Obligation and Prohibition Extraction Using Hierarchical RNNs

Ilias Chalkidis[1,2], Ion Androutsopoulos[1], and Achilleas Michos[2]

[1]Department of Informatics, Athens University of Economics and Business, Greece
[2]Cognitiv+ Ltd., London, UK

Abstract

We consider the task of detecting contractual obligations and prohibitions. We show that a self-attention mechanism improves the performance of a BILSTM classifier, the previous state of the art for this task, by allowing it to focus on indicative tokens. We also introduce a hierarchical BILSTM, which converts each sentence to an embedding, and processes the sentence embeddings to classify each sentence. Apart from being faster to train, the hierarchical BILSTM outperforms the flat one, even when the latter considers surrounding sentences, because the hierarchical model has a broader discourse view.

1 Introduction

Legal text processing (Ashley, 2017) is a growing research area, comprising tasks such as legal question answering (Kim and Goebel, 2017), contract element extraction (Chalkidis et al., 2017), and legal text generation (Alschnerd and Skougarevskiy, 2017). We consider *obligation* and *prohibition extraction* from contracts, i.e., detecting sentences (or clauses) that specify what *should* or should *not* happen (Table 1). This task is important for legal firms and legal departments, especially when they process large numbers of contracts to monitor the compliance of each party. Methods that would automatically identify (e.g., highlight) sentences (or clauses) specifying obligations and prohibitions would allow lawyers and paralegals to inspect contracts more quickly. They would also be a step towards populating databases with information extracted from contracts, along with methods that extract contractors, particular dates (e.g., start and end dates), applicable law, legislation references etc. (Chalkidis and Androutsopoulos, 2017).

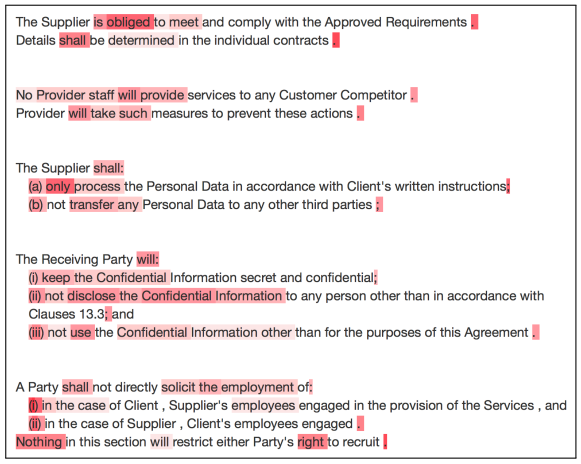

Figure 1: Heatmap visualizing the attention scores of BILSTM-ATT for some examples of Table 1.

Obligation and prohibition extraction is a kind of *deontic* sentence (or clause) classification (O'Neill et al., 2017). Different firms may use different or finer deontic classes (e.g., distinguishing between payment and delivery obligations), but obligations and prohibitions are the most common coarse deontic classes. Using similar classes, O' Neill et al. (2017) reported that a bidirectional LSTM (BILSTM) classifier (Graves et al., 2013) outperformed several others (including logistic regression, SVM, AdaBoost, Random Forests) in legal sentence classification, possibly because long-term dependencies (e.g., modal verbs or negations interacting with distant dependents) are common and crucial in legal texts, and LSTMs can cope with long-term dependencies better than methods relying on fixed-size context windows.

We improve upon the work of O' Neill et al. (2017) in four ways. First, we show that self-attention (Yang et al., 2016) improves the performance of the BILSTM classifier, by allowing the system to focus on indicative words (Fig 1). Second, we introduce a hierarchical BILSTM, where a first BILSTM processes each sentence word by

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 254–259
Melbourne, Australia, July 15 - 20, 2018.

No.	Gold Class	Sentences/Clauses
1	Obligation	The Supplier ***is obliged to*** meet and comply with the Approved Requirements.
	None	Details shall be determined in the individual contracts.
2	Prohibition	***No*** Provider staff ***will*** provide services to any Customer Competitor.
	Obligation	Provider ***will*** take such measures to prevent these actions.
3	Prohibition	Provider ***is not entitled*** to suspend this Agreement prior to the lapse of the fifth year.
4	Oblig./Prohib. List Intro	The Supplier ***shall***:
	Obligation List Item	***(a)*** only process the Personal Data in accordance with Client's written instructions;
	Prohibition List Item	***(b) not*** transfer any Personal Data to any other third parties;
5	Oblig./Prohib. List Intro	The Receiving Party ***will***:
	Obligation List Item	***(i)*** keep the Confidential Information secret and confidential;
	Prohibition List Item	***(ii) not*** disclose the Confidential Information to any person other than in accordance with Clauses 13.3; and
	Prohibition List Item	***(iii) not*** use the Confidential Information other than for the purposes of this Agreement.
6	Oblig./Prohib. List Intro	A Party ***shall not*** directly solicit the employment of:
	Prohibition List Item	***(i)*** in the case of Client, Supplier's employees engaged in the provision of the Services,
	Prohibition List Item	***(ii)*** in the case of Supplier, Client's employees engaged.
	None	Nothing in this section will restrict either Party's right to recruit.

Table 1: Examples of sentences and clauses, with human annotations of classes. Terms that are highly indicative of the classes are shown in bold and underlined here, but are not marked by the annotators.

Gold Class	Train	Dev	Test
None	15,401	3,905	4,141
Obligation	11,005	2,860	970
Prohibition	1,172	314	108
Obligation List Intro	828	203	70
Obligation List Item	2888	726	255
Prohibition List Item	251	28	19
Total	**31,545**	**8,036**	**5,563**

Table 2: Sentences/clauses after sentence splitting.

word producing a sentence embedding, and a second BILSTM processes the sentence embeddings to classify each sentence. The hierarchical BILSTM is similar to Yang et al.'s (2016), but classifies sentences, not entire texts (e.g., news articles or product reviews). It outperforms a flat BILSTM that classifies each sentence independently, even when the latter considers neighbouring sentences, because the hierarchical BILSTM has a broader view of the discourse. Third, we experiment with a dataset an order of magnitude larger than the dataset of O' Neill et al. Fourth, we introduce finer classes (Tables 1–2), which fit better the target task, where nested clauses are frequent.

2 Data

We experimented with a dataset containing 6,385 training, 1,595 development, and 1,420 test sections (articles) from the main bodies (excluding introductions, covers, recitals) of 100 randomly selected English service agreements.[1] The sections were preprocessed by a sentence splitter, which in clause lists (Examples 4–6 in Table 1) treats the introductory clause and each nested clause as separate sentences, since each nested clause may belong in a different class.[2]

The splitter produced 31,545 training, 8,036 development, and 5,563 test sentences/clauses.[3] Table 2 shows their distribution in the six gold (correct) classes. Each section was annotated by a single law student (5 students in total). All the annotations were checked and corrected by a single paralegal expert, who produces annotations of this kind on a daily basis, based on strict guidelines of the firm that provided the data.

We used pre-trained 200-dimensional word embeddings and pre-trained 25-dimensional POS tag embeddings, obtained by applying WORD2VEC (Mikolov et al., 2013) to approx. 750k and 50k English contracts, respectively, as in our previous work (Chalkidis et al., 2017). We also pre-trained 5-dimensional token shape embeddings (e.g., all capitals, first letter capital, all digits), obtained as in our previous work (Chalkidis and Androutsopoulos, 2017). Each token is represented by the concatenation of its word, POS, shape embeddings (Fig. 2, bottom). Unknown tokens are mapped to

[1] The splitting of the dataset into training, development, and test subsets was performed by first agglomeratively clustering all sections (articles) based on Levenshtein distance, and then assigning entire clusters to the training, development, or test subset, to avoid having similar sections (e.g., based on boilerplate clauses) in different subsets.

[2] We use NLTK's splitter (http://www.nltk.org/), with additional post-processing based on regular expressions.

[3] There are at most 15 sentences/clauses per section in the training set. We hope to make the dataset, or a similar anonymized one, publicly available in the near future, but the dataset is currently not available due to confidentiality issues.

pre-trained POS-specific 'unk' embeddings (e.g., 'unk-n', 'unk-vb'). The dataset of Table 2 has no overlap with the corpus of contracts that was used to pre-train the embeddings.

3 Methods

BILSTM: The first classifier we considered processes a single sentence (or clause) at a time. It feeds the concatenated word, POS, shape embeddings ($e_1, \ldots, e_n \in \mathbb{R}^{230}$) of the tokens $w_1, w_2, \ldots, w_n$ of the sentence to a forward LSTM, and (in reverse order) to a backward LSTM, obtaining the forward and backward hidden states ($\overrightarrow{h}_1, \ldots, \overrightarrow{h}_n \in \mathbb{R}^{300}$ and $\overleftarrow{h}_1, \ldots \overleftarrow{h}_n \in \mathbb{R}^{300}$). The concatenation of the last states ($h = [\overrightarrow{h}_n; \overleftarrow{h}_1]$) is fed to a multinomial Logistic Regression (LR) layer, which produces a probability per class.

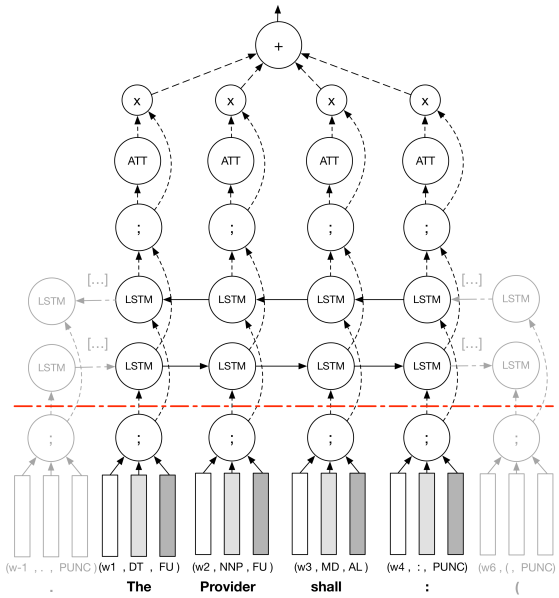

Figure 2: BILSTM with self-attention (ATT nodes) used on its own (BILSTM-ATT) or as the sentence encoder of the hierarchical BILSTM (H-BILSTM-ATT, Fig. 3). In X-BILSTM-ATT, the two LSTM chains also consider the words of surrounding sentences. The red dashed line is a drop-out layer.

BILSTM-ATT: When self-attention is added (Fig. 2), the sentence (or clause) is represented by the weighted sum (h) of the hidden states ($h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t] \in \mathbb{R}^{600}$) of the BILSTM, where $a_1, \ldots, a_n \in \mathbb{R}$ are attention scores, $v \in \mathbb{R}^{600}$, $b \in \mathbb{R}$:

$$h = a_1 h_1 + \cdots + a_t h_t + \cdots + a_n h_n \quad (1)$$

$$a'_t = \tanh(v^T h_t + b) \quad (2)$$

$$a_t = \text{softmax}(a'_t; a'_1, \ldots, a'_n) \quad (3)$$

Again, h is then fed to a multinomial LR layer. Figure 1 visualizes the attention scores ($a_1, \ldots, a_n$) of BILSTM-ATT when reading some of the sentences (or clauses) of Table 1. The attention scores are higher for modals, negations, words that indicate obligations or prohibitions (e.g., 'obliged', 'only'), and tokens indicating nested clauses (e.g., '(a)', ':', ';'), which allows BILSTM-ATT to focus more on these tokens (the corresponding states) when computing the sentence representation (h).

X-BILSTM-ATT: In an extension of BILSTM-ATT, called X-BILSTM-ATT, the BILSTM chain is fed with the token embeddings (e_t) not only of the sentence being classified, but also of the previous (and following) tokens (faded parts of Fig. 2), up to 150 previous (and 150 following) tokens, 150 being the maximum sentence length in the dataset.[4] This might allow the BILSTM chain to 'remember' key parts of the surrounding sentences (e.g., a previous clause ending with 'shall not:') when producing the context-aware embeddings (states h_t) of the current sentence. The self-attention mechanism still considers the states (h_t) of the tokens of the current sentence only, and the sentence representation (h) is still computed as in Eq. 1.

H-BILSTM-ATT: The hierarchical BILSTM classifier, H-BILSTM-ATT, considers all the sentences (or clauses) of an entire section. Each sentence (or clause) is first turned into a sentence embedding ($h \in \mathbb{R}^{600}$), as in BILSTM-ATT (Fig. 2). The sequence of sentence embeddings is then fed to a second BILSTM (Fig. 3), whose hidden states ($h_t^{(2)} = [\overrightarrow{h}_t^{(2)}; \overleftarrow{h}_t^{(2)}] \in \mathbb{R}^{600}$) are treated as context-aware sentence embeddings. The latter are passed on to a multinomial LR layer, producing a probability per class, for each sentence (or clause) of the section. We hypothesized that H-BILSTM-ATT would perform better, because it considers an entire section at a time, and salient information about a sentence or clause (e.g., that the opening clause of a list contains a negation or modal) can be 'condensed' in its sentence embedding and interact with the sentence embeddings of distant sentences or clauses (e.g., a nested clause several clauses after the opening one) in the upper BILSTM (Fig. 3).

[4] Memory constraints did not allow including more tokens. We used a single NVIDIA 1080 GPU. All methods were implemented using KERAS (https://keras.io/) with a TENSORFLOW backend (https://www.tensorflow.org/). We padded each sentence to the maximum length.

	BILSTM				BILSTM-ATT				X-BILSTM-ATT				H-BILSTM-ATT			
Gold Class	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC	P	R	F1	AUC
None	0.95	0.91	0.93	0.98	0.97	0.90	0.93	**0.99**	0.96	0.90	0.93	0.98	**0.98**	**0.96**	**0.97**	**0.99**
Obligation	0.75	0.85	0.79	0.86	0.75	0.88	0.81	0.86	0.75	0.87	0.81	0.88	**0.87**	**0.92**	**0.90**	**0.96**
Prohibition	0.67	0.62	0.64	0.75	0.74	0.75	0.74	0.80	0.65	0.75	0.70	0.74	**0.84**	**0.83**	**0.84**	**0.90**
Obl. List Begin	0.70	0.86	0.77	0.81	0.71	0.85	0.77	0.83	0.72	0.75	0.74	0.80	**0.90**	**0.89**	**0.89**	**0.93**
Obl. List Item	0.53	0.66	0.59	0.64	0.48	0.70	0.57	0.60	0.49	0.78	0.60	0.66	**0.85**	**0.94**	**0.89**	**0.94**
Proh. List Item	0.59	0.35	0.43	0.50	0.61	0.55	0.59	0.62	**0.83**	0.50	0.62	0.67	0.80	**0.84**	**0.82**	**0.92**
Macro-average	0.70	0.70	0.70	0.74	0.73	0.78	0.74	0.78	0.73	0.76	0.73	0.79	**0.87**	**0.90**	**0.89**	**0.94**
Micro-average	0.90	0.88	0.88	0.94	0.90	0.88	0.89	0.96	0.90	0.88	0.89	0.94	**0.95**	**0.95**	**0.95**	**0.98**

Table 3: Precision, recall, F1, and AUC scores, with the best results in bold and gray background.

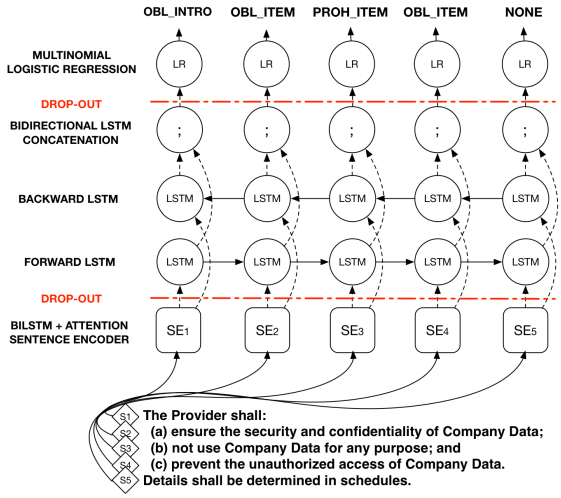

Figure 3: Upper part of the hierarchical BILSTM (H-BILSTM-ATT). The sentence embeddings (SE_i) are generated by the encoder of Fig. 2.

4 Experimental Results

Hyper-parameters were tuned by grid-searching the following sets, and selecting the values with the best validation loss: LSTM hidden units {100, 200, 300}, batch size {8, 16, 32}, drop-out rate {0.4, 0.5, 0.6}. The red dashed lines of Fig. 2–3 are drop-out layers.[5] We used categorical cross-entropy loss, Glorot initialization (Glorot and Bengio, 2010), Adam (Kingma and Ba, 2015), learning rate 0.001, and early stopping on the validation loss. Table 3 reports the precision, recall, F1 score, area under the precision-recall curve (AUC) per class, as well as micro- and macro-averages.

The self-attention mechanism (BILSTM-ATT) leads to clear overall improvements (in macro and micro F1 and AUC, Table 3) comparing to the plain BILSTM, supporting the hypothesis that self-attention allows the classifier to focus on indicative tokens. Allowing the BILSTM to consider tokens of neighboring sentences (X-BILSTM-ATT) does not lead to any clear overall improvements.

[5] We resample the drop-out mask at each time-step.

The hierarchical H-BILSTM-ATT clearly outperforms the other three methods, supporting the hypothesis that considering entire sections and allowing the sentence embeddings to interact in the upper BILSTM (Fig. 3) is beneficial.

Notice that the three flat methods (BILSTM, BILSTM-ATT, X-BILSTM-ATT) obtain particularly lower F1 and AUC scores, compared to H-BILSTM-ATT, in the classes that correspond to nested clauses (obligation list item, prohibition list item). This is due to the fact that the flat methods have no (or only limited, in the case of X-BILSTM-ATT) view of the previous sentences, which often indicate if a nested clause is an obligation or prohibition (see, for example, examples 4–6 in Table 1).

H-BILSTM-ATT is also much faster to train than BILSTM and BILSTM-ATT (Table 4), even though it has more parameters, because it converges faster (5-7 epochs vs. 12-15). X-BILSTM-ATT is particularly slow, because its BILSTM processes the same sentences multiple times, when they are classified and when they are neighboring sentences.

Network	Training Time	Parameters
BILSTM	5h 30m	1,278M
BILSTM-ATT	8h 30m	1,279M
X-BILSTM-ATT	25h 40m	1,279M
H-BILSTM-ATT	2h 30m	1,837M

Table 4: Training times and parameters to learn.

5 Related Work

As already noted, we built upon the work of O'Neill et al. (2017). The dataset of O'Neill et al. contained financial legislation, not contracts, and was an order of magnitude smaller (obligations, prohibitions, permissions had 1,297 training, 622 test sentences in total, cf. Table 2), but also included permissions, which we did not consider.

Waltl et al. (2017) classified statements from German tenancy law into 22 classes (including prohibition, permission, consequence), using active learning with Naive Bayes, LR, MLP classifiers, experimenting with 504 sentences.

Kiyavitskaya et al. (2008) used grammars, word lists, and heuristics to extract rights, obligations, exceptions, and other constraints from US and Italian regulations.

Asooja et al. (2015) employed SVMs with *n*-gram and manually crafted features to classify paragraphs of money laundering regulations into five classes (e.g., enforcement, monitoring, reporting), experimenting with 212 paragraphs.

In previous work (Chalkidis et al., 2017; Chalkidis and Androutsopoulos, 2017) we focused on extracting contract elements (e.g., contractor names, legislation references, start and end dates, amounts), a task which is similar to named entity recognition. The best results were obtained by stacked BILSTMs (Irsoy and Cardie, 2014) or stacked BILSTM-CRF models (Ma and Hovy, 2016); hierarchical BILSTMs were not considered. By contrast, in this paper we considered obligation and prohibition extraction, treating it as a sentence (or clause) classification task, and showing the benefits of employing a hierarchical BILSTM model that considers both the sequence of words in each sentence and the sequence of sentences.

Yang et al. (2016) proposed a hierarchical RNN with self-attention to classify texts. A first bidirectional RNN turns the words of each sentence to a sentence embedding, and a second one turns the sentence embeddings to a document embedding, which is fed to an LR layer. Yang et al. use self-attention in both RNNs, to assign attention scores to words and sentences. We classify sentences (or clauses), not entire texts, hence our second BILSTM does not produce a document embedding and does not use self-attention. Also, Yang et al. experimented with reviews and community question answering logs, whereas we considered legal texts.

Hierarchical RNNs have also been developed for multilingual text classification (Pappas and Popescu-Belis, 2017), language modeling (Lin et al., 2015), and dialogue breakdown detection (Xie and Ling, 2017).

6 Conclusions and Future Work

We presented the legal text analytics task of detecting contractual obligations and prohibitions. We showed that self-attention improves the performance of a BILSTM classifier, the previous state of the art in this task, by allowing the BILSTM to focus on indicative tokens. We also introduced a hierarchical BILSTM (also using attention), which converts each sentence to an embedding, and then processes the sentence embeddings to classify each sentence. Apart from being faster to train, the hierarchical BILSTM outperforms the flat one, even when the latter considers the surrounding sentences, because the hierarchical model has a broader view of the discourse.

Further performance improvements may be possible by considering deeper self-attention mechanisms (Pavlopoulos et al., 2017), stacking BILSTMs (Irsoy and Cardie, 2014), or pre-training the BILSTMs with auxiliary tasks (Ramachandran et al., 2017). The hierarchical BILSTM with attention of this paper may also be useful in other sentence, clause, or utterance classification tasks, for example in dialogue turn classification (Xie and Ling, 2017), detecting abusive user comments in on-line discussions (Pavlopoulos et al., 2017), and discourse segmentation (Hearst, 1997). We would also like to investigate replacing its BILSTMs with sequence-labeling CNNs (Bai et al., 2018), which may lead to efficiency improvements.

Acknowledgments

We are grateful to the members of AUEB's Natural Language Processing Group, for several suggestions that helped significantly improve this paper.

References

W. Alschnerd and D. Skougarevskiy. 2017. Towards an automated production of legal texts using recurrent neural networks. In *Proceeding of the 16th International Conference on Artificial Intelligence and Law*, pages 159–168, London, UK.

K.D. Ashley. 2017. *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press.

K. Asooja, G. Bordea, G. Vulcu, L. O'Brien, A. Espinoza, E. Abi-Lahoud, P. Buitelaar, and T. Butler. 2015. Semantic annotation of finance regulatory text using multilabel classification. In *Proceedings of the International Workshop on Legal Domain and Semantic Web Applications*, Portoroz, Slovenia.

S. Bai, J.Z. Kolter, and V. Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271.

I. Chalkidis and I. Androutsopoulos. 2017. A deep learning approach to contract element extraction. In *Proceedings of the 30th International Conference on Legal Knowledge and Information Systems*, pages 155–164, Luxembourg.

I. Chalkidis, I. Androutsopoulos, and A. Michos. 2017. Extracting contract elements. In *Proceedings of the 16th International Conference on Artificial Intelligence and Law*, pages 19–28, London, UK.

X. Glorot and Y. Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 249–256, Sardinia, Italy.

A. Graves, N. Jaitly, and A. Mohamed. 2013. Hybrid speech recognition with deep bidirectional LSTM. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 273–278, Olomouc, Czech Republic.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

O. Irsoy and C. Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pages 2096–2104, Montreal, Canada.

M.Y. Kim and R. Goebel. 2017. Two-step cascaded textual entailment for legal bar exam question answering. In *Proceedings of the 4th Competition on Legal Information Extraction/Entailment*, London, UK.

D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 5th International Conference on Learning Representations*, San Diego, CA, USA.

N. Kiyavitskaya, N. Zeni, Travis D. Breaux, Annie I. Antón, James R. Cordy, L. Mich, and J. Mylopoulos. 2008. Automating the extraction of rights and obligations for regulatory compliance. In *Proceedings of the 27th International Conference on Conceptual Modeling*, pages 154–168, Barcelona, Spain.

R. Lin, S. Liu, M. Yang, M. Li, M. Zhou, and S. Li. 2015. Hierarchical recurrent neural network for document modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 899–907, Lisbon, Portugal.

X. Ma and E. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1064–1074, Berlin, Germany.

T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Stateline, NV.

J. O'Neill, P. Buitelaar, C. Robin, and L. O' Brien. 2017. Classifying Sentential Modality in Legal Language: A Use Case in Financial Regulations, Acts and Directives. In *Proceedings of the 16th International Conference on Artificial Intelligence and Law*, pages 159–168, London, UK.

N. Pappas and A. Popescu-Belis. 2017. Multilingual hierarchical attention networks for document classification. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, Tapei, Taiwan.

J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos. 2017. Deeper attention to abusive user content moderation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark.

P. Ramachandran, P.r J. Liu, and Q. V. Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark.

B. Waltl, J. Muhr, I. Glaser, G. Bonczek, E. Scepankova, and F. Matthes. 2017. Classifying legal norms with active machine learning. In *Proceedings of the 30th International Conference on Legal Knowledge and Information Systems*, pages 11–20, Luxembourg City Luxembourg.

Z. Xie and G. Ling. 2017. Dialogue breakdown detection using hierarchical bi-directional LSTMs. In *Proceedings of the 6th Dialog System Technology Challenges (Track 3: Dialog Breakdown Detection)*, Long Beach, USA.

Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, CA, USA.