Title	Classifying sentential modality in legal language: A use case in financial regulations, acts and directives
Author(s)	O'Neill, James; Buitelaar, Paul; Robin, Cécile; O'Brien, Leona
Publication Date	2017-06-12
Publication Information	O'Neill, James, Buitelaar, Paul, Robin, Cécile, & O'Brien, Leona. (2017). Classifying sentential modality in legal language: a use case in financial regulations, acts and directives. Paper presented at the 16th International Conference on Artificial Intelligence and Law London, London.
Publisher	ACM
Link to publisher's version	https://dl.acm.org/proceedings.cfm
Item record	http://hdl.handle.net/10379/6845
DOI	http://dx.doi.org/10.475/123_4

Classifying Sentential Modality in Legal Language: A Use Case in Financial Regulations, Acts and Directives

James O' Neill
Insight Centre for Data Analytics
IDA Business Park
Galway, Ireland
james.oneill@insight-centre.org

Paul Buitelaar
Insight Centre for Data Analytics
IDA Business Park
Galway, Ireland
paul.buitelaar@insight-centre.org

Cecile Robin
Insight Centre for Data Analytics
IDA Business Park
Galway, Ireland
cecile.robin@insight-centre.org

Leona O' Brien
Governance, Risk and Compliance Technology Centre
University College Cork
Cork, Ireland
leona.obrien@ucc.ie

ABSTRACT

Texts expressed in legal language are often difficult and time consuming for lawyers to read through, particularly for the purpose of identifying relevant deontic modalities (obligations, prohibitions and permissions). By nature, the language of law is strict, hence the predominant use of modal logic as a substitute for the syntactical ambiguity in natural language, specifically, deontic and alethic logic for the respective modalities. However, deontic modalities which express obligations,prohibitions and permissions, can have varying degree and preciseness to which they correspond to a matter, strict deontic logic does not allow for such quantitative measures. Therefore, this paper outlines a data-driven approach by classifying deontic modalities using ensembled Artificial Neural Networks (ANN) that incorporate domain specific legal distributional semantic model (DSM) representations, in combination with, a general DSM representation. We propose to use well calibrated probability estimates from these classifiers as an approximation to the degree which an obligation/prohibition or permission belongs to a given class based on SME annotated sentences. Best results show 82.33 % accuracy on a held-out test set.

CCS CONCEPTS

•Applied computing → Law; •Theory of computation → Modal and temporal logics;

KEYWORDS

Sentence Classification, Deontic Modality, Financial Law

ACM Reference format:
James O' Neill, Paul Buitelaar, Cecile Robin, and Leona O' Brien. 2017. Classifying Sentential Modality in Legal Language: A Use Case in Financial Regulations, Acts and Directives. In *Proceedings of International Conference on Artificial Intelligence and Law, London, UK, June 12–15, 2017 (ICAIL'17),* 10 pages.
DOI: 10.475/123_4

ICAIL'17, London, UK

DOI: 10.475/123_4

1 INTRODUCTION

Legal language is often difficult and time consuming for lawyers to traverse through. By nature, law language is strict, hence the use of modal logic to replace the syntactical ambiguity in natural language. This benefits the formulation of rule sets for many applications in the legal domain. However, logical rules are not expressive enough for many tasks within the legal domain. One such application is posed in this work; the degree to which a deontic modality is expressed in legislative texts. The financial law domain is a particularly interesting use case because of the significant uprise in regulatory change seen in recent years, influenced by the financial crisis in 2008. Hence, the experiments are carried out on sub-domains within the financial domain such as Anti-Money Laundering (AML), Serious Organized Crime and Police Act 2005, Financial Instruments Directives, Financial Instruments Regulations and Central Bank Acts. We begin with a background to modality, modality types and the role modal verbs play in modality.

2 MODALITY

Modality is concerned with expressing grammatical moods and attitudes that can come in the form of alethic modality (possibilities, impossibilities and necessities), deontic modality (obligations and permissions) and dynamic modality (ability). Modality can be broadly split into two categories, propositional modality and event based modality. Propositional modality deals with epistemic (judgement on a proposition) and evidential (provided evidence about the factual proposition) speaker judgements. Event modality is the attitude towards events that have not yet happened, usually containing modal verbs, such as "will", "must", "shall". The most common event modality in legal text is deontic, containing obligatory and permissive statements. Dynamic modalities are also considered events which express an agents ability to carry out an action, as opposed to deontic modality where the speaker expresses *what ought to be* or *what ought to do*, usually the latter in legal language. We focus on deontic modality as it is ubiquitous within EU/UK

legislation and many of lawyers needs revolve around regulatory compliance, often expressed in terms of deontic modalities.

2.1 Deontic Modality

Deontic modality, greek for "duty", is an event based modality that is used to reason about norms. These modalities are used ubiquitously for representing legal knowledge, in the form of obligations, prohibitions and permission. The degree to which deontic statements are applied is important from a pragmatic perspective. It is necessary for law practitioners to identify obligations/prohibitions and permissions in updated legislation on a regular basis, particularly in financial law, an area that is growing in demand for semi-automated solutions for fundamental tasks. Acquiring knowledge about the degree and preciseness to which these deontic modalities are expressed allows the lawyer to preference sections within legislation.

2.2 Modal Verbs

Modal verbs consist of "can", "could", "may","will", "would", "should" and "shall", the latter 3 used for expressing deontic modality. The modal verbs "will" and "can" often express the volition of an action and are not explicitly modalities in the sense that it is not expressing the attitude of the speaker but rather the associated ability for the subject to carry out an action. These types of dynamic modalities can be considered semantically further apart from deontic and epistemic modalities, although they often co-occur in the same sentence. As described by Verstraete [28], it is difficult to quantify the semantic range between modalities, hence the proposed use of probability estimates instead of delimiting the semantic relationship between modal classes. Additionally, modal verbs can have more than one function; "may" often expresses permission, but can imply possibility e.g "Member States may decide that persons that engage in a financial activity on an occasional basis are at little risk of terrorist financing and do not fall within the scope of this Directive"[1]. Moreover, the misuse of modal verbs is another complexity e.g *"Authorities shall gain access to legal tender in the State for the payment of any amount."*. Here, "shall" expresses a permission instead of an obligation, where "may" would be more suitable. Likewise *Authorities shall then have the right to carry out an investigation according to section..* expresses the verb "will" as it expresses an act to be carried out in the future. This also occurs between obligations and necessities, in these cases it is clear that context is important for deciphering what is meant, and it is thus a motivating factor for a data-driven approach to overcome these challenges. Using classical logic in these scenarios to formulate rules for a legal application can be ineffective. Artificial Neural Networks (ANNs) have been effective for many text classification tasks, hence we introduce the two most widely used variants of ANNs. But first introduce competitive classifiers that have shown good performance in the past.

3 NON-NEURAL NETWORK CLASSIFIERS

This section provides a brief description the models tested that are not trained in the same manner as ANNs. Evaluating against these models as a baseline provides more clarity as to if ANNs are the most suitable classifier for the given task. We start with an introduction to logistic regression, a stable and well known probabilistic classifier.

3.1 Logistic Regression

Logistic regression is a linear model that produces posterior probabilities for the output using a sigmoid function. The sigmoid function is shown in equation 1,

$$\hat{y} = \frac{1}{1 + e^{-(w^T x + b)}} \tag{1}$$

where the transposed parameters w^T are multiplied by the input x and where the bias term b accounts for the intercept to allow the sigmoid to adjust its position on the x-axis, without it the sigmoid is centered around the origin which is not desired unless data points are known to be generated around the origin [2] Maximum likelihood estimation (MLE) is commonly used to estimate the parameters w, which attempts to find the parameters that maximize the likelihood of producing the dataset observations (in this case sentences), although in practice it is more convenient to maximize the log-likelihood as it requires summations instead of multiplications, resulting in less computation and eliminating the likelihood of underflow (when a value is too small to store in memory). Thus, the log-likelihood is expressed as,

$$log(L(w)) = \sum_{i=1}^{M} y^{(i)} log(\hat{y}^{(i)}) + (1 - y^{(i)}) log(1 - \hat{y})) \tag{2}$$

where $\hat{y}$ is the class estimate shown in equation 2 and y represents the actual class (e.g a deontic class), computed for M number of observations (e.g legal sentences) in the training set. Stochastic Gradient descent (SGD) (Stochastic because we only use one observation to update our weights w and b instead of using all observations in D) can then be used to tune the parameters as shown in equation 3 according to the equation 2 loss function, where α is the learning rate used to move weights w and b to their optimal values according to MLE.

$$w_j := w_j - \alpha(y^{(i)} - \hat{y}^{(i)}) x_j^{(i)} \tag{3}$$

The final produces well calibrated posterior probabilities that represent the probability of a sentence belonging to a particular type of deontic class. Although, linear classifiers are quite stable they trade-off in expressiveness where the boundary can not be separated linearly. In these cases, careful feature engineering is often used in practice to find a linearly separable plane before using logistic regression.

3.2 Support Vector Machines

Support Vector Machines (SVMs), first introduced by Vapnik and Cortes [7], try to find a boundary between classes by finding the widest margin, where the support vectors are the instances that define the boundary to separating the boundary. In contrast, logistic regression can only deal with linearly separable classes, SVMs can learn a nonlinear boundary between classes by using high order

[1] http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32005L0060

[2] From herein, we denote the sigmoid function as σ for subsequent sections.

kernels while maintaining a wide margin [3] between classes that improves the classifiers ability to generalize well on unseen data (legal sentences in the context of this work) in prediction. Widely used kernels are the radial basis function (RBF) and the polynomial function. Equation 4 shows the loss function which is the goal to minimize,

$$L = \frac{1}{2}|w|^2 - \sum_{i=1}^{M} \lambda_i [y(w^T x + b) - 1] \tag{4}$$

where $\frac{1}{2}|w|^2$ is the unconstrained loss term and the remainder represents the constraint posed for finding the widest margin between the classes[4]. Partial derivatives $\frac{\partial L}{\partial X}$ and $\frac{\partial L}{\partial b}$ are then obtained to minimize the function [5] and we come to a solution for the loss function as shown in its dual form [6] in equation 5:

$$L = \sum_{i=1}^{M} \lambda_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{N} \lambda_i \lambda_j y_i y_j x_i * x_j \tag{5}$$

As previously mentioned this $x_i * x_j$ dot product pair can be transformed using a kernel which is referred to as the "kernel trick". SVMs are used extensively for text classification and have long been established in this area [13, 14] due their ability to deal with high dimensional and sparse data, resulting in good performance for these tasks.

3.3 Decision Trees

Decision trees (DTs) are another classifier considered for classification. DTs create a tree of decisions by iteratively computing the information gain (IG) for each class c in dataset D. The feature f with the highest IG is a parent node and child nodes appended downstream contain less IG than the parent node. Nodes near the top of the tree have a stronger relationship with the class. The initial step involves computing the entropy (a measure of uncertainty) for all classes C, given as:

$$Entropy(D) = \sum_{i=1}^{C} p_i \log_2 p_i \tag{6}$$

Information gain (IG), shown in equation 7, is used to measure the difference in entropy once the data has been split according to the entropy of an attribute to what it was before splitting on that attribute. This measures how much uncertainty was lessened once an attribute has been split. This carried out iteratively down the tree to, usually to a specified tree depth.

$$IG(D, F) = Entropy(D) - \sum_{i=1} \frac{|D_i|}{|D|} Entropy(D_i) \tag{7}$$

The C4.5 algorithm is a decision tree developed by Quinlan [23] and is the decision tree that is used in this work. The C4.5 is an extension of the original ID3 algorithm (also developed by Quinlan) by incorporating continuous attributes (in our case, continuous word vectors), missing data, tree pruning (a way of removing decision paths that are redundant or have not occurred in D) and computational speed ups. Decision trees have the advantage of clearly explaining the decision process which is quite an attractive proposition in the legal domain. However, decision trees can be prone to overfitting [7] even when the tree pruning and tree depth are chosen appropriately to mitigate this. A common way to mitigate this is by ensembling many decision trees (or any classifiers for that matter) together in an attempt to better generalize on unseen data.

3.3.1 ***Ensembling of Decision Trees.*** Ensembles are a family of meta learning algorithms that take existing machine learning classifiers and ensemble a number of them together which are then used together to produce an output according to a scoring criterion. The ensemble approach can be used to combine more than one type of classifier, also referred to as stacking. However, in this work we use an ensemble of the same classifier, namely a C4.5 decision tree. These ensembles employed can be broadly split into two approaches:

Bagging. Bagging refers to an ensemble of decision trees where each decision tree is built from bootstap sampling a dataset D. The Random forests algorithm further samples on the features (e.g word vectors) as well as the instances (e.g the sentences) when building decision trees, hence their name. The model then takes a vote amongst all classifiers to decide the final classification. [8] This has the advantage of reducing bias of more complex trees since the model is .

Boosting. In contrast, boosting attempts to weight "weak" classifiers iteratively for a more expressive model. AdaBoost (also known as Adaptive Boosting) weights each classifier sequentially, placing more weights on misclassified points, typically using an exponential loss function for weighting each instance. The algorithms assumes initial weights for instances to be equal, learns a decision tree $h(t)$ such as C4.5 and then update the weights using an exponential function shown in equation 7,

$$D_{t+1}(i) = \frac{D_t(i) e^{(-\alpha_t y_i h_t(x_i))}}{Z_t} \tag{8}$$

where α_t is the weight given to a single point at time t and Z_t is a normalization term containing the sum of all weighted instances. This is iteratively carried out for up to T number of decision trees. All classifiers take a weighted vote given as $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$ when deciding the overall prediction made. In summary, boosting differs from bagging in that it combines "weak" classifiers for a collectively expressive one in a sequential manner, with low variance. Bagging combines "strong" classifiers in parallel that have low bias and higher variance.

4 DEEP ARTIFICIAL NEURAL NETWORKS

The adoption of machine learning (ML) to legal language has grown in recent years due to advances in the field, particularly the use

[3] This is referred to as constraint optimization.

[4] λ_i is a Lagrange multiplier which are used when minimizing/maximizing a function with constraints

[5] For the sake of brevity we leave some of the algebraic steps out here since they are not essential for the overall understanding

[6] This relates the use of lagrange multpiers for finding the boundary, having the advantage of less computation since any dot pair that are not found to lie on the boundary between the classes (ie. the support vectors) are 0.

[7] Overfitting refers to the use of a function that performs well on the observed data, but poorly on unobserved data.

[8] The voting scheme can use weights, however in this work all C4.5 decision trees are equally weighted.

of deep ANNs. Although, for legal tasks that require human-like cognitive ability, they commonly struggle to perform well. However, as pointed out by Surden [26], there are tasks that benefit from computationally efficient algorithms that attempt to overcome the inadequacies of human reasoning, or at least speeding up human processes drastically. ANNs have re-emerged in recent years due to advances in deep learning, showing exceptional results in fundamental tasks e.g perception [16], speech recognition [12] and machine translation [2]. Two ANN architectures widely used in Natural Language Processing (NLP) are Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) e.g for sentence classification tasks [15, 27]. The former learn spatially-based semantic representations by using locally-connected layers (also referred to as filter mappings), while the latter learn sequential-based features. Long Short-Term Memory (LSTM) networks overcome the limitations associated with traditional RNNs, namely the exploding or vanishing gradient issue [9]. Figure 1 illustrates the memory cell mechanism graphically for a single unit. They allow for longer dependencies using memory gates, meaning that arguments to modal verbs could be learned as opposed being constrained to a narrower context window.

4.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are ANNs that are locally connected as opposed to fully connected layers, resulting in a convolution over the data space. This results in neurons becoming feature detectors in parts of the feature space. In recent years, deep CNN architectures have become quite popular in image recognition tasks, outperforming many SOTA systems for facial recognition, object recognition [16, 25] that were previously thought to be a difficult task both computationally and for significant performance improvements. Although text does not have spatial dependencies like images, the results from text classification tasks have resulted in quite surprising results [15], where the convoluted hidden layers learn high level semantic representations of the raw word vector input. The results would seem to be slightly non-intuitive in comparison to their recurrent counterparts, nonetheless, they have shown impressive results in NLP and have the advantage of being processed in parallel using a Graphical Processing Unit (GPU), a particularly important consideration in NLP when processing large amounts of text.

4.2 Recurrent Neural Networks

4.2.1 Long Short-Term Memory Networks. Long Short-Term Memory (LSTM) networks are a type of RNN that use a gating mechanism that store past information in memory over time and also overcome the vanishing gradient problem apparent in the traditional RNN. Since LSTMs keep information in memory over longer timesteps, it results in fewer gradient multiplications during backpropagation [10] Figure 1 shows the standard configuration of a single LSTM unit.

All w weights denote the parameters for all input word vectors x, while hidden layer weights are represented as u, and b signifying

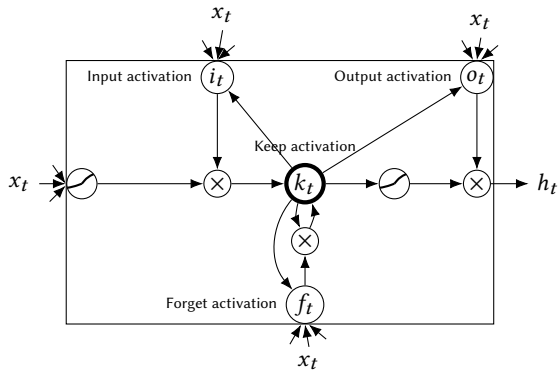

Figure 1: A Single Long Short Term Memory Unit

the bias term for each gating mechanism except the keep gate K. Firstly, the input gate i_t and the potential values for the keep memory cell $\tilde{K_t}$ are computed such that,

$$i_t = \sigma(w_i x_t + u_i h_{t-1} + b_i) \tag{9}$$

where w_i represent weights that are applied for each d dimensions in the input vector, likewise u_i represents the weights to be applied to the hidden layer output which is used as input for timestep $t+1$ combined with the new input vector x_{t+1}. It is common that for each timestep t that the weights are tied over time instead of weights for each t, this reduces the number of degrees of freedom by $1/T$ given that T is the fixed length of a sentence.

$$\tilde{K_t} = tanh(w_k x_t + u_k h_{t-1} + b_k) \tag{10}$$

With the new state of the memory cells, the value of the forget gate f_t is computed as shown in equation 11. In order to learn what dependencies to store over time, the memory mechanism also needs to forget irrelevant information for a given time t.

$$f_t = tanh(w_f x_t + u_f h_{t-1} + b_f) \tag{11}$$

Once the forget gate f_t is computed, the candidate gate K_t is calculated where $\odot$ denotes a pointwise multiplication, as shown in 12. This purpose of this is to filter out information from the previous gate K_{t-1} that has kept dependencies in memory and adjust it for the new input i_t which influences the candidate gate $\tilde{K_t}$.

$$K_t = i_t \odot \tilde{K_t} + f_t \odot K_{t-1} \tag{12}$$

Equation 13 shows output weights w_o, u_o, S_o and bias weight b_o. In the context of this work, o_t is only computed at the end of the sentence (o_T) instead of each time step (however, in tasks such as sequence labeling this is necessary) [11]

$$o_t = \sigma(w_o x_t + u_o h_{t-1} + S_o C_t + b_o) \tag{13}$$

The final filtered output o_t is then multiplied with the output of a $tanh$ nonlinear transformation of K_t as show in equation 9, however other activations can also used in practice.

[9] This is a particular limitation in legal language given that sentences tend to be quite long

[10] The original gradient vanished due to many multiplications of gradients over long time steps. Apart from memory gate based solutions, there has been other approaches to circumvent this issue such as Hessian-free optimization.

[11] Here we demonstrate a sigmoid that is commonly used for binary classification producing output values between 0 and 1. However, we use a *softmax* function for the multi-class problem posed in this work that restricts the sum of all probabilities of each class to sum to 1.

$$h_t = (\tanh K_t) * o_t \quad (14)$$

Connecting these units recurrently for time step $t = 0, \ldots N$ allows the network to learn long-term relations and dependencies, which is particularly useful in legal language as sentences potentially contain nested statements and intricate applicability conditions. Take the following example: "Member States ***shall*** ensure that money laundering and terrorist financing are ***prohibited***". Without the use of memory gates, a standard RNN would likely classify this statement as a prohibition since "...financing are prohibited" is closer to the output, terms at the start of the start of the sentence have much less influence since retaining it in the keep gate dissipates. In contrast, the gated RNN attempts to keep "Member States shall ..." in memory. ANNs also have the capability of learning distributed word representations in an unsupervised manner, which brings us to word embeddings.

4.3 Word Embeddings

Word embeddings are used to represent words according to the distributional hypothesis [10]. The *distributional hypothesis* states that words which have similar meaning will occur within a similar context. This is the basic hypothesis which underlies techniques to embed words as vectors, also known as Distributional Semantic Models (DSMs). There has been a plethora of work in representing semantics in vector space, which can be broadly split into two approaches: 1) count-based vectors using dimensionality reduction techniques such as the use of Singular Value Decomposition (SVD) in Latent Semantic Analysis (LSA), Non-Negative Matrix Factorization (NMF), Global Vectors for Word Representations (*Glove*), and 2) prediction based vectors using gradient based learning (*word2vec*). Comprehensive comparisons in the literature [5] suggest the latter show consistent improvements on a varying number tasks. Mikolov et al. [21] first introduced the skip-gram model [12] to predict a word w given a context c, $p(w|c)$, and vice versa $p(c|w)$ with the Continuous Bag of Words (CBOW) model, encoding the contextual information as continuous vectors. Using similar techniques provides a basis for semantically rich legal vector representations. Therefore, we discuss literature on classification tasks that use word vectors in similar settings, but first we discuss the use of deontic logic for the representation of legal language.

5 RELATED RESEARCH

We start with a brief description of related literature on deontic logic and how it has been applied in the legal domain to reason about legal norms and to construct legal rules. The relevance of covering literature on deontic logic and related modal logic is to address the fundamental construction of legal norms as expressed in legal language, as these features are what the proposed models are essentially attempting to learn, along with the context in which they appear. Moreover, applying probabilistic models for sentential modality classification that incorporate context as opposed to traditional logic based systems is a relatively untested notion. However, we find instances of similar scenarios in the literature.

[12]The skipgram model implements negative sampling to maximize the dissimilarity/similarity between words in quicker manner by sampling contexts, otherwise the model is quite slow to train.

5.1 Standard and Dynamic Deontic Logic in Legal Representations

Deontic logic has been used to reason about legal norms and for curating legal rules required for creating expert systems in the legal domain. In von Wrights work [29], he refers to two types of modal logic; alethic logic and modals of truth. The paper gives an extensive overview of Standard Deontic Logic (SDL) and makes the important distinction that deontic modalities are not connected to factual statements, which makes creating legal rules difficult in this sense, one cannot simply use predicate calculus in such cases. Furthermore, ambiguity arises when applying logic to contradictory or paradoxical legislation making it difficult for lawyers to comply. Hilpinen [11] discuss the theory of SDL, permissions being stated as absent of any prohibitions, which can possibly lead to non-deterministic acts and that permissions have been found to substitute previously known prohibitions, presenting a modification to the original act. In Dynamic Deontic Logic (DDL), the objective is to reason after the action is carried out. Meyer [20] present DDL for when obligations, prohibitions and permissions are abided by to determine what will be the future action, deontic logic is only applied to *"ought to do"* actions. Gordon et al. [9] summarize what is needed in the legal domain for ease of rule exchange using semantic modeling of business vocabularies, facts, and rules, as provided by systems such as RuleML[13], and SBVR [14]. Royakkers [24] discuss the lack of deontic concepts when creating legal rules for distinguishing between SDL *"what ought to be"* legal rules and dynamic deontic logic (DDL) *"what ought to do"*, mentioning the need for a single framework to account for both. McCarty [19] define obligations and permissions by a set of actions, attempting to overcome the paradoxical limitations of SDL. Indirectly, this work can help by providing a probability, $\hat{p}$, that relates to the use of a deontic expression and its associated action/s. Pioneering work from Chellas [6] describe systems for deontic modality tenses, distinguishing between the future tense use of actions to be taken (like that expressed in regulations, acts and directives) as opposed to past tense use which is seen as being critical. The confounding argument for rule-based approaches is that they are well suited to the strict modal logic in legal language. But, for the purpose of analyzing the preciseness of the use of deontic modality, rule-based approaches cannot succinctly quantify this.

5.2 Modality Classification Approaches

Probabilistic classifiers can provide a posterior estimate $\hat{p}$ to predict a deontic class given the legislative context, which can also be interpreted as how precise and non-ambiguous a deontic modality may be. A similar task was addressed by Marasovic and Frank [17] where a CNN architecture was applied for modal sense classification (MSC) in English and German. A follow-up study [18] looked at modal verb senses (deontic, dynamic and epistemic) on a larger scale to show the need for good generalization, similarly they found that skewed proportions of modal senses coupled with small sample sizes led to poor generalization in previous work, which they focused on overcoming. In a more distant setting, Baker et al. [3]

[13]RuleML is available at: http://wiki.ruleml.org/index.php/RuleML_Home
[14]SBVR is available at: http://www.omg.org/spec/SBVR/

applied a modality and negation tagger used for event detection in Statistical Machine Translation (SMT) in order to differentiate between probability and certainty, and the polarity of an event. Maat et al. [8] compared ML approaches to knowledge based approaches for legal text classification in Dutch legislation and found that ML classifiers generalized poorly on new legislation and cross-domains modelling, favoring "pattern" based approaches as they are easily extended and interpretable. However, they express that the lack of annotated data led to poor out of sample results. In our work we are presented with a similar scenario, however our input representations are partly pretrained on a general corpus provided by Google which should help with lack of annotated data. Baker [4] present a modality lexicon for tagging modalities and the respective arguments by using flattened parse trees and rules based on the structure of the sentences for tagging, achieving 86 % precision on a test set. We conjecture that this is a particularly useful application for the legal domain, although the nature of the approach could lead to labored reconstruction of rules for automatic tagging if legislation changes, in comparison to training a parametric model. Asooja et al. [1] proposed modality classification by using semantic frames, n-grams, a negation feature and POS tags. Best results incorporated all features excluding the negation feature, with a weighted $F_1 = 0.712$. These results can be considered as a starting point for approaches considered in this work. However, direct comparisons are not suitable since annotations were performed on the paragraph level. Instead, we have extracted sentence level modalities. Furthermore, our test set consists of held-out documents as opposed to randomly shuffling sentences in documents together in the train and test process. In such cases, testing can lead to biased and unreliable results hence why whole documents are separated as a hold-out test set.

6 METHODOLOGY

6.1 Datasets and Annotation Guidelines

The annotations are carried out by Subject Matter Experts (SME) using the General Architecture for Text Engineering[15] (GATE) text analysis software tool. The training set consists of 1297 SME annotated sentences, including 596 obligations, 94 prohibitions, and 607 permissions. The gold standard test set consists of 6 separate documents: EU/UK AML 2013/2015[16,17], UK AML 2007, 2014 EU Markets in Financial Instruments Directive (MiFID) i/ii, Central Bank Act 1998, Dormant Accounts 2005, EU Consolidated Accounts and Markets in Financial Instruments Regulation (MiFIR) 2014. Accounting for all sentences, there are 312 Obligations, 248 Permissions (some are constraints e.g "may not do the following.."), 62 Prohibitions, all of which are tested for types of deontic meaning. The kappa coefficient (i.e inter-rater agreement) for both SMEs is $\kappa = 0.74$ with only few disagreements, all of which appear between obligation and prohibition classes. Although training and testing sentences are different, if they come from the same document this can introduce bias. For this reason, these sentences are strictly used as a test set and are not used in the training procedure.

[15]GATE homepage: https://gate.ac.uk/overview.html
[16]EU legislation source: http://eur-lex.europa.eu/homepage.html
[17]UK legislation source: http://www.legislation.gov.uk/

6.2 Pretrained and Tuned Embeddings

The purpose of introducing legal word and phrase embeddings is to capture domain-specific semantics and modal verb senses. For example, *The banks can do X to allow stakeholders to ...*, here the modal verb could refer to a possibility (epistemic) or permission (deontic), hence the importance of using the contextual information in the input representation. This work experiments with count based vectors (*Glove*) trained on Wikimedia[18] and GigaWord[19]. The *word2vec* vectors are trained for the legal domain and 300*d* Google embeddings, the latter containing over 100 billion words[20]. The legal embeddings are trained on corpora containing 5861 regulatory documents, 1121 directives and 597 Acts from EU legislation[21]. The non-ANN classifiers are experimented with using feature transformations and feature selection on the original sentence embeddings since these classifiers are known to suffer from the curse of dimensionality, hence the use of ensembled approaches. In contrast, ANNs can encode the large dimensions through the hidden layers, hence we keep all dimensions intact.

6.3 A Combined Architecture

The combined architecture refers to using Bidirectional LSTMs (BiLSTMs) in an ensembled architecture that trains a model using Google News embeddings and trained legal word and phrase embeddings, shown in Figure 2. All 3 embeddings are used as input for the model, where the outputs of each are concatenated at an intermediary stage and finally passed to a 1-hidden layer ANN.

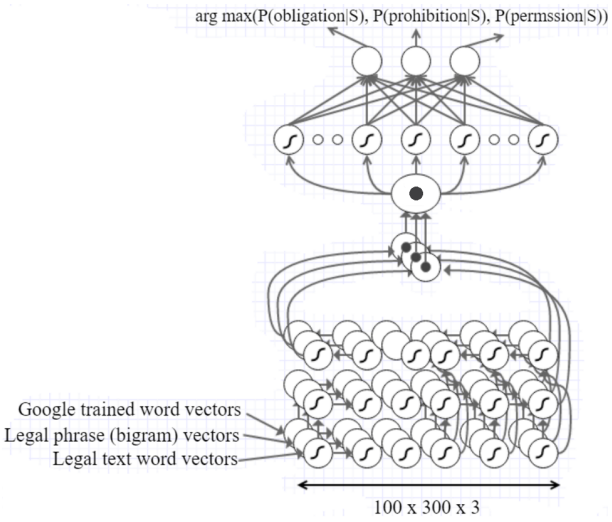

Figure 2: A Combined Deep Learning Architecture

The reason for keeping Google News pretrained embeddings separate to the legal vector embeddings is to ensure that the legal vectors are given the same weighting as the general vectors during

[18]https://dumps.wikimedia.org/enwiki/20161020/
[19]https://catalog.ldc.upenn.edu/LDC2011T07
[20]Datasets available here: https://code.google.com/archive/p/word2vec/, https://catalog.ldc.upenn.edu/LDC2011T07
[21]Retrieved from a multilingual thesaurus of the European Union (EUROVOC):http://eurovoc.europa.eu/

backpropagation in neural network learning, subsequently to the predictions. In other words, if legal texts were simply used to tune an existing pretrained word embeddings the effects would be minimal as the legal vectors are considerably smaller to the billions of terms trained by Google word vectors.

7 EXPERIMENTAL SETUP

The initial experiments compared *Glove* and *word2vec*, however *word2vec* consistently showed improvements, hence following sections proceeds with results obtained using *word2vec* input representations. We start by outlining design decision and parameters for all models used in the analysis. From this point, the classification is evaluated on numerous architectures, including a CNN, LSTM, CNN-LSTM, BiLSTM, ANN, Gradient Boosted Machines (GBM), AdaBoost, Random Forests and SVMs. Although the focus of this paper is on ANN based approaches, it is important to evaluate candidate classifiers which tend to perform well. A summary of steps taken are summarized as follows:

(1) Collect legal documents within the financial domain that contain Acts, Directives and Regulations from EU and UK legislation.
(2) Train word vectors on a large corpora to be used in conjunction with Google News embeddings.
(3) SME carries out sentence-level modality anntotations on the EU and UK legislation.
(4) Once annotation is complete, apply machine learning classifiers:
    (a) Train word2vec embeddings specifically for the legal domain and concatenate these with pretrained Google News embeddings.
    (b) Choose suitable classifiers for the task and the respective optimal parameter/hyper-parameters based on cross-validation evaluation measures. Provide feature transformations of the word embeddings for classifiers which do not deal with high dimensionality, otherwise, transformations are unnecessary and the original embeddings are used.
(5) Test all configurations of each classifier on a held out test data set and compare results and assess the probability estimates from the highest performing classifiers.

In summary, the approach taken consists of two core steps: 1) Train legal word embeddings and Google News embeddings using *word2vec*, 2) choose and evaluate ANN classifiers against non-ANN classifiers, using feature transformations and selection on non-sequential based models. Therefore, the proceeding subsections provide a summary of both steps and a description of the dataset and annotation guidelines.

7.1 Design Decisions and Parameter Settings

Prior design choices that are common to ANN and non-ANN algorithms are listed as follows:

- Sentences exceeding $|s| > N$ are truncated,where $N = 100$, as 96 % of training sentences $|s| \leq N$.
- All non-ANN classifiers use a sentence concatenation of the word embeddings, $S_j = w_i \oplus w_{i+1} ... w_N$, when $|S_j| < N$, $S_j = S_j \oplus X_{j+1:N:N}$, where $X_{j+1:N:N}$ is a vector of padded zeros.
- The $2d$ representation of sentences is reduced to a vector using simple column-wise mean, max and squared feature transformations. Feature selection is then implemented prior to classification using Information Gain (IG) for non-ANN classifier i.e Selecting only dimensions from pretrained google word vectors, legal domain word/phrase vectors that encode the most information for this specific task.

There are several unique challenges presented for modality classification, particularly in legal text. The first fundamental challenge is the lack of annotated text which is necessary for supervised learning, partially because domain experts are required for large scale supervised learning. Therefore, leveraging cheap unsupervised techniques for semantically rich input representations is a critical starting point to overcome the lack of a large annotated corpus. Secondly, the inherent imbalance of the numerous modalities expressed is undesirable for classification. It is more often than not that a particular document expresses a single (obligations and possibilities are quite common) modality disproportionately. Moreover, a minority of sentences express more than one deontic modality, hence the usage of probabilistic scores to estimate the most dominant modality.

7.1.1 Neural Network Classifiers. The baseline ANN includes two hidden layers using tanh functions with dropout, where $p = 0.2$, forming an *500-100-20-3* architecture. Similarly, the many-to-one RNN implemented also uses 2-hidden layer using tanh functions, leaky ReLU was also experimented to circumvent derivatives becoming too large or small, but this suffered in performance. The LSTM implemented outputs a fixed sequence, $a^2_{N-l:N}$, where the output length $l = 10$ and sentence length $N = 100$. This is then passed to a softmax layer to predict the modality class. This architecture and parameter setting is identical for the Bidirectional LSTM seen in fig. 1. CNNs are used with a filter size of 2 and 3 over the length of the embedding vector dimension, $d = 100$, for all embeddings except the trained google embeddings that are of dimension, $d = 300$. Both layers use a max pooling layer of size 10 before they are both concatenated, followed by a fully-connected layer before being passed to a softmax function for prediction.

7.1.2 Non-Neural Network Classifiers. C4.5 algorithm uses error pruning and restricts each leaf of the tree to having at least 2 instances. IG is also used to select features from all 3 *word2vec* combinations. The Support Vector Classifier (SVC) uses a 2^{nd} degree Radial Basis Function (RBF) with a cost, $C = 1$ and gamma $\gamma = 0.01$. Platt's Scaling is also used to produce well-calibrated probability estimates for the classifier [22]. This was not necessary for Logistic Regression as the posterior is estimated using the Negative Log Likelihood (NLL) and the Newton-Raphson solver for updating weights, naturally producing reliable posteriors. Adaptive Boosting (AdaBoost) multi-classifier (AdaBoost.M1) and Gradient Boosted Machine (GBM) both use the C4.5 decision tree as an ensemble. GBM differs from AdaBoost.M1 by applying gradient descent to find the optimal weights instead of using exponential weights to instances according to an exponential loss at each iteration. GBM

and Adaboost.M1 algorithms both use 50 decision trees, of the same parameter settings of the aforementioned C4.5 algorithm. Random Forests is also implemented in a similar fashion, sampling with replacement over sentences and terms (or word embeddings when not using hand crafted features). The max depth of the 50 trees used is restricted to a max depth of 5.

8 RESULTS

Results from the proposed classifiers shall clarify what ML classifiers perform optimally, specifically on the held out SME annotated test set, as the documents are outside of the sub-fields which the models have been trained on. Classifiers are evaluated with and without word embedding vectors to gain insight into how contextual information can improve performance and also how an ensembling of in-domain and out-of-domain predict vectors in a problem specific architecture can further improve performance.

8.1 Non-Neural Network Classifier Results

Table 1 and 2 shows the results of the non-ANN classifiers to neural network based approaches. GBM and AdaBoost, which both use an ensemble of 50 decision trees, show interesting results in comparison to SVMs which are known to deal with high dimensional data quite well. This can be attributed to both models exponentially weighting sentences which are being misclassified at each iteration. However, all classifiers struggle to correctly classify prohibitions, even considering upsampling prohibitions by augmenting available obligations with negation modifiers. This suggests that it becomes too difficult for the classifiers to find a decision boundary that effectively separates negative modalities.

Classifier	10-fold CV			Test		
	F1	**AUC**	**Acc. (%)**	**F1**	**AUC**	**Acc. (%)**
C4.5	0.85	0.93	89.13	0.62	0.75	63.32
Logistic	0.82	0.92	89.97	0.59	0.74	61.11
AdaBoost.M1	0.89	0.96	90.83	0.65	0.78	**66.57**
RF	0.87	0.95	90.10	0.63	0.76	65.06
GBM	0.90	0.97	91.04	0.67	0.79	68.41
SVM	0.83	0.92	85.38	0.61	0.74	63.12

Table 1: Evaluation of non-ANN models using baseline features such as n-grams,POS tags and normalized tf-idf scores

Results of the embedded transformations from table 2 show a significant decrease in performance compared to the handcrafted features seen in table 1. Using basic transformations on selected Google News embeddings (minimum, maximum and average), followed by IG for feature selection, has not captured the semantics within the legislative sentences. The models are not incorporating spatial or temporal dependencies in comparison to RNN and CNN architectures which is evidently a significant disadvantage.

However, some interesting results are shown with ensemble techniques, most notably using the AdaBoost.M1 classifier, however the results are not deemed to be satisfactory as the classifiers have particularly failed to perform well in distinguishing between obligations and prohibitions. Decision trees traditionally perform poorly in generic text classification problems due to the sparsity of terms

Classifier	10-fold CV			Test		
	F1	**AUC**	**Acc. (%)**	**F1**	**AUC**	**Acc. (%)**
C4.5	0.94	0.93	95.13	0.52	0.51	53.32
Logistic	0.95	0.80	91.12	0.53	0.62	54.71
AdaBoost.M1	0.87	0.79	85.83	0.53	0.61	54.56
RF	0.94	0.84	92.83	0.51	0.57	51.56
GBM	0.95	0.84	93.04	0.54	0.63	55.35
SVM	0.95	0.95	97.89	0.55	0.65	**56.12**

Table 2: Evaluation of non-ANN models employing feature transformation and IG feature selection on Google word embeddings

in the vocabulary and the algorithms inability to deal with high-dimensional data in general, considering this it is surprising that the C4.5 algorithm has performed similarly to the expessive SVM classifier. Training accuracy could suggest overfitting, however, many measures were taken to ensure this would not occur such as the use of learning curves and regularization techniques. We attribute the decline in test accuracy due to the documents being of a different section within the financial law such as the Central Bank Act 1998, Dormant Accounts 2005 and EU Consolidated accounts.

8.2 Neural Network Results

The most notable findings are that all ANN classifiers outperform those from section 8.1, although these models use different features and transformations of the original sentence level word embeddings. Both CNN and RNN based models have been able to capture the roles of modal verbs in the legal text, specifically the AML and MiFID/MiFIR related legal sub-domains as they account for a large fraction of the test sentences. From table 3, LSTM based models have shown the best results across all evaluation metrics. The CNN-LSTM architecture allows for higher level semantic representations as input to the LSTM which has improved the test accuracy, however training a model to predict from begin to end and vice-versa has shown to make a significant improvement in performance. The Bidirectional LSTM learns to remember information from both the past and the future which has evidently led to a more expressive model since it is taken the context from both sides when deciding if a sentence is within a deontic modality class.

Classifier	10-fold CV			Test		
	F1	**AUC**	**Acc. (%)**	**F1**	**AUC**	**Acc. (%)**
ANN	0.89	0.93	91.42	0.56	0.61	58.71
CNN	0.92	0.95	90.02	0.68	0.81	71.40
LSTM	0.94	0.96	93.39	0.72	0.85	74.22
CNN-LSTM	0.92	0.97	95.17	0.73	0.86	75.93
BiLSTM	0.93	0.97	94.89	0.76	0.89	**78.56**

Table 3: Evaluation results on gold standard using only Google News word embeddings

Classifier	10-fold CV			Test		
	F1	**AUC**	**Acc (%)**	**F1**	**AUC**	**Acc. (%)**
ANN	0.79	0.91	83.49	0.65	0.81	68.63
CNN	0.85	0.92	89.16	0.68	0.85	71.45
LSTM	0.89	0.96	93.23	0.75	0.89	79.52
CNN-LSTM	0.92	0.97	95.17	0.73	0.86	78.71
BiLSTM	0.91	0.97	94.80	0.77	0.91	81.20
BiLSTMnew	0.95	0.98	96.51	0.79	0.93	**82.33**

Table 4: Evaluation results on gold standard using Google News word embeddings and legal word and phrase embeddings

Table 4 shows that an ensemble of domain-specific embeddings along with general word representations in a merged BiLSTM architecture has slightly improved over exclusively using Google News embeddings, highlighting the importance of incorporating domain specific embeddings. In contrast, applying normative rules for each sub-domain to allow for contextual information in identical or similar application is very invasive, and almost not feasible given the rapid change in legislation in recent years. Furthermore mean 10 cross fold training accuracy for all classifiers in table 3 and 4 is 92.54 % is quite high, which suggests that provided more data test accuracy would rise significantly.

The architecture denoted *BiLSTMnew* is that shown from figure 2, which uniquely incorporates the embeddings in a combined architecture. Here, we have shown the benefits of combining architectures which incorporate domain-specific semantic representations. The results show promise since the importance of particular types of deontic modalities is directly driven by the SME annotations and interpretation of the legal norms within the legislation, in comparison to deontic logic which does not provide the precise use of modality within a given context.

Table 5 presents the confusion matrix for the 3 deontic classes of interest. A primary consideration before the experiments was that negation modifiers that distinguish prohibitions from obligations would be difficult to account for. Although the majority of incorrect classifications on the gold standard test set are between prohibitions and obligations, the results are promising considering that AML, Financial Instrumentation, Bank Acts and Consolidated Accounts and Market based document sentences are left completely separate for testing. LSTMs have identified the negation and modal verbs within the legislation to some degree, meaning they are accounting for broad context but also flexible enough to account for single terms with high influence. In comparison to the traditional approach of incorporating deontic logic in rule based systems, this approach is flexible enough to learn both high influence terms that can also provide context.

	10-CV prediction			Test prediction		
Obligation	675	1	38	235	95	9
Prohibition	0	456	0	66	311	9
Permission	76	1	801	14	9	41

Table 5: Confusion Matrix for BiLSTMnew Classifier

Similar to previously mentioned work [8] there is a slight drop off in performance when testing on documents outside of the sub-domains on which they were trained on. Although, ANN classifiers have shown the capability to maintain reasonable performance on a new set of documents, which suggests this could be replicated for other domains in law. Post analysis suggested that sentences which were longer than the mean length of sentences in the training set were more likely to be misclassified, particularly between prohibitions and obligations as the influence of a single negation becomes more difficult to detect.

9 CONCLUSIONS

This paper has presented a deep learning architecture for classifying deontic modalities in regulatory texts, providing a comparison to other competitive algorithms ranging from ensemble-based decision tree classifiers to largest margin classifiers. ANN classifiers have shown a consistent improvement over other classifiers, specifically LSTM based networks. Data augmentation for class balancing has improved test set results by including modifiers after modal verbs in the minority class i.e. Upsampling the prohibition class. The input representations of terms and phrases using an ensemble of *word2vec* embeddings have also shown an improvement over using the embeddings independently, and similarly over the non-ANN classifiers. Probability estimates have been used to gauge the preciseness of a deontic statement in a given law context, estimates are high in a non-ambiguous context and inversely probability estimates for deontic classes are when they overlap. This can allow lawyers to preference well defined deontic statements in legislation, and analyze those which overlap with different legal norms. The practical benefits is that it allows for the detection of legal norms and also to detect regulatory change. In summary, this paper has viewed the problem of extracting deontic modalities within a given context as probabilistic instead of logical and has shown success on a held out test set. Given the constant change in regulation, acts and directives, the likelihood of misusing modal verbs, contradicting prior legislation and lack of contextual awareness would be expected to increase given the demands on legal practitioners. Automated systems that can account for these problems will need to be considered in the future and we believe the presented classification approaches will be will play a significant role in such systems.

10 FUTURE WORK

To extend this body of work, we will expand the existing to work for alethic modalities such as possibility, impossibility and necessity. Moreover, encompassing a larger set of domains in law will allow for a comparison of domain specific usages of both deontic and epistemic modality. Probability estimates can also be used to identify change across sub-domains by analyzing the change in probability estimates given contextual information and amended legislation, which is of particular importance in regulatory change management (RCM).

REFERENCES

[1] Kartik Asooja, Georgeta Bordea, Gabriela Vulcu, Leona O'Brien, Angelina Espinoza, Elie Abi-Lahoud, Paul Buitelaar, and Tom Butler. Semantic Annotation of Finance Regulatory Text using Multilabel Classification. (????).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Kathryn Baker, Michael Bloodgood, Bonnie J Dorr, Chris Callison-Burch, Nathaniel W Filardo, Christine Piatko, Lori Levin, and Scott Miller. 2012. Modality and negation in simt use of modality and negation in semantically-informed syntactic mt. *Computational Linguistics* 38, 2 (2012), 411–438.

[4] Kathryn Baker, Michael Bloodgood, Bonnie J Dorr, Nathaniel W Filardo, Lori Levin, and Christine Piatko. 2014. A modality lexicon and its use in automatic tagging. *arXiv preprint arXiv:1410.4868* (2014).

[5] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors.. In *ACL (1)*. 238–247.

[6] Brian F Chellas. 1980. *Modal logic: an introduction*. Cambridge university press.

[7] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[8] Emile de Maat, Kai Krabben, and Radboud Winkels. 2010. Machine Learning versus Knowledge Based Classification of Legal Texts.. In *JURIX*. 87–96.

[9] Thomas F Gordon, Guido Governatori, and Antonino Rotolo. 2009. Rules and norms: Requirements for rule interchange languages in the legal domain. In *International Workshop on Rules and Rule Markup Languages for the Semantic Web*. Springer, 282–296.

[10] Zellig S Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[11] Risto Hilpinen. 2012. *Deontic logic: Introductory and systematic readings*. Vol. 33. Springer Science & Business Media.

[12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97.

[13] Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98* (1998), 137–142.

[14] Thorsten Joachims. 2002. *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers.

[15] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[17] Ana Marasović and Anette Frank. 2016. Multilingual Modal Sense Classification using a Convolutional Neural Network. *arXiv preprint arXiv:1608.05243* (2016).

[18] Ana Marasović, Mengfei Zou, Alexis Palmer, and Anette Frank. 2016. Modal Sense Classification At Large. Paraphrase-Driven Sense Projection, Semantically Enriched Classification Models and Cross-Genre Evaluations. *LiLT (Linguistic Issues in Language Technology)* 14 (2016).

[19] L Thorne McCarty. 1983. Permissions and obligations. In *IJCAI*, Vol. 83. Citeseer, 287–294.

[20] John-Jules Ch Meyer et al. 1988. A different approach to deontic logic: deontic logic viewed as a variant of dynamic logic. *Notre dame journal of formal logic* 29, 1 (1988), 109–136.

[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[22] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* 10, 3 (1999), 61–74.

[23] J Ross Quinlan et al. 1996. Bagging, boosting, and C4. 5. In *AAAI/IAAI, Vol. 1*. 725–730.

[24] Lamber Royakkers. 2013. *Extending deontic logic for the formalisation of legal rules*. Vol. 36. Springer Science & Business Media.

[25] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[26] Harry Surden. 2014. Machine learning and law. (2014).

[27] Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* (2015).

[28] Jean-Christophe Verstraete. 2005. Scalar quantity implicatures and the interpretation of modality: Problems in the deontic domain. *Journal of pragmatics* 37, 9 (2005), 1401–1418.

[29] G. H. von Wright. 1951. Deontic Logic. *Mind* 60, 237 (1951), 1–15. http://www.jstor.org/stable/2251395