories: face-to-face conversations [1] (FACE-TO-FACE), government content (GOVERNMENT), Letters (LETTERS), report of the terrorist attack in 9/11[2] (9/11), sentence from non-fiction work of Oxford University Press (OUP), Slate Magazine contents (SLATE), telephone conversation[3] (TELEPHONE), travel guide content (TRAVEL), verbatim content[4] (VERBATIM), and fiction content (FICTION).

This MultiNLI corpus provides the following sets: train set, dev matched/mismatched sets and test matched/mismatched set. The mismatched version of the dev set contains sentences from the same sources of training set while the mismatched version sentences of training and dev set differ substantially. The test set has the same division of dev set but with the unlabeled sentence pairs. To evaluate a model using the test set, the predicted values must be submitted in a Kaggle Competition[5][6] since the test labels are hidden, which each test prediction will be evaluated by their platform.

[1] https://newsouthvoices.uncc.edu/

[2] https://9-11commission.gov/

[3] https://catalog.ldc.upenn.edu/LDC97S62

[4] http://www.verbatimmag.com/

[5] https://www.kaggle.com/c/multinli-matched-open-evaluation

[6] https://www.kaggle.com/c/multinli-mismatched-open-evaluation

Table 3.1 – Example of sentence pairs contained in MultiNLI dataset with their respective NLI labels.

Premise	**Hypothesis**	**Label**
The Old One always comforted Ca'daan, except today.	Ca'daan knew the Old One very well.	Neutral
At the other end of Pennsylvania Avenue, people began to line up for a White House tour.	People formed a line at the end of Pennsylvania Avenue.	Entailment
A woman selling bamboo sticks talking to two men on a loading dock.	A woman is not taking money for any of her sticks.	Contradiction

3.3.2 Norm Dataset

The Norm Conflict Dataset is a corpus that contains 11557 sentences that represent clauses given a variety of contracts [1]. Aires *et al* develop this dataset in a semi-automated way using volunteers to create a conflicting second norm given the original. The dataset contains 111,329 non-conflicting norm pairs and 238 conflicting norms annotated with their respective conflicting types. The conflicting types covered by this dataset follows the typology introduced in Section (*deontic-modality, deontic-meaning, deontic-object, object-conditional*).

3.4 Implementation and Training Details

In both model instances, we implement a compact version of XLNet, named XLNet-Base, instead of using the large version (XLNet-Large) for multi-label classifier model. Although the XLNet-Large results in better performance compared to the XLNet-Base, we use the smaller version in order to speed up training by having fewer trainable parameters. Importantly, the fewer parameters help ups more easily refine the network with a relatively smaller training dataset in the Norms Dataset. We implement the two models using XLNet-Base pretrained weights made available by Yang *et al* [39] and use auxiliary code provided by HuggingFace [38]. Table 3.2 shows the specification of both XLNet models describing the number of transformer blocks, the number of heads used in multi-head attention, the hidden state size and the total number of trainable parameters.

Table 3.2 – Specification of each XLNet model.

Model	**Blocks**	**Heads**	**Hidden**	**Parameters**
XLNet-Base	12	12	768	110M
XLNet-Large	24	16	1024	340M

We train both models using the Adam algorithm [19] to optimize the neural network weights and we measure the model error using the Negative Log-Likelihood (NLL) loss function in output probabilities. Since we deal with a multi-class classification task, our loss function accumulates the log loss values of each class prediction. To avoid the exploding gradient problem [28], we clipped the gradient norm within 1. Given an output label y_c for class c and a premise-hypothesis pair (p, h), the goal is to minimize the function shown in Equation 3.4.

$$NLL = -\sum_{c} y_c . \log P(c|p, h) \tag{3.4}$$

3.4.1 NLI-Model Training

In the NLI-Model, our training procedure is similar to the procedure by Yang *el al* [39] in single-task XLNET training for the NLI task. However, since their focus is not on hyperparameter finetuning for MultiNLI, we finetune for the NLI task using specific hyperparameter values. Given a specified number of steps during the training process, we validate our model in MultiNLI matched and mismatched dev sets to select the best values given the model performance on these datasets. To use small train batches, we accumulate the back-propagated values before update the NLI-Model weights. The gradient accumulation allows us to train the model in a GPU with less memory since we use a batch size 16 times smaller than that of Yang *et al.* As we accumulate gradient given a specific number of steps, we scale up the number of training steps to increase the number of updates in model weights. Table 3.3 shows the hyperparameter values that we use in NLI-Model comparing with values used by Yang *et al* in NLI task finetuning.

Table 3.3 – Hyperparameters values for NLI-Model train.

	Ours	**Yang *et al* [39]**
Learning rate	3e-5	3e-5
Batch size	8	128
Gradient acc. steps	16	-
Adam epsilon	1e-8	1e-6
Input sequence size	170	128
Training steps	150K	10K

We monitor the training of the NLI-Model by creating checkpoints in intervals of 2000 steps to extract its loss value and accuracy in the validation dataset. Although we limit the training execution to 50000 steps, we apply the early stop technique to suspend the execution when the loss value in the validation set stops to decrease. Figure 3.3 shows the loss and the accuracy progress of the NLI-Model throughout the steps on the MNLI validation set.

We apply a learning rate scheduler to decay its value in each weight update step through a linear function. Equation 3.5 shows how we compute the linear rate decay where lr_{t+1} and lr_t are,respectively, the next and actual learning rate value, $step_t$ is the actual train step, and $step_{total}$ is the number of weight update steps.

$$lr_{t+1} = lr_t \cdot \frac{step_t - step_{total}}{step_{total}} \tag{3.5}$$

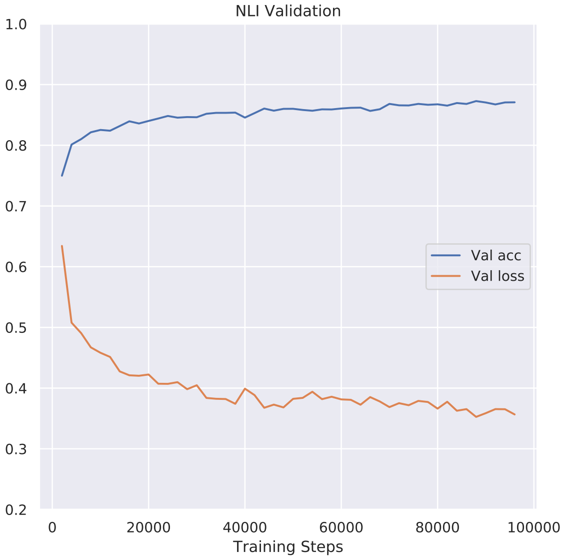

Figure 3.3 – Progress of the loss and the accuracy of the NLI-Model in MNLI validation set throughout the training steps using the hyperparameter values of Table 3.3

3.4.2 Norm Conflict Classification Model Training

To train the Norm Conflict Classification (NCC-Model), we select the single fold using as a criterion the balance between classes to train the NCC-Model to produce a fair comparison. Aires split this dataset into ten folds to fit their models using the k-fold cross-validation process. Table 3.4 shows the data statistics of our selected fold, which contains 369 samples in the train set and 41 for the validation set, reporting the number of examples per class.

We train NCC-Model using two procedures: training reusing NLI-Model weights NCC-Model$_{\text{nli-ft}}$, and training from scratch using directly pretrained XLNet weights. In the first experiment, we initialize the NCC-Model reutilizing the NLI-Model weights to explore the relation between two tasks. Our goal is to determine the existence of a link between logical relations such as contradiction, and conflict typology as defined by Aires *et al* [1]. In contrast

Table 3.4 – Samples per class contained in the train and val sets that we use to train NCC-Model.

Class	**Train**	**Validation**
nonconflicting	184	20
deontic-modal	80	8
deontic-structure	47	7
deontic-object	31	4
object-conditional	27	2

to NLI, which contains a significant volume of labeled datasets, finding an openly available norm dataset is difficult. Thus, we investigate whether transferring weights from NLI-model to the NCC-Model has the potential to improve it. Alternatively, considering that might be an excess of weight update, in the second experiment we use the XLNet-Base pretrained weights to train directly on the selected norm dataset fold.

Note that in the first experiment we use a smaller learning rate than used in NLI-Model to prevent large magnitude changes to the pretrained NLI weights. Applying the gradient accumulation in both experiments improves the results even using a dataset with few examples. Table 3.5 shows the hyperparameter values of the NCC-Model training process.

Table 3.5 – Hyperparameters values used in NCC-Model training.

Learning rate	2e-6 with NLI weights 1e-5 w/o NLI weights
Batch size	4
Gradient Accumulation Steps	3
Adam epsilon	1e-8
Input Sequence Size	250
Training Steps	5000

In the NCC-Model training process, we use checkpoints per epoch instead of per steps since the volume of the MNLI dataset is much greater than the Norm Dataset. Thus, we use the early stop technique monitoring the result obtained in the epoch's last step. We use weights of the best checkpoint regarding its accuracy and loss in the validation set of the selected fold as the final model.

Figures 3.4 and 3.5 show the training progress of NCC-Model and NCC-Model$_{\text{nli-ft}}$ respectively throughout the epochs. We note that the NCC-Model$_{\text{nli-ft}}$ training procedure is slower when compared to the NCC-Model given that it needs more training epochs to early stop. On the other hand, the progress of the NCC-Model$_{\text{nli-ft}}$ validation loss is stable when compared to NCC-Model, which indicates that the NCC-Model is more prone to overfitting.

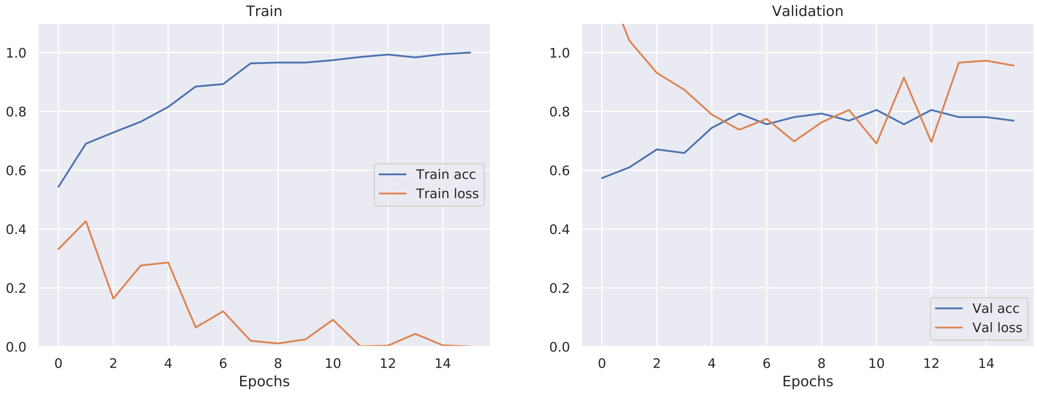

Figure 3.4 – Progress of the loss and the accuracy of the NCC-Model throughout the training steps using the hyperparameter values of Table 3.5

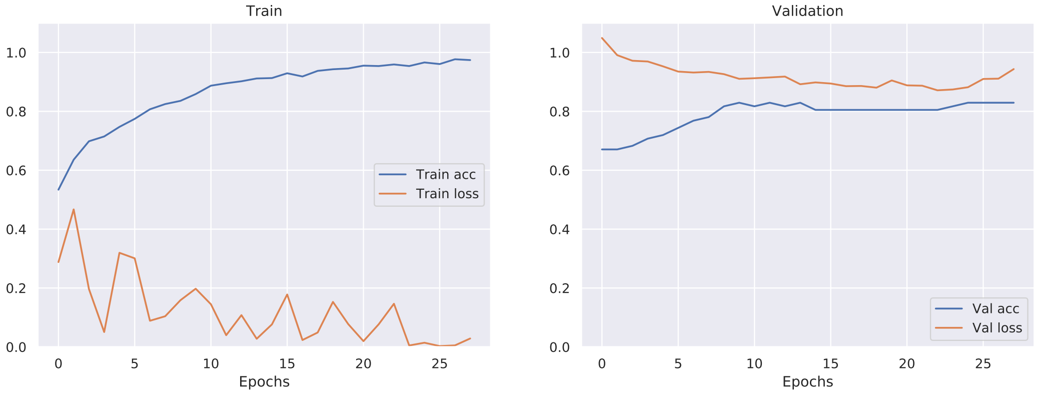

Figure 3.5 – Progress of the loss and the accuracy of the NCC-Model$_{\text{nli-ft}}$ throughout the training steps using the hyperparameter values of Table 3.5

4. RESULTS

In this chapter, we report the results of our models using the MultiNLI dataset for the NLI-Model and Norm Dataset for the norm conflict classification models. First, we report a qualitative comparison between related approaches describing a quantitative analysis. Finally, we report the results of specific norm pairs to compare with related approaches describing a qualitative analysis.

4.1 Quantitative Analysis

4.1.1 NLI-Model

In this section, we compare the results of our trained NLI-Model with similar models, such as the XLNet original work and BERT [13]. BERT is the previous state-of-the-art model and XLNet-Base introduced by Yang *et al* is the base of our NLI-Model. We choose XLNet-Base and BERT-Base models in our comparison considering their similar architecture since these two models include 12 transformer blocks in its architecture. Table 4.1 shows a comparison of our approach with similar models using the MultiNLI dev matched (m) and mismatched (mm) datasets.

Table 4.1 – Comparison of our trained NLI-Model and similar models using the MultiNLI dev datasets.

Model	**Acc (m/mm)**
BERT-Base [13]	84.34/84.65
XLNet-Base [39]	85.84/85.43
XLNet-Base (Our NLI-Model)	87.06/86.30

Our NLI-Model obtains slightly better results compared to BERT-base and even with the original XLNet in both datasets. Both models have the same neural network architecture as well as the number of trainable parameters. However, the main difference between our pretrained XLNet-Base and Yang *et al* [39] is the training procedure, since we do hyperparameter tuning for NLI task specifically.

4.1.2 NCC-Model

We now compare the results of our trained NCC-Model with related work models as well as both training procedures. We select the works of Aires and Meneguzzi [2], which uses a Convolutional Neural Network (CNN), and Aires *et al* [1], which uses learned semantic representations as inputs of an off-the-shelf Support Vector Machine model. We compare the results using all norm dataset labels, which include non-conflict pairs. The comparison with Aires *et al* uses the the best performing approach (the concatenation of norm embeddings) for all five labels. We report, for all approaches, accuracy (A), precision (P), recall (R), and F-measure (F). In this comparison, we include the following two NCC-Models with different training procedures: NCC-Model$_{\text{nli-ft}}$, which we train reusing NLI-Model weights and NCC-Model, which we train using directly Norm Dataset. As shown in Table 4.2, our approach surpasses the state-of-the-art results in all metrics using both training procedures by a considerable margin.

Table 4.2 – Comparison between our approaches with current and previous state-of-the-art approaches considering all classes (4 types of conflicts and non-conflicts norm pair) in test dataset. We compare accuracy (A), precision (P), recall (R) and F-measure (F) values among the approaches.

Approach	**A**	**P**	**R**	**F**
Aires and Meneguzzi[2]	0.63	0.59	0.64	0.61
Aires *et al*[1]	0.70	0.71	0.64	0.66
NCC-Model	0.91	0.95	0.91	0.92
NCC-Model$_{\text{nli-ft}}$	**0.93**	**0.96**	**0.93**	**0.95**

Although using NLI-Model weights as the starting point of the NCC-Model obtains better results in the test set, we do not have enough evidence that that the NLI pretraining is indeed superior to the NCC-model alone, since the improvement relies on a single example. Since this problem is one of binary classification (conflicting and non-conflicting norm pairs) and the improvement relies on just two examples, we prefer to be cautious about claims of superior accuracy. Analyzing the confusion matrices shown in Figures 4.1 and 4.2, we note that NCC-Model misclassified one conflicting norm pair as non-conflicting and one conflicting norm pair as non-conflicting when compared to NCC-Model$_{\text{nli-ft}}$. On the other hand, NCC-Model$_{\text{nli-ft}}$ misclassified examples belonging to similar conflict (*deontic-modality* and *deontic-structure*) since both types rely on the modal verb. We argue that the NCC-Model$_{\text{nli-ft}}$ misclassification is subtler since the error concerns on similar conflict types and, on the other hand, the NCC-Model could not recognize a conflict regardless of its type. This error comparison provides an additional piece of evidence of the model improvement using the NLI pretrained weights.

4.1.3 Statistical Test

To test whether the use of pretrained weights improves the performance of the NCC-Model, we use the McNemar's Statistical Test[24] to compare the error proportion of

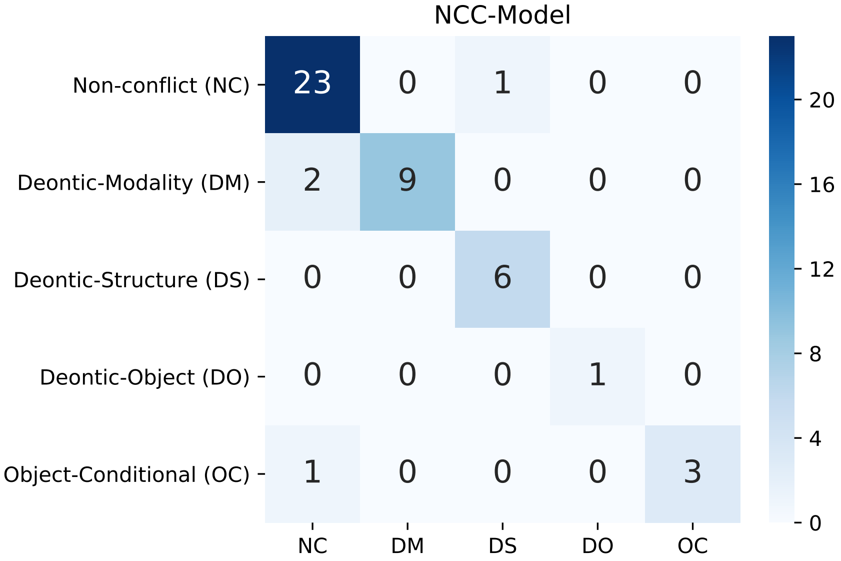

Figure 4.1 – Confusion matrix of NCC-Model in the test set.

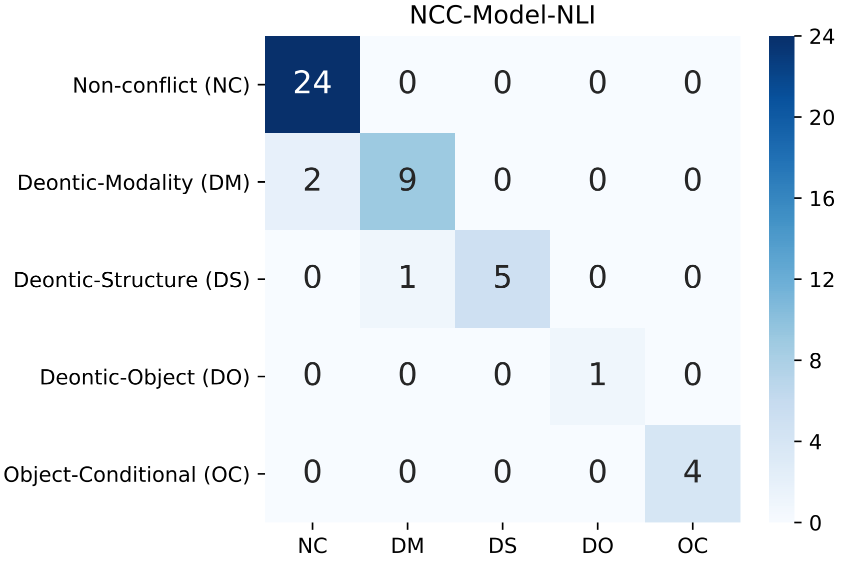

Figure 4.2 – Confusion matrix of NCC-Model$_{\text{nli-ft}}$ in the test set.

the two approaches (NCC-Model and NCC-Model$_{\text{nli-ft}}$). In this statistical test, we apply two different methods in the same dataset and collect their results to test its null hypothesis, which indicates whether the two methods have the same proportion of errors [14]. The rejection of the null hypothesis of the McNemar's test occurs when the two approaches result in different proportion of error, which shows that the two models have a significant difference.

To apply the McNemar's test, we create a contingency table that contains the number of correct/incorrect instances of each method. Table 4.3 shows the contingency table of two approaches using the test set of Norm Dataset.

Table 4.3 – Contingency Table with number of instances that were correct/incorrect classified by the norm classification model trained reusing NLI-Model weights (NCC-Model$_{\text{nli-ft}}$) and another norm classification model without NLI-Model weights (NCC-Model).

	Correct (NCC-Model$_{\text{nli-ft}}$)	**Incorrect (NCC-Model$_{\text{nli-ft}}$)**
Correct (NCC-Model)	41	1
Incorrect (NCC-Model)	2	1

As both models result in a high accuracy on the test set, the contingency table shows a contrast between the number of correctly classified pairs with other numbers. Given such disparity, the test fails to reject the null hypothesis showing that both approaches have a similar proportion of errors on the test set, since the probability value (p-value) resulted from our contingency table is 1. Such results corroborate our findings that using a pretrained NLI model does not certainly improve the accuracy of the Norm Classification task, as mentioned in the previous section.

4.2 Qualitative Analysis

In this section, we analyze the results of typical examples extracted from the test set to compare our approach against the related work. Aires *et al* work [1] the following norm pair to compare its approach performance against Aires and Meneguzzi work [2]:

- Invoice cost shall not be adjusted for, and Customer shall not be entitled to, promotional allowances, cash discounts, prompt pay discounts, growth programs or any other supplier incentives received by USF.
- If the cost of the invoice is adjusted, Customer will not be entitled to promotional discounts, cash discounts, growth programs or any other supplier incentives received by USF.

Aires and Meneguzzi's approach misclassified as non-conflicting with 64% confidence while Aires *et al*'s approach correctly classified as object-conditional conflict with 40% confidence. Both of our models correctly classify this norm pair as well with 97% and 98% of confidence respectively. Such results show that our approaches not only infer the correct conflict type for this norm pair but also do it with increased confidence level when compared to related work.

For the next example, we compare the results of both approaches to investigate the effects of using NLI weights for the NCC task. The difference between both approaches relies solely on the following norm pair:

- ACTII shall be responsible for the design, construction, equipment, validation and maintenance of the ACTII Facilities, including the Janssen Equipment.
- Except for the Janssen Equipment, ACTII must be in charge of the all production processes involving ACTII Facilities.

The NCC-Model$_{\text{nli-ft}}$ correctly classified this norm pair as *object-conditional*, spreading the confidence levels across conflict types assigning 35% confidence to *object-conditional*, 30% to *deontic-structure*, and 27% to *deontic-object*. By contrast, the NCC-Model misclassified as non-conflicting pair with 84% confidence, which is a notable error since this example is a conflicting norm pair. Our intuition is that the NCC-Model$_{\text{nli-ft}}$ could capture better the modal verb information since it can indicate an entailment or even a contradiction. We discuss further this issue in Section 5.3, which we report the attention weights of each word in norms.

5. EXPERIMENTAL ANALYSIS

In this chapter, we describe how both models work combined to help find potential contractual problems. First, we explore the entailment relation of conflicting pairs that relies on deontic meaning. Second, we explore the transitive relations to find conflicts that may arise within three norms. Third, we explore the self-attention weights that highlight terms with a strong correlation with each conflict or inferential relation. Finally, we describe the limitations of our models as a result of the relatively small available dataset.

5.1 NLI Predictions on Norm Dataset

In this section, we describe the predicted NLI class for each norm pair contained in the selected fold of norm dataset. Given a norm pair, we show the NLI predicted class by the NLI-Model and compare its inferential relation with its conflict type. Instead of using the entire dataset, which is unbalanced regarding the number of pairs for each conflict type, we select a single fold created using Aires *et al* sampling strategy [1]. We show the results obtained from both models on the norms dataset fold reporting each NLI class confidence grouped by each conflict type.

5.1.1 NLI in Non-Conflicting Pairs

Since nonconflicting pairs might not refer to the same norm parties or actions, the NLI-Model predicts the neutral relation for most norm pairs as expected. The NLI-Model predicts the neutral relation in 80% of nonconflicting pairs processed, and its confidence surpasses all other NLI classes by a large margin. We compute each confidence percentage using the Softmax function over the logits predicted by the NLI-Model. We use the average confidence of the NLI-Model to report how much an NLI class may be related to a non-conflicting norm pair considering all instances contained in Norm Dataset fold. Table 5.1 shows the number of nonconflicting norm pairs predicted for each NLI class including the average of each NLI class confidence.

5.1.2 NLI in Conflicts Related to Modal Verbs

In this section, we cover conflicts related to deontic meaning divergence through modal verbs such as deontic-modality and deontic-meaning. These two conflict types occur

	Avg conf. (n1, n2)	**Predictions (n1, n2)**	**Avg conf. (n2, n1)**	**Predictions (n1, n2)**
Contradiction	21,91 %	27	23,60%	31
Entailment	8,01%	9	6,89%	5
Neutral	70,07%	148	69,49%	148

Table 5.1 – The average confidence of the NLI-Model considering all non-conflicting pairs with the respective number of norm pairs predicted for each NLI Class. We include results considering both directions in the NLI-Model regarding the premise/hypothesis roles (n1,n2 and n2,n1).

when a norm pair express a different deontic modality. In such conflict types, we report a significant number of norm pairs that the NLI-Model classified as a contradiction. We observe that in norm pairs where one sentence conveys an obligation or permission and the other conveys a prohibition, the NLI-Model could identify a contradiction relation. This shows that our model tends to indicate a contradiction in occurrences of negation of a modal verb, which often illustrates a prohibition norm. Tables 5.2 and 5.3 show the number of norm pairs annotated with deontic-modality and deontic-structure with its respective NLI class.

	Avg conf. (n1, n2)	**Predictions (n1, n2)**	**Avg conf. (n2, n1)**	**Predictions (n2, n1)**
Contradiction	60,52%	57	61,57%	58
Entailment	32,12%	28	25,27%	27
Neutral	0,07%	6	11,14%	6

Table 5.2 – The average confidence of the NLI-Model considering all *deontic-modality* conflicting norm pairs in selected fold with the respective number of norm pairs predicted for each NLI Class.

	Avg conf. (n1, n2)	**Predictions (n1, n2)**	**Avg conf. (n2, n1)**	**Predictions (n2, n1)**
Contradiction	73,77%	41	64,53%	36
Entailment	15,68%	8	7,6%	13
Neutral	10,54%	4	27,83%	4

Table 5.3 – The average confidence of the NLI-Model considering all *deontic-structure* conflicting norm pairs in selected fold with the respective number of norm pairs predicted for each NLI Class.

The NLI-Model recognizes an entailment relation in norm pairs that the premise informs an obligation and the hypothesis informs permission. On the other hand, the NLI-Model shows in some examples that the opposite is not true when the obligation comes from the hypothesis norm resulting in neutral relation in this case. This relation reflects the differences between modal verb intensity recognized by our NLI Model given a norm pair. This case illustrates that even an entailment relation can represent conflict in a contract since the pair contains different deontic meanings for the same parties and the same action.

We show examples of norm pairs annotated with deontic-modality and deontic-structure that describe this relation in Table 5.4. The two first rows show examples of relations between a permissible and an obligatory norm, where the entailment and neutral relation reflects the inference relation. The last two rows show examples of contradiction cases where a pair represents permission and prohibition of the same action for the same party. The norm pair shows that our model could detect a contradiction in a prohibition with modal verb not being negated directly. However, we note that in the last row, in which norms contain a different structure, the NLI-Model decreases the confidence of classifying contradiction correctly when norm (b) is the premise and (a) is the hypothesis.

Norm Pair	**a,b**	**b,a**	**Conflict**
(a) CBSI will retain the originals in its archives. (b) CBSI may retain the originals in its archives.	E (49,39%)	N (64,11%)	*DM*
(a) All prices quoted are exclusive of federal state and local excise sales use and similar taxes and any duties and VA Research shall be responsible for all such items. (b) VA Research may be liable for any of these items if anywhere they would be federal state and local taxes sales tax use and similar and any rights.	E (84,56%)	N (51,28%)	*DS*
(a) The Specifications may only be amended by the NCR design release process. (b) The Specifications shall not be amended by the NCR design release process.	C (98,74%)	C (96,46%)	*DM*
(a) No material changes may be made to the Headcount Plan without the prior approval of the Global Supply Team. (b) Parties may perform changes on materials whenever they want.	C (95,41%)	C (49,37%)	*DS*

Table 5.4 – Norm pairs (a,b) annotated with conflict types deontic-modality (DM) and deontic-structure (DS) with their respective NLI class predicted by the NLI-Model. The NLI classes entailment (E), neutral (N) and contradiction (C) are related to their confidence and we provide NLI-Model results for both sides (a,b) and (b,a).

5.1.3 NLI in Conflicts Related to Norm Object

In cases where the normative conflict emerges from norm objects (deontic-object and object-conditional), we note there is a low correlation between NLI classes and the detected conflicts. These conflict types result in more diverse cases regarding the inferential relation between norms. We observe that our NLI-Model predictions do not follow the strict definition of contradiction in NLI, such that the predicted class does not reflect necessarily the norm pair inference relation. In some cases where the norm conflict occurs when the norm object diverges, the NLI-Model predicts a contradiction relation. Although the NLI-

Model can detect a contradiction when norm objects are contradictory, in cases where the objects of hypothesis norm do not necessarily imply in the negation of premise object, the NLI-Model tends to predict the norm pair as a contradiction relation. The norm pairs annotated with deontic-object expose some limitations of our NLI model regarding our training dataset and time relations described in Section 5.4.

Table 5.5 – Norm pairs annotated as deontic-object conflict in norm dataset fold with the respectives NLI-Model predictions. We include the confidence level of NLI-Model for each predicted NLI class: Contradiction (C), Entailment (E) and Neutral (N).

Norm Pair	**a,b**	**b,a**
(a) Autotote shall make available to Sisal one 1 working prototype of the Terminal by May 1 1998. (b) Autotote shall make available to Sisal one 1 working prototype of the Terminal by June 12 1998.	C (87,65%)	C (85,99%)
(a) NCR shall pay the freight carrier directly. (b) NCR must pay the freight carrier making a back deposit.	C (74,43%)	C (91,93%)
(a) MOPAC and Biopure will cooperate in causing an orderly connection of the Separation Facility with the System. (b) MOPAC shall make an orderly connection of the Separation Facility with the System.	E (95,66%)	N (99,14%)
(a) Hershey will cooperate in no shipping procedures. (b) Hershey will cooperate in all shipping procedures.	C (99,93%)	C (99,82%)

Table 5.5 shows NLI-Model results in conflicting norm pairs annotated as deontic-object. The first and second rows illustrate norm pairs that our NLI-Model predicts a contradiction with high confidence when norm actions are different but not necessarily represents an opposition. While the first row represents a norm pair that contains a simple time-related divergence, the second row illustrates a conflict that emerges from divergence in specification details about the norm actions. In such cases, the difference does not necessarily indicate a contradictory relation between norm actions and, hence, the NLI-Model prediction is uncertainty regarding the norm action meaning. The third row illustrates a norm pair with a conflict that arises from the definition of which party will perform the same norm action. Both norms express the same action but the norm (a) includes the subject of the norm (b), which leads the NLI-Model to predict an entailment relation considering the norm pair (a, b). On the other hand, the opposite pair (b, a) is not a valid entailment association since (b) does not include all subjects of (a), which results in a neutral relation. In contrast with the two first examples, the fourth norm pair is an instance where the norm (a) negates the action of the norm (b), leading the NLI-Model to classify a contradiction relation correctly.

In norm pairs that the conflict arises from its conditional actions (object-conditional), we observe that the NLI-Model predicts contradiction when a norm condition may lead to the opposite action of another one when satisfied. Alternatively, the NLI-Model predicts an entailment relation where the condition of one norm may lead to the same action of another one when satisfied. In such cases, the conflict emerges when one norm establishes a condition to its action, and another one expresses the same action definitively.

Table 5.6 – Norm pairs annotated as object-conditional conflict in norm dataset fold with the respectives NLI-Model predictions. We include the confidence level of NLI-Model for each predicted NLI class: Contradiction (C), Entailment (E) and Neutral (N).

Norm Pair	**a,b**	**b,a**
(a) If such action by Apple will impact SCI's cost or delivery schedule such cost or schedule will be equitably adjusted. (b) Costs and schedule will not be adjusted.	C (99,80%)	C (51,57%)
(a) MOPAC shall not be liable to Biopure for consequential damages arising out of System shutdown caused by any event of force majeure. (b) If natural causes System to shutdown MOPAC will be responsible for the damages.	C (44,78%)	C (75,93%)
(a) All prices quoted are exclusive of federal state and local excise sales use and similar taxes and any duties and VR Research shall be responsible for all such items. (b) VR Research may be liable for any of these items if any where they would be federal state and local taxes sales tax use and similar and any rights.	E (84,44%)	N (51,24%)

The two first rows of Table 5.6 describe contradictions between one norm that contains a conditional action that declares the opposite action of another one. In the second norm pair, the norm (a) indicates that the MOPAC shall not be liable for damages caused by any event while the norm (b) includes a conditional event that MOPAC may be responsible for damages. Thus, this norm pair presents a contradiction not only in norm action but also in its conditions to perform it.

The third row shows an entailment relation between the norm pair (a,b) since both norms express the same action. On the other hand, it is not true in the reverse pair (b, a) given that the premise norm imposes a condition to perform the hypothesis norm action. In this case, the hypothesis is not necessarily true since the premise introduces a condition to be valid and, consequently, the NLI-Model cannot infer the entailment relation in norm pair (b, a).

5.2 Transitive Conflicts

In this section, we explore transitive relations to find potential contractual problems that may arise from more than two sentences using the NCC-Model$_{\text{nli-ft}}$. We select entailed norm pairs containing obligation × permission relation and create a third norm that involves a prohibition that conflicts with the pair. Formally, given an entailed norm pair (n_1,n_2), we introduce a norm n_3 that conflicts with n_1 and, transitively, conflicts with n_2 as well. In the following example, we intentionally create the n_3 with a deontic-modality conflict with n_1 with the same norm structure, associating a prohibition that concerns the same norm object and action.

- n_1: Customer should notify USF at least days in advance of special promotions that may cause unusual or excessive demand on inventory.
- n_2: Customer may inform USF before any promotions that may result in an unexpected demand on inventory.
- n_3: Customer should not notify USF at least days in advance of special promotions that may cause unusual or excessive demand on inventory.

The NLI-Model predicts the relation between n_1 and n_2 as entailment with 97% confidence and between n_2 and n_1 as neutral with 78% confidence. Due to their different norm structure, the NCC-Model$_{\text{nli-ft}}$ predicts a *deontic-structure* conflict between n_1 and n_2 with 60%. Since the norm structure between n_2 and n_3 are different as well as n_1 and n_2, NCC-Model$_{\text{nli-ft}}$ predicts a *deontic-structure* conflict between n_2 and n_3 with 96% illustrating a transitive conflict between an entailed norm pair with another norm. Such a complicated relation shows that by using both models jointly we could infer other conflicts based on entailed norms. Although we guided this experiment concerning logic aspects, such transitive relations are not strictly logical since an NLI model deals with the informal and ambiguous reasoning encoded by natural language.

5.3 Exploring the Self-Attention Weights

In this section, we show the relevance of each term to predict a specific inferential relation or conflict type reporting its attention weights. To help humans to investigate normative conflicts, we explore the attention weights to highlight words in a norm pair showing the correlation with its conflict computed by the neural network. In this analysis, we use the NCC-Model$_{\text{nli-ft}}$ instead of NCC-Model due to its better performance on the Norm Dataset test set, as stated in Chapter 4. Since our model uses only the *CLS* token to predict the output label, we show in this analysis only the attention weights computed for the *CLS* token. Voita *et al* analyze how to interpret each head contained in multi-head self-attention for the neural machine translation task describing specifics syntactic relation between words [36]. However, we could not find a clear pattern of each head in our classification tasks and, hence, we use the average value of all heads in this analysis.

In conflicts that arise from different deontic meanings, we note that the attention weights produced by NCC-Model$_{\text{nli-ft}}$ show a strong correlation with the modal verb. We use as an example the following pair with deontic-modality conflict, which is the same pair of the first row of Table 5.3:

- n_1: CBSI will retain the originals in its archives
- n_2: CBSI may retain the originals in its archives

The NCC-Model$_{\text{nli-ft}}$ predicts the correct label for this norm pair with high attention weights for the modal verb of the second norm. Regarding its inferential relation, we note that the attention weights produced by the NLI-Model reflect the same modal verb pattern of NCC-Model$_{\text{nli-ft}}$. Such attention weights to predict this entailment relation corroborate our analysis of Section 5.1.2. Figures 5.1 and 5.2 illustrate the attention weights of NCC-Model$_{\text{nli-ft}}$ and NLI-Model respectively of this example for each subword unit.

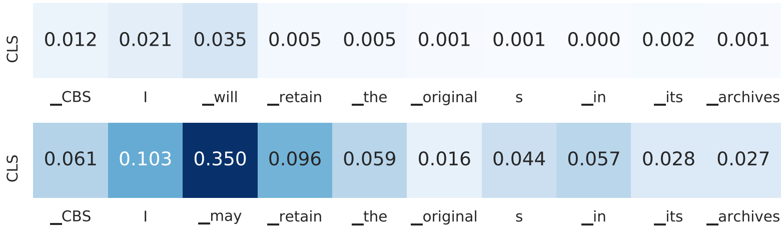

Figure 5.1 – Heat map with attention weights produced by NCC-Model$_{\text{nli-ft}}$ for the norm pair of the first row of Table 5.3, which contains a *deontic-modality* conflict. Due to space limitations, we divide the self-attention vector in two row to represent each sentence in a row and we omit the special characters.

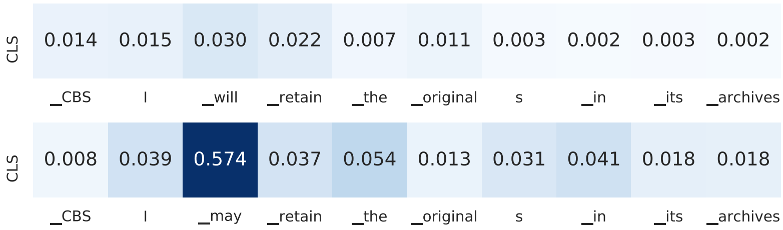

Figure 5.2 – Heat map with attention weights produced by NLI-Model for the norm pair of the first row of Table 5.3, which contains a entailment relation.

We use the next sentence pair to show a more diverse example, which contains a *deontic-structure* conflict and a contradiction as well:

- n_1: No material changes may be made to the Headcount Plan without the prior approval of the Global Supply Team.
- n_2: Parties may perform changes on materials whenever they want.

We extracted the norm pair above from the last row of Table 5.5. Although this pair contains sentences with different structures, the NCC-Model$_{\text{nli-ft}}$ could detect the divergence between deontic meanings highlighting the modal verb. The NLI-Model detects which terms of n_1 contradicts the n_1, resulting in high attention weights for the word “may”, which is the modal verb negated indirectly, and “whenever”, which reinforces the permission of material changes. Figures 5.3 and 5.4 illustrate the attention weights of NCC-Model$_{\text{nli-ft}}$ and NLI-Model respectively of this example.

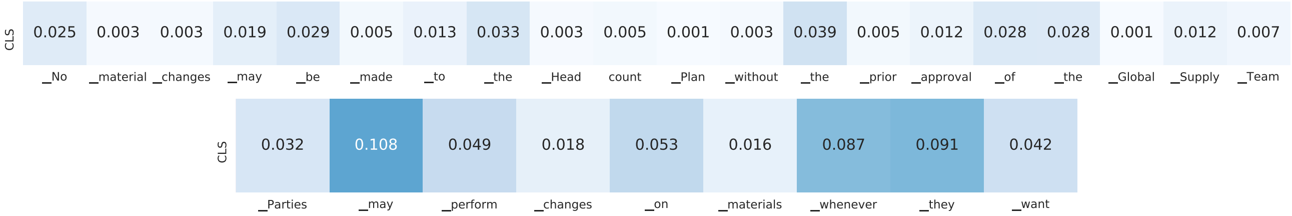

Figure 5.3 – Heat map with attention weights produced by NCC-Model$_{\text{nli-ft}}$ for the norm pair of the last row of Table 5.3, which contains a *deontic-structure* conflict.

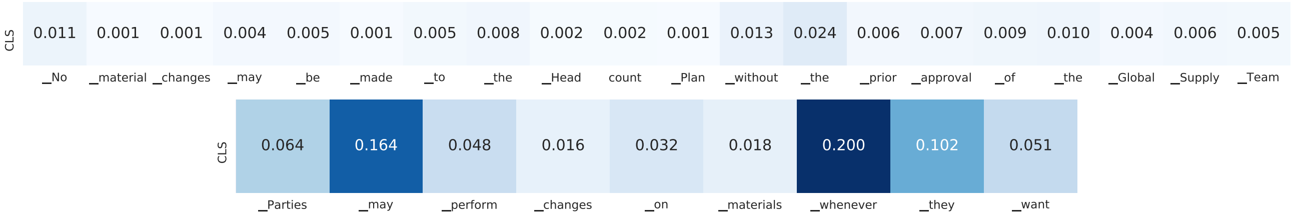

Figure 5.4 – Heat map with attention weights produced by NLI-Model for the norm pair of the last row of Table 5.3, which contains a contradiction relation.

We use the following norm pair, extracted from the second row of Table 5.5, to report attention weights of *deontic-object* conflict type:

- NCR shall pay the freight carrier directly.
- NCR must pay the freight carrier mkaing a back deposit.

In this case, we note that the attention weights of both models highlight the conflicting object in the second norm. Since the object diverges between norms, the attention weights of both models result in high values on object words. Figure 5.5 shows an example of attention weights of a norm pair with *deontic-object* conflict, highlighting the divergence of the payment form to the freight carrier. Similarly, Figure 5.6 shows the attention weights of this norm pair that NLI-Model computes to predict the contradiction relation.

The following norm pair represents an *object-conditional* conflict and contains an entailment relation:

- All prices quoted are exclusive of federal state and local excise sales use and similar taxes and any duties and VR Research shall be responsible for all such items.

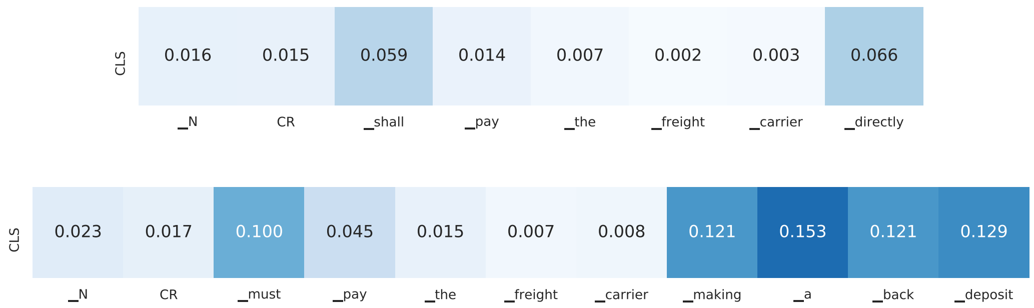

Figure 5.5 – Heat map with attention weights produced by NCC-Model$_{\text{nli-ft}}$ for the norm pair of the second row of Table 5.5, which contains a *deontic-object* conflict.

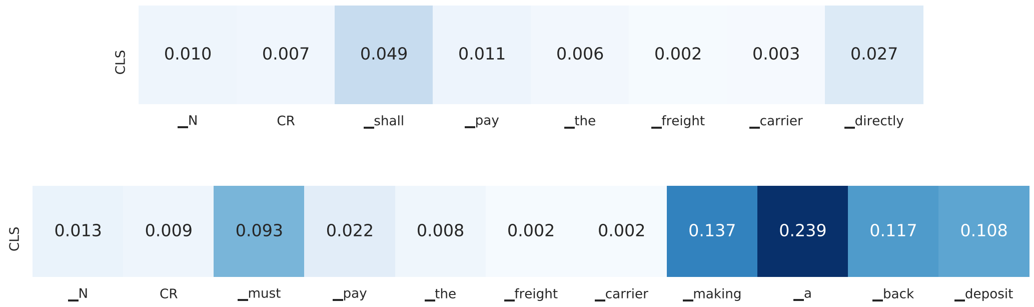

Figure 5.6 – Heat map with attention weights produced by NLI-Model for the norm pair of the second row of Table 5.5, which contains a contradiction relation.

- VR Research may be liable for any of these items if any where they would be federal state and local taxes sales tax use and similar and any rights.

As well as previous norm pairs, the NCC-Model$_{\text{nli-ft}}$ highlights the modal verb assigning high attention weight to the verb of the second norm. Given that this conflict type relies on norm conditions, we note that the NCC-Model$_{\text{nli-ft}}$ also highlights the conditional word “if”, which in this example contains the highest attention weight of the norm pair. Figure 5.7 shows the attention weights of this norm pair that NCC-Model$_{\text{nli-ft}}$ uses to predict the *object-conditional* conflict.

In this analysis, we explore the attention weights of norm pairs produced by our approach to discuss which words it considers to be the cause of the conflict and which words reflect its inferential relation. Importantly, we argue that our models can not only detect the conflict type or inference relation but also help human experts to fix potential problems in contracts. We show that the attention weights can guide humans to fix such problems indicating which norm words should be adjusted relying on its attention weights.

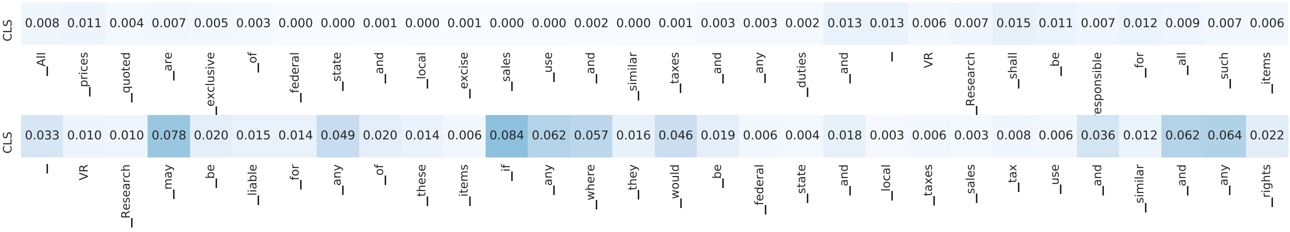

Figure 5.7 – Heat map with attention weights produced by NLI-Model for the norm pair of the last row of Table 5.6, which represents an *object-conditional* conflict.

5.4 Approach Limitations

In this section, we discuss the limitations of our approach and describe related work that covers topics related to the dataset linguistic bias. Although the NLI-Model yields accurate results on the MultiNLI dataset, we observe some limitations while evaluating this model on normative sentences, which motivate us to investigate such failures in recognizing the norms inferential relation. First, we describe the Gururangan *et al* [17] work that raises uncertainty about the success of NLI systems exposing issues of NLI datasets. Second, we describe the uncertainty between neutral and contradiction class that Willians *et al*[37] raises in the MultiNLI dataset. Finally, we discuss limitations about inferential relations that rely on temporal aspects.

Gururangan *et al* implement an off-the-shelf classifier to predict NLI labels using only the hypothesis sentence to examine the statistic cues in sentence pair annotations [17]. With a model that is oblivious to the premise, this classifier predicts the correct NLI label in 67% of sentence pairs contained in the SNLI dataset and 53% in the MultiNLI dataset. Such results show that the dataset contains an annotation pattern that introduces a bias to the hypothesis sentence, which leaves clues to identify the inferential relation without considering the premise sentence. Our analysis of the attention weights produced by our models corroborates with this hypothesis bias, given that the second sentence receives the highest attention values in the NLI-Model.

To see whether a certain word contains a bias for an inference class, Gururangan *et al* compute the point-wise mutual information (PMI) between each word and class in training dataset for SNLI and MultiNLI as stated in Equation 5.1. For example, negation words such as *none, no, never and nothing* are high frequency in contradictory pairs, which means that these words are strong indicators of contradiction. Table 5.7 shows example of PMI values of words in the class *contradiction*. This analysis corroborates the intuition that the NLI-Model contains such bias given that 38 of 48 non-conflicting pairs classified as contradictions contain at least one of the four most PMI in MultiNLI contradiction pairs.

$$PMI(word, class) = \log \frac{p(word, class)}{p(word)p(class)} \tag{5.1}$$

word	**PMI(word, 'contradiction')**
never	5.0%
no	7.6%
any	4.1%
nothing	1.4%
none	0.1%

Table 5.7 – Top 5 words ordered by its PMI with the NLI class contradiction.

We note that the annotation process of the MultiNLI dataset does not necessarily follow the strict definition of the NLI contradiction, stated in Section 5.1.3. The standard adopted in this process is the definition introduced by De Marneffe *et al*: sentences A and B are contradictory if there is no possible world in which A and B are both true [11]. De Marneffe *et al* argues that coreference is essential describing that a contradiction occurs when both sentences refer to the same event. However, the lack of coreference information of each sentence in MultiNLI turns uncertain the difference between the labels contradiction and neutral labels. For example, the first row of Table 5.5 contains a pair of sentences in which the hypothesis sentence does not infer a negation of the premise. On the other hand, we can consider that the date divergence between norms represents a mutual exclusion considering that both sentences refer to the same event. Thus, the mutual exclusion denotes a contradiction since the norm subject cannot perform the same action on different dates.

Besides such dataset limitations, we note that our NLI-Model could not detect inferential relations that relies on temporal aspects. Where two norms contain a deadline for compliance, if one of the norms contains an earlier deadline for the same norm than the second, the earlier deadline implies that the later deadline will be fulfilled, which is not true on the other way. However, the NLI-Model tends to consider this time-related object divergence as a contradiction as shown in the first row of Table 5.5 and, therefore, ignoring the temporal relation between the norms. As well as other issues raised above, our intuition is that such limitations are related to our training dataset, which does not cover time associations enough to train the model to infer such relations. Willians *et al* include in the MultiNLI dataset sentences with terms that represent abstract temporal interpretation, (e.g., then, today) and month names and days of the week [37], but not sentences with finer temporal details as illustrated in the first row of Table 5.5.

6. RELATED WORK

In this chapter, we present related works that analyze normative sentences and contract clauses. We describe the related works explaining the problem dealt, their objectives and how they represent a normative sentence as well as its conflict detection approach. Finally, we compare the objective of this work with the related work and discuss the differences.

6.1 Norm Conflict Identification in Contracts

Aires [3] develop an approach that identifies potential conflicts between norms in contracts. They divide their approach into two steps. First, they focus on norm identification, which results in a formal representation of a norm. Second, they use the formal representation to detect and classify potential conflicts between norms using techniques of the formal logic.

In the norm identification step, they evaluate a sentence written in the natural language to determine if it is a norm or not. This assumes that a norm follows a well-defined 4-component structure: an indexing number or letter, one or more named parties, a modal verb, and a behavior description. Given this structure, they apply a regular expression to decide whether a sentence is a norm sentence or not. After identifying norms, they create a formal representation of the norm sentence extracting three components: party name, deontic meaning, and the norm action. With the formal representation, they detect potential conflicts following three relations between deontic meaning in norm pairs [31]:

- Permission and Prohibition
- Permission and Obligation
- Obligation and Prohibition

As the pre-requisites to apply this comparison, the norm pairs assume the following conditions:

- Both norms are applied to the same party
- Both norms have conflicting deontic meanings
- Both norms refer to the same act

Instead of using a formal representation to use a strict logic approach, we intend to explore the use of techniques that deal with the informal reasoning of natural language.

While this approach deals with normative conflict detection as binary classification (contains a conflict or not), we handle this problem as multi-label classification relying on a conflict typology. In our multi-model approach, we develop a neural network considering the dataset introduced by Aires *et al*[1], which contains the conflict type specified.

6.2 Norm Conflict Identification using Deep Learning

Aires and Meneguzzi [2] approach uses a *LeNet* CNN [21] to process a norm pair at a character level to classify whether a norm pair contains a conflict. Their approach represents a norm pair (n_1, n_2) as a matrix that consists of the characters from n_1 in its lines and n_2 in its columns. The matrix cell value is 1 when the character of its line and column are equals and 0 otherwise. The *LeNet* CNN receives this norm pair representation and processes the matrix using two convolutional layers followed by a max-pooling layer and two fully connected neural networks.

Although we use a deep learning approach as well as this work, we process a norm pair in a word-level instead of character level using a novel neural network architecture. Instead of using matrix representation, we represent each norm pair as a single vector. Our approach results in a higher accuracy considering the typology introduced by Aires et al [1] when compared with their work.

6.3 Classification of Contractual Conflicts via Learning of Semantic Representations

Aires *et al* [1] introduce a typology of conflicts in normative sentences and present machine learning methods that classify these conflict types. These learning methods rely on the semantic representation of norms using Sent2Vec [27] to create embedding vectors to represent the norms. First, they describe an extension of Aires Norm Dataset to include the conflict typology, which introduces 228 new conflicting norms including the existing 111 from the previous dataset. Second, they present an unsupervised learning method to detect the presence or absence of norm conflicts. Finally, they present a supervised learning method that deals with binary (i.e., conflicts and non-conflict) and multi-class classification method to classify the conflict types created.

They manually extend the Norm Dataset to include the following conflict types: *deontic modality*, *deontic structure*, *deontic object* and *object conditional*. The *deontic modality* is the conflict based on the divergence of deontic meanings between the modal verbs of the pair of norms. The *deontic structure* occurs when both of deontic meaning and sentence structure are different given a pair of norms. The *deontic object* conflict type emerges from the difference of norm actions and specification details, although the deontic meaning may be the same. Finally, the *object conditional* type appears when a conflict occurs in the condition of actions in a pair of normative sentences.

Table 6.1 – Examples of norm pairs with the respective conflict type.

Norm Pair	**Conflict Type**
- The Specifications may be amended by the NCR design release process. - The Specifications shall not be amended by the NCR design release process.	deontic modality
- All inquiries that Seller receives on a worldwide basis relative to Buyer's air chamber "Products" as specified in Exhibit III, shall be directed to Buyer. - Seller may not redirect inquiries concerning Buyer's air chamber "Products".	deontic structure
- Autotote shall make available to Sisal one (1) working prototype of the Terminal by May 1, 1998. - Autotote shall make available to Sisal one (1) working prototype of the Terminal by June 12, 1998.	deontic object
- The Facility shall meet all legal and administrative code standards applicable to the conduct of the Principal Activity thereat. - Only if previously agreed, the Facility ought to follow legal and administrative code standards.	object conditional

The unsupervised learning method consists of including a centroid that represents the norm pairs with conflicts and another centroid to represent non-conflicting norm pairs. They compute the centroids based on two different distance-based approaches to calculate the mean of embedding space. The first approach considers the concatenation of norm pairs and the second uses the offset embeddings of norm pairs. Given an unseen concatenated norm pair, this method selects the output class based on the near centroid in the embedding space.

In supervised learning, they use a Support Vector Machine (SVM) [8] to compute a hyperplane that maximizes the margin between the norm pairs of different classes. In this learning method, they represent the norm pairs using both approaches used in unsupervised learning (concatenation and offset). To investigate which conflict type is hardest or easiest, they execute the SVM removing non-conflicting pairs of the dataset.

In contrast with this work, which represents each normative sentence in a distinct embedding vector, we develop a neural network that receives both normative sentences as a single sequence. Using the self-attention mechanism, we develop a neural network that learns each token representation carrying information not only of tokens contained in its sentence but tokens in another pair sentence. Our approach surpasses the Aires *et al* [1] work in all measure metrics with a large margin.

7. CONCLUSION AND FUTURE WORK

In this work, we developed an approach to identify potential problems in contract clauses. We summarize our contribution in two parts: first, we created a model to recognize inferential relations between norms in a contract; second, we created a model to identify conflicts between norms in contracts and its conflict type. We show that using such models jointly allows more sophisticated reasoning about the deontic meanings conveyed in the sentences, as well as reasoning about transitive conflicts (i.e. conflicts involving the combination of more than two clauses).

We developed both models using a novel neural network approach that relies on a pretraining method that enables learning bidirectional context of words, known as XLNet. Using pretrained weights, we train a neural network by finetuning to the NLI task, which contains a wide openly annotated datasets. To identify conflicts between normative sentences, we use the same pretrained weights to train a neural network by finetuning to the task of norm conflict classification using the dataset and the conflict typology provided by Aires *et al*[1]. In contrast with the first model, the volume of annotated data is scarce, motivating us to experiment with the use of transfer learning by reusing the trained NLI model.

We report a set of experimental analyses that shows aspects that our models could capture from normative sentences regarding its inferential relation and conflict. We show that the divergence of deontic meaning not only shows a conflict but also can determine its inferential relation. The attention weights generated by our models show that our approach can highlight terms correlated with a specific conflict or its inferential relation, which its a feature that can help human experts indicating potential problematic terms in a set of norms.

Dealing with both tasks jointly opens a number of possible avenues of further research, motivating 3 avenues for future work: First, we aim to evaluate our NLI model in more diverse NLI datasets that can complement the limitations of the MultiNLI dataset as discussed in Section 5.4. Second, we aim to create a new typology of conflicts in contracts that combine the deontic meaning conveyed in the sentences with its logical relations. Third, we aim to create a tool that uses both models to help human experts in the law area by dealing with real cases of contractual problems.

REFERENCES

[1] Aires, J. a. P.; Granada, R.; Monteiro, J.; Barros, R. C.; Meneguzzi, F. “Classification of contractual conflicts via learning of semantic representations”. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019, pp. 1764–1766.

[2] Aires, J. P.; Meneguzzi, F. “Norm conflict identification using deep learning”. In: Proceedings of the Autonomous Agents and Multiagent Systems, 2017, pp. 194–207.

[3] Aires, J. P.; Pinheiro, D.; Lima, V. S. d.; Meneguzzi, F. “Norm conflict identification in contracts”, *Artificial Intelligence and Law*, vol. 25–4, Dec 2017, pp. 397–428.

[4] Aires, J. P.; Pinheiro, D.; Meneguzzi, F. “Norm Dataset: Dataset with Norms and Norm Conflicts”. Source: https://doi.org/10.5281/zenodo.345411, 08/12/19.

[5] Bos, J.; Markert, K. “Recognising textual entailment with logical inference”. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, 2005, pp. 628–635.

[6] Bowman, S. R.; Angeli, G.; Potts, C.; Manning, C. D. “A large annotated corpus for learning natural language inference”. In: Proceedings of the Empiricial Methods in Natural Language Processing, 2015, pp. 632–642.

[7] Carmo, J.; Jones, A. J. I. “Deontic Logic and Contrary-to-Duties”. Dordrecht: Springer Netherlands, 2002, chap. 4, pp. 265–343.

[8] Crammer, K.; Singer, Y. “On the algorithmic implementation of multiclass kernel-based vector machines”, *Journal of Machine Learning Research*, vol. 2, Dec 2001, pp. 265–292.

[9] Dagan, I.; Glickman, O.; Magnini, B. “The pascal recognising textual entailment challenge”. In: Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment, 2006, pp. 177–190.

[10] Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J. G.; Le, Q.; Salakhutdinov, R. “Transformer-xl: Attentive language models beyond a fixed-length context”. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 2978–2988.

[11] De Marneffe, M.-C.; Rafferty, A. N.; Manning, C. D. “Finding contradictions in text”. In: Proceedings of Association for Computational Linguistics - Human Language Technologies, 2008, pp. 1039–1047.

[12] de Souza Aires, J. P. “Identifying potential conflicts between norms in contracts”, Master’s Thesis, Faculdade de Informática – PUCRS, Porto Alegre, RS, Brasil, 2015, 67p.

[13] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: Proceedings of Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

[14] Dietterich, T. G. “Approximate statistical tests for comparing supervised classification learning algorithms”, *Neural Computation*, vol. 10–7, Oct. 1998, pp. 1895–1923.

[15] Goodfellow, I.; Bengio, Y.; Courville, A. “Deep Learning”. MIT Press, 2016, 800p.

[16] Griffiths, H. “Introduction to sociology”. Open Textbook Library, 2015, 501p.

[17] Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; Smith, N. A. “Annotation artifacts in natural language inference data”. In: Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 107–112.

[18] Ioffe, S.; Szegedy, C. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, 2015, pp. 448–456.

[19] Kingma, D. P.; Ba, J. “Adam: A method for stochastic optimization”. In: Proceedings of the 3rd International Conference on Learning Representations, 2015, pp. 1–15.

[20] Lakoff, G. “Linguistics and natural logic”, *Synthese*, vol. 22–1, Dec 1970, pp. 151–271.

[21] LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; Jackel, L. D. “Handwritten igit recognition with a back-propagation network”. In: Proceedings of the Advances in Neural Information Processing Systems, 1990, pp. 396–404.

[22] Maccartney, B. “Natural language inference”, Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2009, 165p.

[23] MacCartney, B.; Manning, C. D. “An extended model of natural logic”. In: Proceedings of the 8th International Conference on Computational Semantics, 2009, pp. 140–156.

[24] McNemar, Q. “Note on the sampling error of the difference between correlated proportions or percentages”, *Psychometrika*, vol. 12–2, Jun 1947, pp. 153–157.

[25] Mitchell, T. “Machine Learning”. McGraw-Hill, 1997, 414p.

[26] Murphy, K. P. “Machine Learning: A Probabilistic Perspective”. The MIT Press, 2012, 1104p.

[27] Pagliardini, M.; Gupta, P.; Jaggi, M. “Unsupervised learning of sentence embeddings using compositional n-gram features”. In: Proceedings of Association for Computational Linguistics: Human Language Technologies, 2018, pp. 528–540.

[28] Pascanu, R.; Mikolov, T.; Bengio, Y. “On the difficulty of training recurrent neural networks”. In: Proceedings of the International Conference on Machine Learning, 2013, pp. 1310–1318.

[29] Rousseau, D. M.; McLean Parks, J. “The contracts of individuals and organizations”. JAI Press Ltd, 1993, 43p.

[30] Russell, S.; Norvig, P. “Artificial Intelligence: A Modern Approach”. Prentice Hall Press, 2009, 1152p.

[31] Sadat-Akhavi, A. “Methods of resolving conflicts between treaties”. Martinus Nijhoff Publishers, 2003, 273p.

[32] Sennrich, R.; Haddow, B.; Birch, A. “Neural machine translation of rare words with subword units”. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 1715–1725.

[33] Sutskever, I.; Vinyals, O.; Le, Q. V. “Sequence to sequence learning with neural networks”. In: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[34] Valencia, V. S. “Studies on natural logic and categorial grammar”, Ph.D. Thesis, Universiteit van Amsterdam, 1991, 197p.

[35] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. “Attention is all you need”. In: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[36] Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. “Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned”. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5797–5808.

[37] Williams, A.; Nangia, N.; Bowman, S. “A broad-coverage challenge corpus for sentence understanding through inference”. In: Proceedings of Association for Computational Linguistics: Human Language Technologies, 2018, pp. 1112–1122.

[38] Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Brew, J. “Huggingface’s transformers: State-of-the-art natural language processing”, *ArXiv*, vol. abs/1910.03771, Oct 2019, pp. 1–11.

[39] Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; Le, Q. V. “Xlnet: Generalized autoregressive pretraining for language understanding”. In: Proceedings of the Advances in Neural Information Processing Systems, 2019, pp. 5754–5764.