Artificial Intelligence and Law (2019) 27:171–198
https://doi.org/10.1007/s10506-018-9238-9

Deep learning in law: early adaptation and legal word embeddings trained on large corpora

Ilias Chalkidis[1] · **Dimitrios Kampas**[2]

Published online: 11 December 2018

Abstract
Deep Learning has been widely used for tackling challenging natural language processing tasks over the recent years. Similarly, the application of Deep Neural Networks in legal analytics has increased significantly. In this survey, we study the early adaptation of Deep Learning in legal analytics focusing on three main fields; text classification, information extraction, and information retrieval. We focus on the semantic feature representations, a key instrument for the successful application of deep learning in natural language processing. Additionally, we share pre-trained legal word embeddings using the WORD2VEC model over large corpora, comprised legislations from UK, EU, Canada, Australia, USA, and Japan among others.

Keywords Natural language processing · Deep learning · Legal word vectors

1 Introduction

Recently, Deep Learning (Goodfellow et al. 2016; Goldberg 2017) has gained significant attention in the natural language processing research community as a promising family of techniques dealing with the complexity of human language. Deep Neural Networks have been rapidly replacing rule-based approaches, dictionary-based models and traditional machine learning techniques, which in their majority require intensive manual feature engineering. The reason lies in the fact of their poor performance (O'Neill et al. 2017; Do et al. 2017) when dealing with the words polysemy, synonyms and the multiplicity of the way humans imprint and structure text. The above-said techniques fall short in capturing language

✉ Ilias Chalkidis
ihalk@aueb.gr

Dimitrios Kampas
dimitrios.kampas@list.lu

[1] Department of Informatics, Athens University of Economics and Business, Athens, Greece

[2] IT for Innovation Services, Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg

semantics and complicated linguistic structures along with their long-distance relationships, as humans do. On the other hand, Neural Networks incorporate many interesting features such as multi-layering, non-linear activation functions and grasp long-term dependencies. The sophisticated recursive and convolutional neural architectures better capture the sequential structure of natural language. Relying on the multi-layer architecture, Deep Neural Networks have extended their analytical and processing capacity to capture subtle language semantics and syntax; closer to human sophistication.

Deep Neural Networks have also been gradually used in the legal domain. Traditionally, researchers incorporated manually crafted knowledge bases and patterns to capture legal concepts, terms of interest and synonyms that were defined beforehand. Many works relied on the structure of the legal documents to be able to segment and process text. Nevertheless, the presumed structure was not consistent across different laws and legislations. The language is heterogeneous, the legal concepts evolve and the maintenance of legal knowledge bases is tedious and expensive (Do et al. 2017). Moreover, ontological representations that model legal knowledge as a whole are not sufficiently good to achieve sharp results as they are too generic. Similarly, taxonomies and information clouds crafted to fit a particular task may be outdated rapidly.

Our primary focus is to explore the word feature representations that are commonly used to feed Neural Networks or capture semantic similarities among text snippets. We observed that the majority of researchers utilized generic purpose word feature representations, or in some cases, domain-specific word embeddings, which have been pre-trained over small legal datasets. To the best of our knowledge, there are no publicly available word embeddings trained on large legal corpora. In Sect. 2 of this article, we present and share for public use legal word embeddings that were trained over a big collection of legal documents by using the WORD2VEC model. Further on, in Sect. 3, we discuss a representative selection of the most notable recent articles that incorporate Deep Learning methods, oriented to address different tasks on legal corpora. Our review spotlights the feature representations, the Deep Neural Networks architectures and important observations derived from the results. We classify the literature reviews, as follows:

1. Text Classification: In this category, we present some representative works relevant with to the categorization of textual units, such as sentences, paragraphs, sections or even long documents.
2. Information Extraction: In this category, we review characteristic articles related to sequence labeling (tagging) tasks, such as chunking and named-entity recognition.
3. Information Retrieval: In this category, we sort works that tackle the problem of retrieving articles of interest out of a collection of legal documents or articles that entail a query. We review the Deep Neural Networks that implement legal question-answering system components under the Competition on Legal Information Extraction/Entailment (COLIEE).

Since the scope of our survey is to scrutinize and highlight the contribution of Deep Learning on legal text analytics, we do not discuss in detail linear models and feature extraction techniques, commonly used in Natural Language Processing, like TF–IDF scores that are possibly presented in the respective articles.

2 Word embeddings: current trends in text preprocessing

2.1 Technical characteristics and implementations

One of the most important aspects of the rapid growth of Deep Learning in NLP was the introduction of word embeddings. Word embeddings are low-dimensional dense vectors employed as word feature representations. This encapsulates two significant differences in contrast to the traditional word representations: Dense versus sparse and low-dimensional versus high-dimensional.

Sparse vectors, such as binary (one-hot), or TF–IDF vectors, are used to represent words, usually in a high dimensional space. The sparsity leads to poor semantic and syntactic representation of words. Therefore, words that humans consider semantically close (e.g., felony, crime) end up being completely irrelevant. Hand-crafted features (e.g., dictionaries, regular expressions/patterns) or scoring schemes (e.g., TF–IDF) may improve the outcome but not alleviate the drawback of the sparse representation. On the contrary, dense vectors represent words based on features that implicitly incorporate semantic and syntactic information, rather than the word itself. The empirical results showed semantic correlations close to human perception. This affects significantly the generalization capability of any trainable model. The dimensionality is the other important factor. Researchers tended to use high-dimensional feature representations with thousands of features to describe a fixed vocabulary based on the most frequent or statistically significant words to a given task. This strategy increases the computational complexity and cost for the selected algorithms to train on these representations.

Learning thorough word embeddings with an embedding layer from scratch by using annotated datasets tailored to a task of interest, may be challenging due to the insufficient vocabulary size of the training dataset. This fact may lead to poor generalization performance. In other words, we expect that many words are underrepresented in the annotated datasets, which affects the decision-making effectiveness of the trained model whenever the underrepresented words are met.

The last quinquennium, there has been great research on building algorithms that would learn word embeddings in an unsupervised fashion. The most popular among them are WORD2VEC, GLOVE and FASTTEXT. The main assumption behind the proposed algorithms is that similar words tend to co-occur in similar contexts, which was first stated in the famous quote "You shall know a word by the company it keeps" by Firth (1935). Based on the former assumption, word embeddings models utilize large corpora to build datasets that train their algorithms.

Mikolov et al. (2013) introduced the WORD2VEC model, which described two different algorithms, skip-gram and Continuous Bag of Words (CBOW). Both algorithms use sliding windows to form word pairs that are present in the same window.

In most implementations, both algorithms are realized as shallow Neural Networks with two fully-connected (dense) layers. The input layer feeds a word as a binary (one-hot) vector of V (size of vocabulary) dimensions, which is successively encoded in N intermediate nodes (dimensions). Then, a second fully-connected layer projects the N-dimensional vector into V output neurons in order to map the input word in a corresponding word presented in the same window. While the skip-gram model tries to predict a word in the window based on the central one, the CBOW model does the opposite by predicting the central word based on the rest of the words in the window.

Pennington et al. (2014) proposed the GLOVE algorithm, which similarly to the skip-gram variation relies upon word pairs (i.e., target and context word), while it differentiates in the fact that it trains two distinct set of vectors, word and context ones. The model is optimized based on the weighted least-squares loss, which rewards the correct predictions for pairs that are more frequent in the corpus. A word is finally represented as the sum of its corresponding word and context vectors.

Bojanowski et al. (2016) more recently introduced the FASTTEXT model, an upgraded form of the preceding WORD2VEC model, which successfully cope with the OOV (out-of-vocabulary) issue of the previous model. Both WORD2VEC and GLOVE rely on a fixed-size vocabulary, which means that there is a moderate possibility to miss a rare word. The main innovation of FASTTEXT is its ability to represent "never-seen" words (OOV) in the training corpora. To do so, FASTTEXT represents also character n-grams (subparts) of the inspected words, which may be later used to construct word embeddings for the unknown words based on the relevant n-grams.

Recent advances in the NLP community (Peters et al. 2018; Howard and Ruder 2018) established a new paradigm of pre-training context-aware word representations by training neural language models as a transfer learning method following the Computer Vision paradigm. Window-based methods, such as WORD2VEC and FASTTEXT do not produce context-aware representations by means of ignoring the word order and the particularities of each independent word. Involving language modeling in the process seems to provide contextual information, which further improves the quality of the word embeddings.

2.2 Current practices in the legal domain

Researchers in the domain of artificial intelligence and law incorporate word embeddings in their experiments for various tasks, like those that we examine in Sect. 3. We pinpointed in the literature the three main approaches related to the use of word embeddings:

- Generic pre-trained embeddings based on WORD2VEC, GLOVE or FASTTEXT models. There are several such models publicly available.[1] The main limitation lies in the fact that they have been trained over generic corpora, including Wikipedia

[1] You may find a large collection of such pre-trained models at https://github.com/3Top/word2vec-api.

articles, news articles, or randomly crawled web pages. They do not capture the semantics of the legal text, as they are rendered in domain-specific documents such as legislations, case laws, and other legal documents.

- Domain-specific word embeddings. Recently, many researchers tend to train their own embeddings, based on their annotated datasets, or a wider collection of relevant documents. Based on the published results, this approach seems to improve the performance of the models, as words are better represented while avoiding to inject noise present in the generic ones. Although this approach seems to improve the results, the initial idea that the word embeddings should be trained over large corpora is not satisfied. The training process mainly relies on the annotated datasets, a few thousand of paragraphs/sentences or, in the best case, a specific subset of documents (e.g., the criminal code of a given country or a selection of European policies, et cetera).
- Both generic and domain-specific embeddings. Many researchers have used both generic and domain-specific embeddings to provide their neural networks with a richer collection of features. This is a common practice when the in-domain corpus is small in order to reach a minimum level of word representation quality.

2.3 Law2Vec: legal word embeddings trained on large corpora

To the best of our knowledge, there are no publicly available word embeddings trained on large corpora. In this work, we present such a rich model called Law2Vec, which we also make publicly available for use in future experiments. The language of interest is English. In order to train the Law2Vec, we used a large number of legal corpora from various public legal sources. The list comprised the following:

- 53,000 pieces of the UK legislation.
- 62,000 pieces of the European legislation.
- 5500 pieces of the Canadian legislation.
- 1150 pieces of the Australian legislation.
- 800 pieces of the English-translated legislation from EU countries.
- 780 pieces of the English-translated legislation from Japanese.
- 68 bound volumes of the US Supreme Court decisions from 1998 to 2017.
- 54 titles of the most recently updated U.S. Code.

The corpus sums up to a total of 123,066 documents which consists of 492M individual words (tokens), including punctuation marks and numbers. The corpus was preprocessed to discard non-UTF8 encoded characters and treat dash-separated words due to different layout styles (e.g., text from PDF documents). The text was sentence splited using the NLTK library to provide the best possible input for the models. All words were lower-cased and all numerical digits where replaced by the character 'D', as in Chalkidis et al. (2017), in order to normalize numerical values.

We opted to train based on the WORD2VEC skip-gram model, instead of the most recent FASTTEXT implementation. The main reason is that WORD2VEC is reported to provide better semantic representation than FASTTEXT, which tends to be highly

Table 1 Top-5 similar words for a set of 20 selected words based on cosine similarity between the associated word embeddings

article	convention, section, articles, clause, provisions
act	statute, provision, mccarranferguson, irca, tvpa
action	suit, actions, lawsuit, claim, proceeding
crime	offense, murder, crimes, felony, violent
felony	offense, misdemeanor, felonies, offenses, convicted
punishment	penalty, punishments, sentencing, sentence, imprisonment
security	social, health, administration, retirement
fraud	fraudulent, theft, deceit, misrepresentation, bribery
privacy	confidentiality, communications, liberty, freedom, freedoms
intellectual	copyrights, patents, copyright, trademark, wipo
terrorism	terrorist, trafficking, counter-terrorism, violent, laundering
immigrant	immigrants, nonquota, alien, asylum, citizenship
illegal	unlawful, corrupt, improper, illicit, fraudulent
drugs	drug, narcotic, addicts, psychotropic, medicines
appeal	appeals, review, hearing, appellate, appealed
abuse	violence, sexual, self-destructive, assault, mistreatment
alcohol	liquor, spirits, intoxicating, beer, vinous
complaint	grievance, allegations, allegation, complaints, counterclaim
indictment	conviction, summary, imprisonment, indictable, triable
motion	motions, petition, dismiss, leave, cross-motion

biased towards syntactic information, as well as the computed n-gram embeddings. Missing words (OOV) is not of concern in most legal-related tasks, as legislators, lawyers and other legal professionals articulate in high quality standards. We empirically observed that legal documents have been consistent by means of misspellings, grammatical/syntactical errors, as well as the vocabulary being formal and pertinent to the domain.

We trained two individual WORD2VEC models for 100-dimensional and 200-dimensional embeddings using the GENSIM library. We used 5-word windows and a threshold of ten occurrences as the minimum for comprising a word in our vocabulary. This leads to a final vocabulary of 169,439 words (equiv. types) for each model. Turning the acceptance threshold higher than the default configuration (i.e., five occurrences) was also based on the intuition and practical observation that missing words (OOV) is not an issue in the current domain. Therefore, we avoided including very rare (most probably misspelled) words or words which were corrupted during the pre-processing phase. These words would likely have inadequate embedding representations due to insufficient training samples.

As there is no formal procedure for the evaluation of word embeddings, we present a qualitative analysis, as it is demostrated in Table 1. We look forward to further experimentation and evaluation on the Law2Vec by the research community. The Law2Vec models are published in https://archive.org/details/Law2Vec.

3 Literature review

3.1 Text classification

3.1.1 Classifying sentential modality in legal language: a use case in financial regulations, acts and directives

O'Neill et al. (2017) studied sentential modality classification in legal language. Particularly, they focused on deontic modalities which express obligations, prohibitions, and permissions. Although modal logic has been effectively applied in many applications in the legal domain to deal with the strict legal language, there are several reasons to indicate that classical logic in classifying sentential modality may be ineffective. According to authors, modality in legal language strongly relies on the use of modal verbs which, as was demonstrated with examples in the article, have multiple functions, as they may also be misused in several occasions. These are strong indications that the context in each case is very important for deciphering the actual role of these modal verbs, hence a machine-learning approach seems more promising.

The authors experimented with an in-house annotated dataset comprised Financial Regulations, Acts, and Directives. The training set consisted of 1297 annotated sentences, including *obligations*, *prohibitions*, and *permissions*; while the gold standard test set was composed of 622 sentences. The documents were annotated by domain-experts with an inter-annotator agreement of 0.74. The authors claimed that disagreements appeared mostly between obligations and prohibitions, which is indicative of the complexity of the task; as we may expect, obligations and prohibitions could be easily separated by means of negation operators (e.g., 'shall [not]', 'have [no] right', et cetera).

Authors experimented with various methods, including Logistic Regression, SVM s, AdaBoost, Gradient Boosting and Random Forests, in order to have a respectful range of benchmarks against Artificial Neural Networks (ANN s). Those methods were examined using two alternative feature representations. In the first, baseline features such as n-grams, POS tags, and normalized TF–IDF scores were considered. In the second, sentence concatenations of the available word embeddings (i.e., Google vectors and legal embeddings) were used. Google vectors are pre-trained in Google news articles, which led to generic 300-dimensional word representations. Therefore, the authors also pre-trained their own 100-dimensional legal embeddings over a corpus of approximately 7500 EU legislation documents. They also pre-trained 100-dimensional legal phrase embeddings, but no further details were provided with respect to the phrase detection/segmentation strategy.

The different Neural Network methods that were considered are the following:

- A Multi-Layer Perceptron (MLP) including two fully-connected hidden layers.
- Unidirectional and bidirectional LSTM s, followed by a fully-connected layer.
- Multi-filter CNN s for windows = [2,3], followed by max-pooling layers that were finally concatenated and followed by a fully-connected layer.

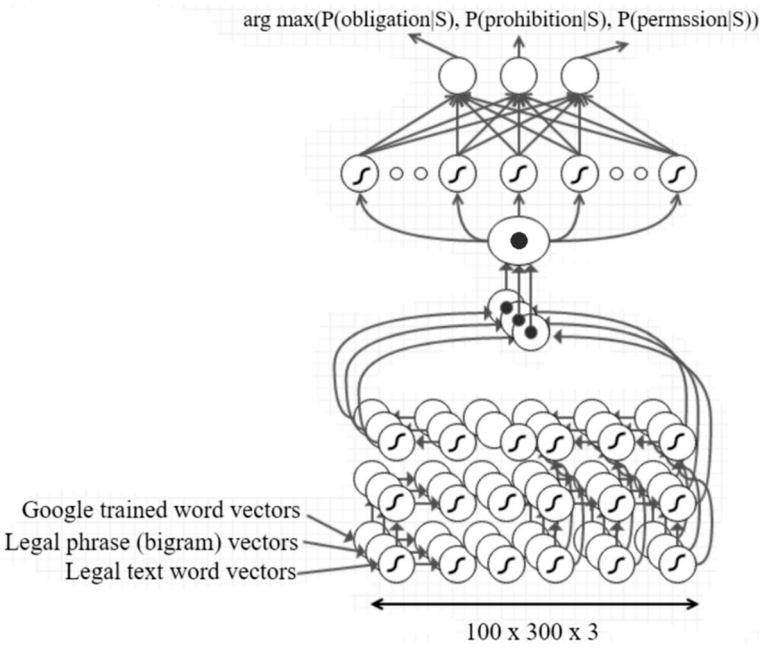

Fig. 1 Multi-embeddings BILSTM s presented in O'Neill et al. (2017)

- CNN s followed by LSTM s, followed by a fully-connected hidden layer.

All neural methods were examined using two alternative feature representations. First, only the Google vectors were considered. Second, as it is demonstrated in Fig. 1, the authors also used their domain-specific pre-trained word and phrase embeddings, to feed LSTM s or CNN s. The output was finally concatenated before feeding the fully-connected layer, which was used in each of the ANN models.

The final results considering all different methods and ablations led to the following conclusions:

- Classic ML algorithms were vastly overfitting the training set trying to cope with the complexity of the task, leading to performance decline (e.g., 0.61 F1, 63.12 acc. for SVM) in the test dataset by approximately 20–25% decrease compared to the great performance on the training set.
- All Neural Networks, except the naive MLP (0.56 F1, 58.71 acc.), outperformed the classic ML algorithms by over 10% on average, indicating the superiority of RNN s and more specifically LSTM s, when dealing with complex language semantics. The RNN s exploited information from the sequential structure of the text, leading to context-aware representations.
- BILSTM s outperformed the unidirectional LSTM s by approximately 3–4% on F1 (0.76) and Accuracy (78.56), which is common in almost every NLP task. The right-to-left direction conditions each word with information that resides on the succeeding words. They also outperformed naive CNN s approaches (0.68

F1, 71.40 acc.), which encode fixed-sized n-grams, instead of arbitrary context dependencies over the word representations.
- Providing domain-specific (legal) embeddings further improved the performance of the neural methods and boosted the accuracy of the best-of method (BILSTM S) from 78.56 to 82.33.

3.1.2 Obligation and prohibition extraction using hierarchical RNNs

Based on the paradigm of O'Neill et al. (2017) and Chalkidis et al. (2018) improved the state-of-art on sentence modality classification. They studied a similar task for extracting obligations and prohibitions from service agreements. The authors discussed the complexity of legal language in terms of the structural notations that lawyers widely use and proposed a more refined scheme of annotations that incorporate sub-sentence classification.

The authors experimented with an in-house dataset containing 6385 training, 1595 validation, and 1420 test sections (articles) from the main bodies of a hundred randomly selected English service agreements. In terms of (sub-)sentences, the sections were finally split into 31,545 training, 8036 validation, and 5563 test sentences. Each service agreement was annotated by five law students (one student per sample), and all annotations were subsequently checked by a single paralegal expert. The annotation scheme included different tags for *obligations*, *prohibitions*, *list headers* that were part of *obligations/prohibitions*, *list items* that were part of obligations and *list items* that were part of prohibitions.

The authors used domain-specific pre-trained 200-dimensional word embeddings and 25-dimensional POS embeddings, obtained by applying WORD2VEC to approximately 750K and 50K English contracts, respectively, provided by Chalkidis and Androutsopoulos (2017). They also used 5-dimensional token shape embeddings (e.g., all capital, all digit), also provided by the same article. Each token was represented by the concatenation of its word, POS and shape embeddings. Unknown tokens were mapped to pre-trained POS -specific unk embeddings (e.g., unk-n, unk-vb).

The experiments included several LSTM -based methods (Fig. 2):

- Word-level BILSTM S followed by a fully connected layer fed with the concatenation of the last hidden representation of both the forward and backward LSTM chains, which operated on each sentence independently.
- Similar word-level BILSTM S were deployed in order to produce context-aware word embeddings. The context-aware word embeddings were subsequently sum up as a weighted average to form the initial sentence representations. The weighting scheme was provided by a self-attention mechanism, which assigned attention scores (weights) in each context-aware embedding, based on a fully connected layer, followed by the softmax function. Similarly to the previous method, each sentence was fed and classified independently.
- Similar BILSTM S with the self-attention mechanism, in which the BILSTM chains incorporated neighboring tokens (150 before/after the target sentence). The outputs of the BILSTM S for the neighboring tokens were not taken into consideration

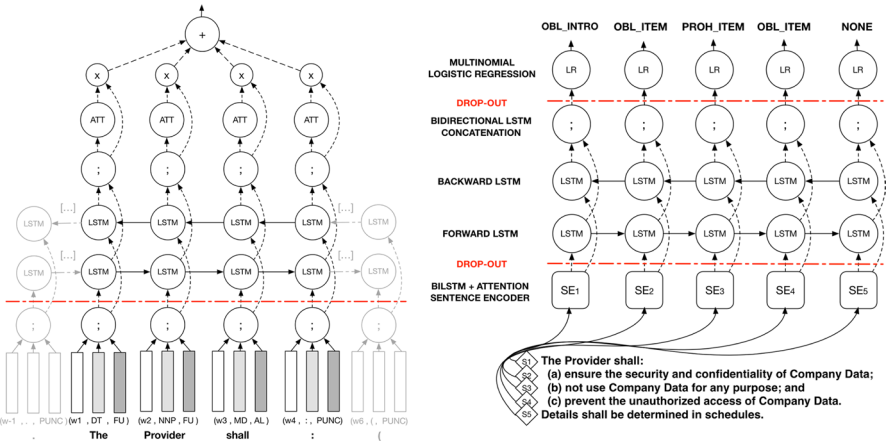

Fig. 2 Hierarchical RNNs presented in Chalkidis et al. (2018)

in the summation of the self-attention mechanism. Each sentence was fed and classified independently according to the previous method.

- A hierarchical model, which facilitated high-order BILSTM s operating over sentence representations, produced by a sentence encoder identical to this used in the second method. Here, an entire section, formed as a sequence of sentences, was fed at once and the individual (sub-)sentences were also co-conditioned and produced context-aware sentence representations. The produced context-aware sentence representation was finally fed into a fully connected layer in order to be classified.
- The hyper-parameters were tuned using grid-search on manually selected sets of values of LSTM hidden units, dropout rate, and batch size.

The evaluation results led to the following conclusions:

- The self-attention mechanism introduced in the second method led to an overall improvement (in macro and micro F1 and AUC) compared to the plain BILSTM, supporting the hypothesis that the classifier focused on indicative tokens.
- Allowing the BILSTM to consider neighbor tokens (i.e., sentences) did not exhibit any improvement. A reason might be that the neighbor tokens were not encoded in a structured manner.
- The hierarchical model significantly outperformed the other three methods, supporting the hypothesis of profiting from considering entire sections and allowing the sentence embeddings to interact in the upper BILSTM.
- This model is also significantly faster to train than the previous ones, even though it had more parameters.

The authors provided a heatmap of attention scores that were produced on selected sentences. The attention scores were higher for modals, negations, words that indicate obligations or prohibitions (e.g., obliged, only) and tokens that indicate nested

clauses (e.g., (a), :, ;). This allows the related methods, which apply attention, to focus on specific tokens that provide a strong indication of the corresponding classes of the sentences. As already mentioned, the attention mechanisms should be considered a respectful path towards prediction explainability.

3.1.3 Inducing predictive models for decision support in administrative adjudication

Branting et al. (2017) experimented with predictive models for assisting Administrative Adjudication. Their intuition was that, predictive models trained over previous decisions for a specific administrative body can improve subsequent decision-making processes. The authors examined three classification methods on three automatically created datasets. Only two of these datasets were used for the training and evaluation of Neural Networks:

- The first dataset includes Board of Veterans Appeals (BVA) cases. These cases comprise the following sections: *issues*, *introduction*, *findings*, *conclusions*, and *reasons*. Many cases include multiple (N) issues but, based on the structure of the filings, the authors achieved to split each case into N individual samples. Each sample included as the fact input (x), the full introduction and the Nth issue's findings, paired with its issue classification (y). The possible decisions of each issue were (1) the requirements for benefits have been met, (2) the requirements have not been met, (3) the case must be remanded for additional hearings, and (4) the case must be reopened. Conversion of all published BVA cases in this fashion yielded 3844 4-class instances or 1605 2-class (met or unmet) instances.
- The second dataset consisted of cases brought to the World Intellectual Property Organization (WIPO), related to complaints about domain names that possibly violated trademarks. The facts (input) of each instance (sample) consisted of the first five sections, excluding findings and decision; and the related classification tags were *transferred* and *not transferred*. The WIPO dataset consisted of 5587 instances. As the authors noted, the specific dataset was quite imbalanced with a rate 10-to-1 in favour of cases which are labelled as *transferred*.

The authors first experimented with SVM s that operated on n-gram frequency vectors in order to establish a benchmark versus the examined Neural Networks. The evaluation was carried out using tenfold cross-validation. With respect to the Neural Networks, the authors extended the hierarchical networks of Yang et al. (2016) in order to better describe the deeper structure of the examined documents. Yang et al. (2016) proposed a 2-level encoding mechanism for text classification tasks. In the first level, a BIGRU encoder was deployed to produce context-aware word embeddings, which were subsequently summed up based on an attention mechanism that indicates the weighting scheme to form sentence representations. In the second level, the BIGRU S produced context-aware sentence representations that were also followed by a similar attention mechanism to create a final document (paragraph) representation. The latter finally passed through a fully-connected layer and softmax function in order to be classified. In our case (Fig. 3), the authors processed documents that comprised

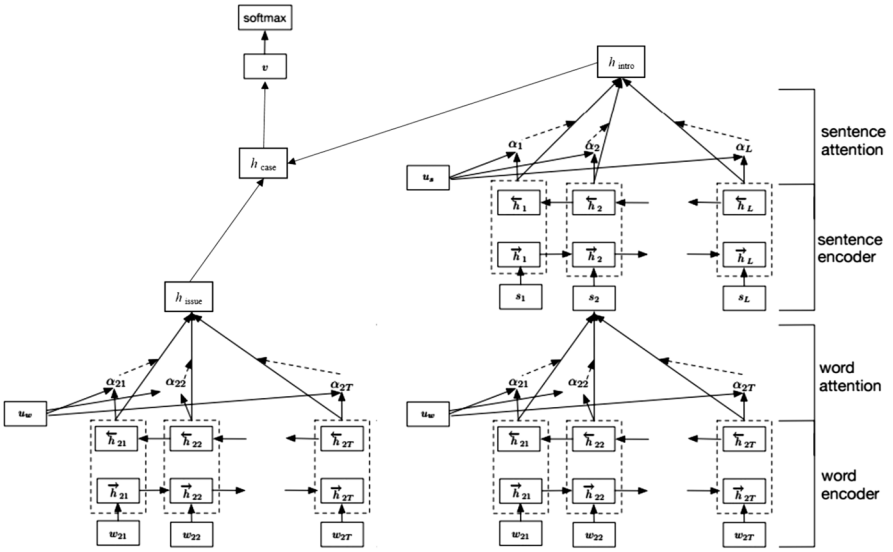

Fig. 3 Hierarchical GRUs presented in Branting et al. (2017)

multiple sections (paragraphs), so they extended the former model with an intermediate fully-connected layer, which combined the underlying paragraph representations in order to form a deeper document representation. The authors stated that an extended RNN (i.e., a 3rd GRU encoder on top of the paragraph encoders) was also examined, but the fully-connected layer performed better. In terms of the word representation used for the Neural Networks, the authors pre-trained domain-specific word embeddings based on the datasets using the WORD2VEC model.

All Neural Network models related to the individual datasets were trained on 80% of data, with an additional 10% as the validation set, and the remaining 10% reserved for testing.

The neural model achieved a mean F1 of 73.8% and overall accuracy of 74.7% in BVA cases. For the WIPO dataset, the same architecture reached a mean F1 of 94.4% and 94.4% accuracy. In tenfold cross-validation, the SVM achieved a mean F1 of 73.1% on the BVA data set, with an overall accuracy of 73%. A mean F1 of 95% was achieved on the WIPO data set yielding an overall accuracy of 90.5%. Based on the inconsistent evaluation methodologies (train/validation/test vs. tenfold CV) between Neural Networks and the SVM approach, it is not clear whether one approach performed better, as the current evaluation results are highly competitive (approx. 0.73–0.74 micro-F1 and 0.73–0.75 accuracy for BVA dataset, and 0.94–0.95 micro-F1 and 0.91–0.94 accuracy for the WIPO dataset). A fair comparison would be plausible if one of the two evaluation approaches was used uniformly in both methods.

As we already highlighted in the previous article, attention mechanisms are able to highlight (i.e., assign bigger attention scores) parts of the texts (i.e., words or sentences) that are highly informative for the given task. The authors observed that

medical ailments and disabilities' percentage described in the facts of the BVA cases were weighted most heavily, while information which resided in the introduction section was less interpretable because that section mainly included legal formalities. Further analysis of the attention scores and the relevant decisions could be very informative of possible correlations between them and also assist the procedure by highlighting the key factors, as the authors demonstrated in a decision support prototype framework.

3.1.4 Discussion

In the aforementioned articles, gated RNN s (i.e., LSTM s, GRU s) were utilized as the main component of the systems (neural networks) in order to provide context-aware/task-specific word representations. In the two latter articles, the authors investigated the use of self-attention as a useful mechanism to build representative sentence encodings, while also providing useful insights (highlighting) for systems' decisions. Both of them also exploited hierarchical networks to better encode information with respect to the text structure and the cross-segment relationships. These three practices are aligned with the advances in the NLP literature. However, there is no extended research with respect to CNN-based architectures that are rapidly introduced in text classification tasks, offering competitive results while being trained much faster than the gated RNN s.

3.2 Information extraction

3.2.1 Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts

Nguyen et al. (2017, 2018) proposed several approaches that utilize Deep Learning models to recognize requisite and effectuation parts (segments) in legal documents. Legal sentences are long, complicated and usually represented in specific structures. In almost all cases, a legal sentence can be separated into two main parts: A *requisite* part and an *effectuation* part. For convenience, we call them RE parts.

The authors experimented with various BILSTM-CRF models, while they also presented several ablations of these models. There were two related datasets for the given task:

- The Japanese National Pension Law RRE dataset (JPL-RRE) contained sentences that were segmented into chunks. Each chunk was labelled using lower level categories (*topic parts*, *antecedent*, and *consequent* parts) to represent RE parts, thus can be used as a unique labeled set.
- The Japanese Civil Code RRE dataset (JCC-RRE), which included the English-translated version of the Japanese Civil Code, was annotated manually by three annotators. This dataset contains three type of logical parts: *requisite*, *effectuation* and *exception* parts, which describe exceptions in law sentences. In contrast to the JPL-RRE dataset, RE parts in JCC-RRE may overlap. Therefore, RE parts

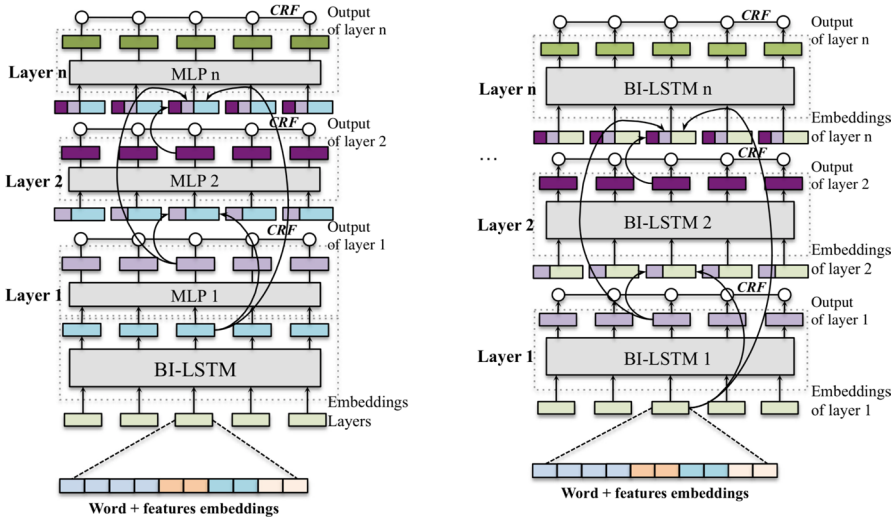

Fig. 4 Cascaded BILSTM-CRF models presented in Nguyen et al. (2018)

in this dataset were labeled in three independent groups, one for each logical part, thus cannot be mixed and form a single set of annotations.

The authors represented each token using the following features: 100-dimensional word embeddings, self-trained by the networks or pre-trained using WORD2VEC over a collection of Japanese legal documents; 10-dimensional self-trained POS tag embeddings; and 10-dimensional self-trained chunk embeddings (verb and noun phrases).

The authors experimented with four derivatives of BILSTM-CRF (Fig. 4):

- A single BILSTM-CRF model, which was deployed for the JPL-RRE dataset. This model was initially fed with word, POS tag, and chunk for each token, which were embedded in three individual vectors representations. These three embeddings were then concatenated and passed into a bidirectional LSTM chain, which produced context-aware token embeddings. Forward and backward context-aware token embeddings were then concatenated and passed into a chain of linear CRF s, which classified each token.
- Three consecutive BILSTM-CRF models, which were deployed for the JCC-RRE dataset. These models were fed with word, POS tag, chunk, as the previous model, while they also leveraged the previous model's prediction for each token. Each model predicted another group of tags (requisite, effectuation, exception) and was trained independently.
- A cascaded network consisting of three BILSTM-CRF s models, which was deployed for the JCC-RRE dataset. This model was a unified version of the previous models, which was trained jointly based on the losses from all CRF layers. In case of the second and third BILSTM-CRF, each BILSTM-CRF sub-model was fed with the initial feature representations (word, POS tag, chunk), con-

catenated with the BILSTM outputs of the previous sub-models. Each CRF layer predicted another group of tags (requisite, effectuation, exception).

- A cascaded network comprised three BILSTM-MLP-CRF S model, which was deployed onto the JCC-RRE dataset. This model is similar to the previous, while it also incorporated one or two fully-connected layers between each bidirectional LSTM chain and the corresponding CRF layer. In the case of the second and third BILSTM-CRF, the initial feature representations (word, POS tag, chunk) were concatenated with the MLP outputs of the previous sub-models, instead of the BILSTM outputs.

The authors evaluated the performance of the aforementioned models in the relevant datasets (JPL-RRE, JCC-RRE), based on precision, recall and F1 scores. They also reported interesting results based on ablation tests related to many different aspects. The main results were the following:

- The BILSTM-CRF model outperformed the CRF baseline by 4.46% (93.27 vs. 88.81) in the JPL-RRE considering the full feature set, which included both the aforementioned features (word, POS tag, chunk) and additional gazeteers of headwords, function words, and punctuation features.
- The authors compared six different methods with the JCC-RRE dataset: CRF S, BILSTM S, BILSTM-CRF S, joint BILSTM-CRF S, joint BILSTM-MLP-CRF S with one or two fully-connected hidden layers. Overall, the joint BILSTM-MLP-CRF model with two fully-connected hidden layers performed better than the rest with a 78.24 macro-averaged F1 score.
- The joint BILSTM-MLP-CRF model outperformed by 4.54% (78.24 vs. 73.7) the best benchmark (i.e., CRF S with pre-trained word embeddings, self-trained POS tag, and chunk embeddings).
- Using pre-trained word embeddings provided an important improvement to the best model by 2.49% in a macro-averaged F1 score, which was also the case with the use of syntactic features (POS tag, chunk) that increased the performance by 1.92%.
- Comparing the training of BILSTM-CRF S models and a single joint BILSTM-CRF S model or joint BILSTM-MLP-CRF S showed that the results were comparable with a minimum difference of 0.08% between BILSTM-CRF S and joint BILSTM-MLP-CRF S with two fully-connected hidden layers. Both training and testing time were comparable too.
- Testing end-to-end independent BILSTM-CRF S models, meaning that the evaluation was performed using the predicted labels of the previous model instead of feeding gold-labels (single evaluation), showed that the second model (effectual parts) underperformed by approximately 20% compared to the single evaluation due to the erroneous feedback from the previous model (requisite parts). This is not the case for the last model, which predicted exception parts indicating that this sub-task is not correlated with the two former ones.

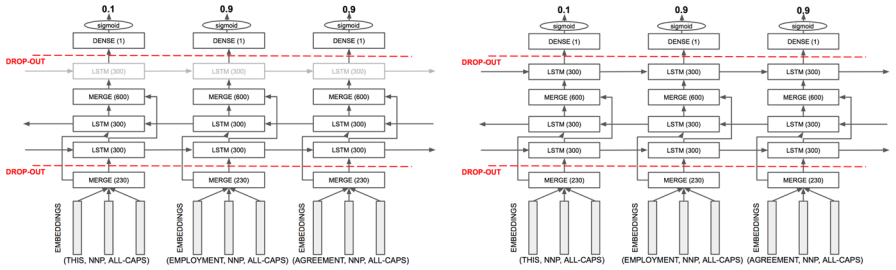

Fig. 5 LSTM-based models presented in Chalkidis and Androutsopoulos (2017)

3.2.2 A deep learning approach to contract element extraction

Chalkidis and Androutsopoulos (2017) 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. In Chalkidis et al. (2017), the authors released a benchmark dataset of approximately 3,500 English contracts, annotated with eleven types of contract elements. Using this dataset, they experimented with Logistic Regression and linear Support Vector Machines (SVM s), both operating on fixed-size sliding windows of tokens, represented by hand-crafted features, pre-trained word embeddings, and/or pre-trained part-of-speech (POS) tag embeddings. They also experimented with manually written rules that replaced the machine learning classifiers or post-processed the classifier's outcome.

The pre-trained 200-dimensional word embeddings and 25-dimensional POS tag embeddings that accompany the dataset were obtained by applying WORD2VEC to approximately 750K unlabeled and 50K POS-tagged English contracts.

In the current article, the authors examined Deep Learning methods and more specifically LSTM -based ones. As in their previous work, they built a separate extractor for each contract element type (e.g., contracting parties). While the authors employed new methods, they also introduced 5-dimensional token-shape embeddings that represented six possible shapes of tokens (i.e., uppercase, lowercase, first character uppercase, digits, punctuations, other). The token shape embeddings were obtained by applying WORD2VEC to approximately 2000 contracts.

The authors examined three alternative LSTM -based methods (Fig. 5):

- The first method included a bidirectional LSTM chain fed with the concatenation of word, POS tag and token-shape embeddings, which produced context-aware token embeddings. The context-aware token embeddings passed through a fully-connected layer followed by a sigmoid activation function.
- The second method extended the first one by injecting an additional unidirectional (forward) LSTM chain between the first bidirectional LSTM chain and the fully-connected layer.
- The third method was similar to the first method but, instead of using a fully connected layer or additional LSTM s, it employed a linear chain of CRF s.

The new methods were trained using binary cross-entropy loss, and the Adam optimizer, with early-stopping to examine the validation loss. The dropout rate, learning rate, and batch size were tuned in a grid-search fashion with threefold cross-validation.

As in their previous work, the authors evaluated the new methods in two different manners. Firstly, they evaluated each method per token, which means evaluating the performance of classification token by token. Secondly, they evaluated each method per element, which means evaluating the performance of classification over complete contract elements, which were multi-word expressions. For both evaluation practices, they reported precision, recall, and F1-score.

With respect to the first evaluation practice (per token), the results were the following:

- The three LSTM -based methods performed better than the linear sliding-window classifiers of their previous work. Even the first method, the simplest of the three LSTM -based methods, exceeded the macro-averaged F1 score of the best previous methods by six points (0.86 vs. 0.80). The additional LSTM chain of the second method slightly improved the macro-averaged F1 score (0.87).
- The second method obtained top F1 scores for all but the *contract period* type, and for some element types (notably for *termination date* and *contract value*) it outperformed the first method (Fi-score: 0.79 vs. 0.75, and 0.68 vs. 0.63). The third method BILSTM-CRF had the same macro-averaged F1 as the second one (0.87), but it did not perform better in any contract element type, except for *contract period*.
- The lowest F1-scores of all three LSTM -based methods were for *contract period*, *termination date*, and *contract value*, which were the three contract element types with the fewest training instances in the dataset. This indicates that the dataset size is a key factor in deep learning.

Considering the second evaluation practice (per element), the results were the following:

- The simplest LSTM -based method, equals the macro-averaged F1 score (0.86) of the best linear sliding-window classifiers, without using any manually written rules, unlike the sliding-window classifiers that relied extensively on the post-processing rules in these experiments.
- The macro-averaged F1 score of the best linear sliding window classifiers, without the post-processing rules, dropped to 0.69. The authors stated that this is particularly important because the post-processing rules were very difficult to maintain in practice. The extra LSTM layer of the second method improved the macro-averaged F1 of the first one by one point (0.87 vs. 0.86), but the third method (BILSTM-CRF) performed even better overall (0.88).
- Overall, BILSTM-CRF appeared to be better than the second method in the experiments of this section, in contrast to the experiments of the previous section, which indicates that the CRF chain can better classify complete contract ele-

ments because it jointly selected the assignment of positive or negative labels to the entire token sequence.

3.2.3 Discussion

In the aforementioned articles, as in those of Sect. 3.1, LSTM s were utilized in order to provide context-aware/task-specific word representations. Both articles reported great performance improvements utilizing BILSTM-CRF models compared to traditional ML algorithms (i.e., SVM s, LR, CRF s), while they also reported marginal improvement compared to networks that do not employ CRF s on top of LSTM s.

3.3 Information retrieval

In this section, we discuss seven selected papers that address the retrieval of relevant excerpts of text with respect to a particular query. The majority of the papers address the legal question answering task of the Competition on Legal Information Extraction/Entailment (COLIEE) from 2014 to 2017. Particularly, the competition focuses on two aspects related to a binary (yes/no) question answering as follows: Phase one of the legal question answering task involves reading a question Q and extract the legal articles of the Civil Code that are relevant to the question. In phase two the systems should return a *yes* or *no* answer if the retrieved articles from phase one entail or not the question Q.

3.3.1 A convolutional neural network in legal question answering

In phase one of COLIEE, Kim et al. (2015) introduced an ad-hoc information retrieval method for retrieving Japan civil law articles related to a given question by employing a TF–IDF weighting method to capture the correlation of a query to an article, according to the word sets overlapping. To increase the methods efficiency and generalization ability, they retained only the words' lemmas, instead of the actual words. The parameters were normalized to prevent a bias towards longer documents, which is a well-known error in ranking methods. The Ranking SVM model was alternatively applied to rank relevant documents according to users' feedback. The three types of features used for this method were binary representations of lemmas, dependency pairs in order to capture the prominent semantic content and TF–IDF scores.

In phase two of COLIEE, a binary classification model was introduced for *yes*/*no* answering to the legal queries. The authors assumed that the correct answer has a high semantic similarity to a question. The semantic representation of the questions was comprised of word embeddings and linguistic features. They trained a classifier based on a triple (q_i, a_{ij}, y_i), where q_i was the question, a_{ij} was the jth sentence of the ith article, and y_i was the response (i.e., yes or no). The classifier learned the probability p(y = yes| q, a) of a sentence being relevant to a question.

For the purpose of their experiments, authors trained 50-dimensional word embedding with the WORD2VEC model based on the data provided during the former

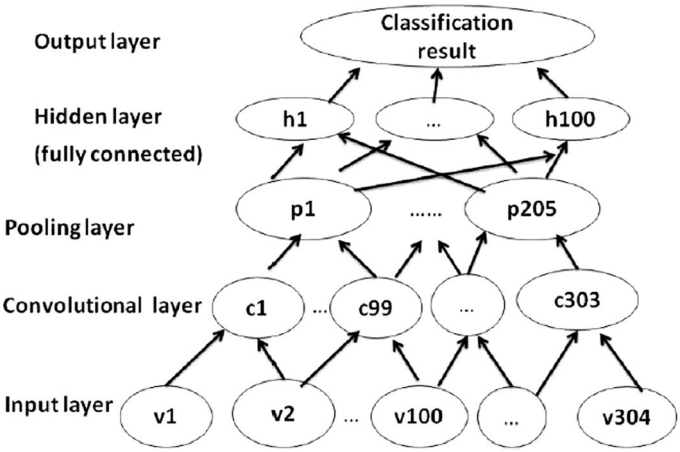

Fig. 6 The architecture of the CNN presented in Kim et al. (2015)

COLIEE 2015 competition. The authors introduced a CNN -based classifier (Fig. 6), structured as follows:

- The feature representation for each (q, a) pair includes the concatenation of word embeddings centroids for both the given question and the current examined article; two pairs of word embeddings for the root element of the dependency trees for the identified condition and the conclusion in both the question and the answer; four binary features for the existence or not of condition and conclusions in both question and the relevant article. The feature representation, consisting of 304 features (real numbers), was fed into a convolution layer, followed by a pooling layer, which produces a summation of the convolution layer outputs.
- Two fully-connected layers were used on top of the pooling (summation) layer; the first one had the same dimensionality as the pooling layer and the second one had a single neuron followed by a sigmoid function for the final prediction (yes/no).
- The dropout technique was used to prevent the model from overfitting to the training data. The authors tuned the dropout rate between 0.6 and 0.7 for the hidden layers and 0.1 for the input layer. The Rectified Linear Unit (ReLU) was used as an activation function across all layers for faster training. The training was performed using mini-batches and the SGD optimizer.

For phase two, the authors evaluated their models in a quite balanced dataset (55.87% yes and 44.13% no answers) including 179 questions. The accuracy of the proposed CNN model with the pre-trained word embeddings and other relevant linguistic features, including dropout, outperformed a linear SVM model, as also a rule-based model with K-means clustering. The same CNN model exhibited a lower accuracy by 1.22% while discarding word embeddings and drop out. The model's performance was comparable to the baseline (majority-class) classifier's accuracy

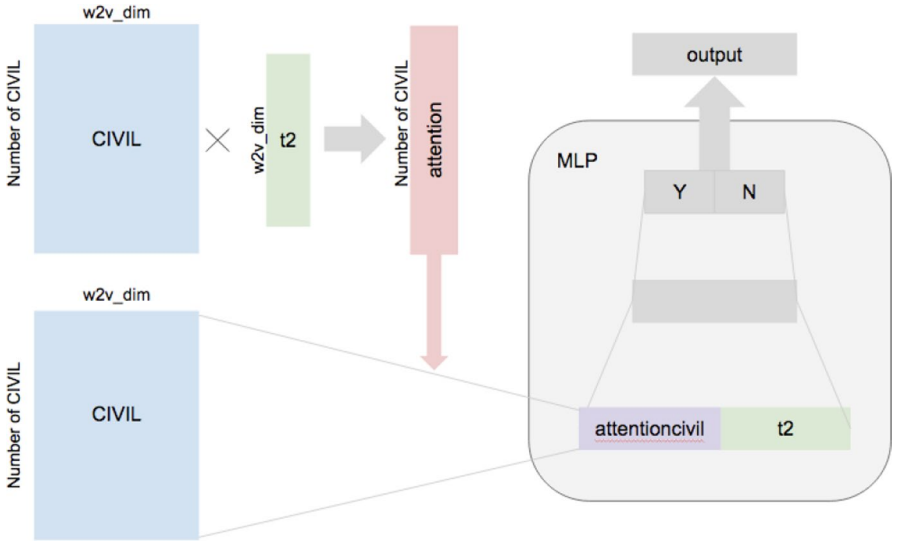

Fig. 7 An Attention model presented in Morimoto et al. (2017)

(55.87%). The highest accuracy reported for the first mentioned model was 63.87%, significantly higher than the baseline score and slightly better than the SVM (60.12%).

3.3.2 Legal question answering system using neural attention

In phase one of COLIEE, the methodology that Morimoto et al. (2017) followed to identify the similarity of a query to a civil law article comprised the extraction of the requirement (condition), the effect (conclusion) in law articles and the examined query T. The authors extracted the legal requirement and effect parts from queries and articles using rule-based (i.e., pattern-matching) methods. The distance between a query Q and an article T was calculated as the sum of the distances of the required parts of Q and T and the distance between the effect parts of Q and T. The articles to be retrieved should exceed a pre-defined threshold. The decision of the selection threshold was tuned until the optimal one was identified. The Word Movers Distance was used to calculate the above-said distances. The authors considered the negations at the end of a sentence since Japanese is a head-final language.

The entailment model was based on Neural Networks with attention mechanism (Fig 7). The civil_vec[i] was the vector representation of the i-th article, which was a TF–IDF weighted sum of the word representations (embeddings) in the article, while t2_vec was the vector representation of the query given the same formula. The calculation of the attention was done by the inner product of the transpose of civil_vec and t2_vec, which led to a single vector whose size was the number of articles.

The authors pre-trained domain-specific word embeddings of 50 and 200 dimensions using WORD2VEC. The word embeddings were trained on judgment documents extracted by the Japanese Supreme Judicial Court. The dataset consisted of 58,808 judgments of 4M sentences in total. To train the word embeddings, the authors first

segmented the words and extracted the POS tags by applying the MeCab tool (Kudo et al. 2004).

The second task of COLIEE was tackled by introducing a multi-layer perceptron (MLP), which received as inputs the concatenated attention_civil and t2_vec vectors. The number of hidden units of the said three-layer perceptron was fifty and the output units were two, encoding the *yes*/ *no* potential output values. They experimented on different inputs, optimizers and activation functions. Particularly, they utilized the sigmoid function and the AdaDelta optimizer - NAIST1 setup. In the NAIST2 setup, they used the tanh activation function and the Adam optimizer, considering the Suffix. In the last experimental setup, named sigMoMWE, they considered both the suffix and functional expressions using the sigmoid for the activation function and the MomentumSGD as the optimizer.

The experimental results showed that the model that achieved the highest accuracy (i.e., 0.6667) on the test set was the sigMoMWE model. The NAIST2 achieved 0.6538, while the NAIST1 achieved 0.6154. An important remark was that all three models achieved 0.6351 on the validation set. The authors claimed that the dropout experimentation between 0.2 and 0.5 in all the aforementioned experimental settings didn't provide any better results. The authors broke down the analysis, measuring the precision and recall of both *yes* and *no* cases separately. They found out that their model was balanced towards the two classes.

3.3.3 Legal information retrieval using topic clustering and neural networks

Nanda et al. (2017) used a combination of a partial string matching and a topic clustering method in order to tackle the Information Retrieval task in phase one. The authors utilized two alternative methods. Initially, they applied a simple pattern matching method by capturing the similarity of a Question Q and an answer A based on the intersection of the common words. Due to the polysemy and synonymy of words, some queries may be relevant to particular articles without significant intersection of the words used. Therefore, the authors introduced a method that relies upon the semantic similarity. They used the Latent Dirichlet Allocation (LDA) topic model to represent each article and query with a topic vector. The similarity closeness of a query-article was calculated on the topic vector. To treat the synonyms of the LDA topic words that may exist in a query, the authors turned to the Wordnet.

Once the most relevant articles were retrieved, the textual entailment model was applied to identify whether an article entails a query. The authors claimed that the Textual Entailment task was usually tackled using systems that rely on feature engineering and selection. Nevertheless, Nanda et al. introduced a system which combines LSTM s with CNN s. Particularly, the proposed system was summarized as follows (Fig. 8):

- The authors used the 300-dimensional word embeddings from the Google News vectors to represent the articles and the queries.
- The neural model was fed with sentential sequences of the article and the query concatenated to a 600-dimensional vector which was fed into the LSTM. The activation function of all the layers was the rectified linear unit (RELU).

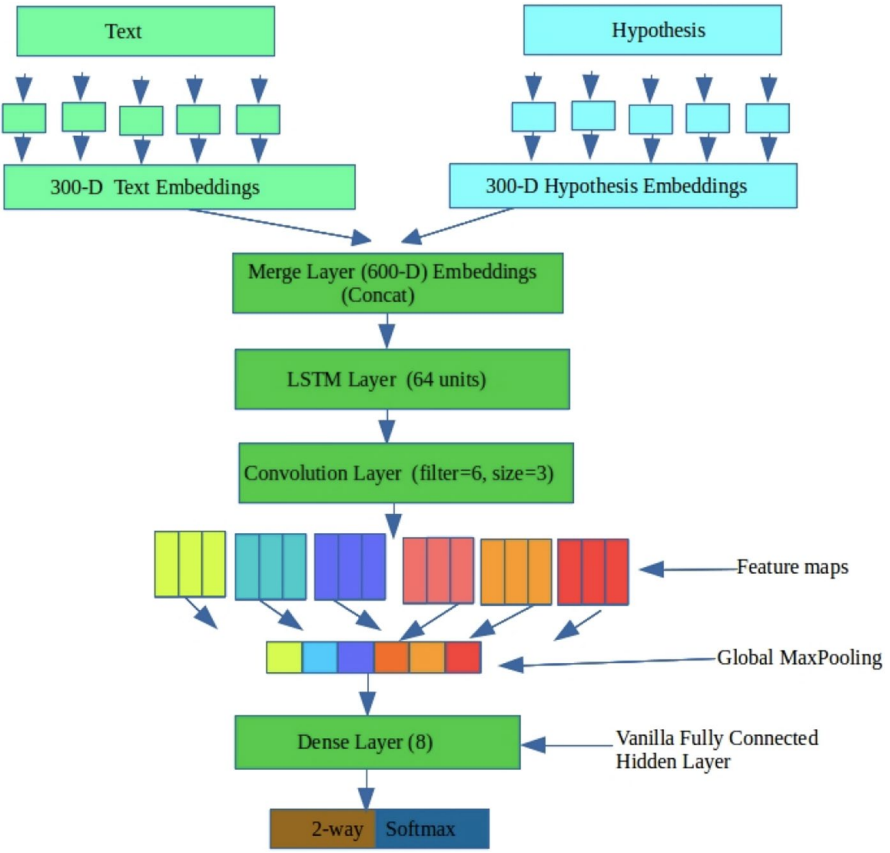

Fig. 8 The LSTM-CNN model presented in Nanda et al. (2017)

- The LSTM layer projected 64-dimensional vectors that were fed sequentially into a convolutional layer of six filters. A maxpooling layer retained the most prominent features.
- In the end, two fully connected layers were added, where the latter is followed by the softmax function to distribute values between 0 and + 1 for representing non-entailment and entailment respectively. The architecture of the network is depicted in Fig. 8.

Regarding the textual entailment task, the LSTM-CNN model achieved a bit lower accuracy (i.e., 0.538) compared to ILiS (i.e., 0.576). The authors stated that this occurred due to the Google News word embeddings that were not tailored to legal text. The LSTM-CNN model outperformed slightly the JAISTNLP (i.e., 0.512) and the JNLP (i.e., 0.487).

3.3.4 Legal question answering using ranking SVM and deep convolutional neural network

Do et al. (2017) highlighted the complexity of the legal text by means of the domain-specific terminology, the concepts and the logical structure, argumentation, and ambiguity. A well-known approach to deal with legal language complexity is the engineering and incorporation of manually crafted knowledge bases into information retrieval systems. Essentially, the prior knowledge incorporated reflects the concepts and domain expertise of legal experts. The main limitation of this approach is the cost to engineer and maintain the information in the knowledge base. In contrast, NLP techniques, although computationally expensive, are more practical since they yield promising results by processing huge amounts of data that humans cannot.

For the legal Information Retrieval (IR) task, a Ranking SVM model was used on the feature vectors of the query-article (QA) pair. The QA pair-fed a binary CNN classifier for question answering. The dataset used was the Japanese Civil Code. After a pre-processing step of splitting each multi-paragraph article into single paragraphs, the authors extracted 1663 single-paragraph articles. Moreover, the common tasks of stopword removal, tokenization, POS tagging, lemmatization were performed. The pre-processing steps were implemented using NLTK and Stanford Tagger.

A typical way for feature engineering for IR is the TF–IDF, BM25 and PL2F models. The authors experimented on a set of different features. They did not adopt the feature set (i.e., lexical words, dependency pairs, and TF–IDF score) as it was proposed by Kim et al. (2015), but they considered the TF–IDF, Euclidean, Manhattan, Jaccard distances, as also the Latent Semantic Indexing (LSI), and Latent Dirichlet Allocation (LDA) using the GENSIM library. The model was applied at the paragraph level.

For the legal question-answering, the authors introduced a CNN -based model as it is depicted in Fig. 9. The binary classification model is summarized as follows:

- Both the question and the text of the most relevant articles were encoded through the network.
- The network was fed with 200-dimensional word embeddings. As previously, the word embeddings were trained by using WORD2VEC on the Japanese data law corpus. Ten convolutional filters of length two were applied. An average pooling layer of length 100 was then utilized to synthesize important features.
- For enhancing the results, two static features (i.e., TF–IDF and LSI) were concatenated at the output of the pooling layer feeding a two-layer MLP for predictions making.

In legal question-answering, the validation was performed on the 10% of the dataset. The CNN with additional features achieved the best performance. Particularly, the aforementioned model achieved an accuracy of 57.6%, while the CNN with LSI only achieved 54.5% and the CNN with TF–IDF achieved 53.0%. Employing only the CNN encodings achieved 51.5%.

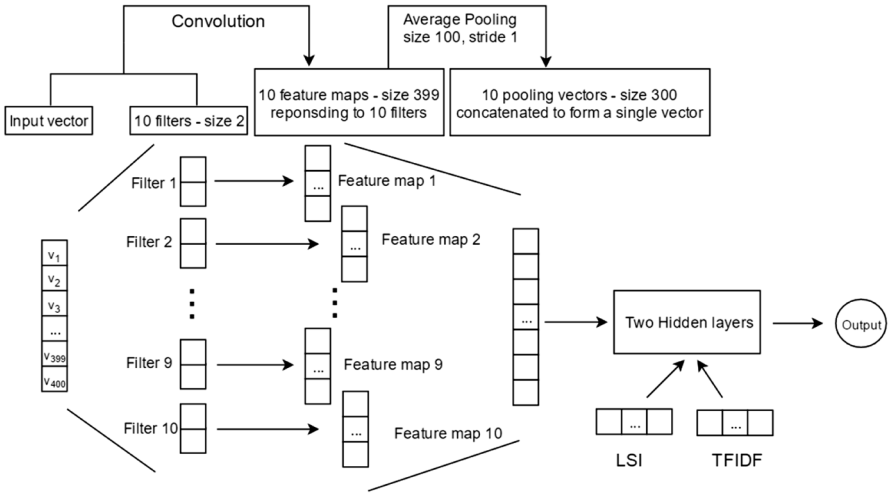

Fig. 9 A CNN model with additional features LSI and TF–IDF in Do et al. (2017)

3.3.5 Matching law cases and reference law provision with a neural attention model

Tang et al. (2016) tackled the problem of finding law provisions relevant to law cases. A similar task was introduced in COLIEE 2018, which is currently under review. The authors proposed a neural attention model for automatically matching reference law provisions. The complexity of the problem was due to the fact that each law case is related to many law provision references, based on the case specifics. Rule-based and keyword-based methods were not effective enough to capture the semantics of legal documents.

The authors proposed a binary classification method to match a law case with a law provision. The model is summarized as follows:

- The 50-dimensional word vectors used to represent the input words were trained from scratch, with an embedding layer as part of the neural network.
- Two independent bidirectional LSTM s with the same output dimensionality were used to build context-aware representations for the words of the case and the law provision.
- A word-by-word self-attention mechanism was used on top of each LSTM layer to produce the case and law provision representations (vectors).
- The Cosine and Euclidean distance were used to identify the element-wise and absolute distance of the vectors.
- An LSTM was fed with both vector distances and, at the last layer, a linear layer, and a log-softmax produced the output label.
- The training set consisted of the triple A_i, B_i, Y_i, where A represents the input case, B the law provision and Y the output. A and B were word sequences where each word was represented by its word embeddings. SGD with the backpropagation was used for the loss function optimization.

- Dropout method and l2-normalization were used on the output layer. The dropout rate was set to 0.2 and the l2 rate was set to three. The number of epochs was set to 200 and the training was performed on batches of 50.

The experimentation and evaluation of the proposed model were conducted with a real-world dataset collected from a set of Chinese credit fraud judgments. The positive sample was formed by the case briefs and reference law provisions. The training set consisted of 1000 case brief and law provision pairs, the validation set counts 2000 pairs and the test set 4000. The positive-negative ratio was balanced in all three datasets. The proposed model was compared to a SVM model and vanilla LSTMs. The proposed model outperformed the other two models by reaching an accuracy of 0.911.

3.3.6 A semi-supervised training method for semantic search of legal facts in Canadian immigration cases

Nejadgholi et al. (2017) introduced a semi-supervised approach to identify fact-asserting sentences from Canadian Immigration cases that are semantically close to a given query. The authors claim that in Canada, legal search engines rely solely on keyword matching, so they introduced a semantic search engine for users not familiar with the jargon used in legal documents. They outlined the challenges of matching fact sentences with different sentence types (e.g., reasoning). For example, a court may raise auxiliary points via hypothetical discussions that have nothing to do with the fact case itself. Matching sentences of different types is misleading due to the low ranking that would not correlate with cases predictive ability.

A single annotated dataset was used for training 100-dimensional word embeddings and a semi-supervised classifier that consisted of 46,000 immigration and refugee cases available on Canadas Federal and Supreme Court websites. After preprocessing, the cleaned corpus contained around 136M words, 4,5M sentences and a vocabulary size of 125,846 words. Considering the outcome of the domain-specific word embeddings, the authors emphasized the fact that they appear to be more appropriate to the immigration law. For example, the word 'immigration' is found to be close to the word 'FCJ' and 'IRPA' terms frequently used in Federal law citations. On the contrary, the same word appeared to be close to 'reform' and 'citizenship' into the general word embeddings of the Spacy library. Further on, the authors demonstrated word analogous with pairs such as (China—Chinese, Sri Lanka—Sri Lankan) and (Palestine—Hamas, Lebanon—Hezbollah), in order to highlight the importance of domain-specific pre-training.

The authors selected a semi-supervised approach using an annotated dataset of 12,220 annotated sentences from 150 cases. A binary classifier was trained to distinguish sentences between *facts* and *non-facts*. The classifier was a fully-connected shallow neural network with one hidden layer fed with the concatenation of word embeddings of each sentence. The trained classifier was then deployed over an unlabelled corpus of sentences, which includes a broader vocabulary, to create additional automatically annotated data in order to re-train the classifier and improve the sentence encoder. The trained classifier was finally used to label

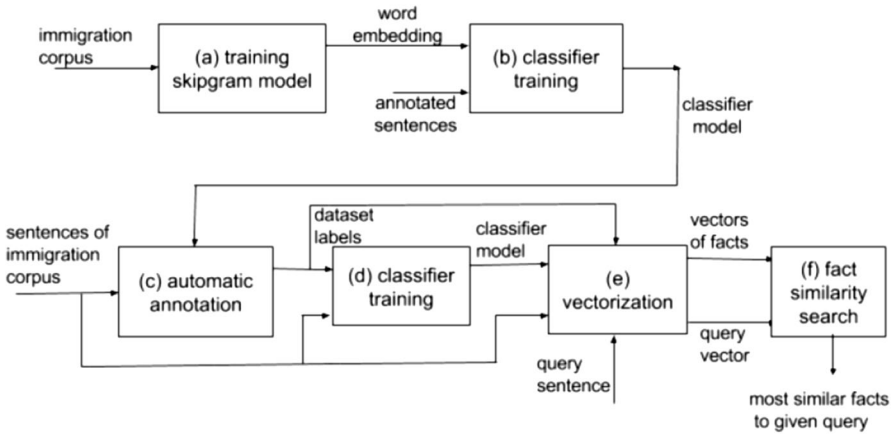

Fig. 10 Steps of the semi-supervised method presented in Nejadgholi et al. (2017)

all the sentences of the immigration corpus as *facts* and *non-facts*. The cosine similarity of a query and a fact sentence was calculated, based on the vector representation produced by the hidden layer of the trained classifier. For every single query, the top three most similar fact sentences were selected as relevant. The steps of the semi-supervised method were depicted in Fig. 10.

The evaluation of the results was carried out on a test set of 300 randomly selected sentences, where the labels were manually annotated by experts with 47% having been assigned with the *fact* label and the rest with the *non-fact*. The proposed method outperformed commonly used classifiers when the training set is relatively small. Particularly, the proposed model was significantly the most accurate classifier reaching an accuracy of 90%. In comparison, using SVM s with the TF–IDF weighted averaging of the domain-specific pre-trained embeddings outperformed the worse neural classifier (i.e., 84%). SVM s with TF–IDF feature representation and domain-specific word embeddings achieved 81% and 83% respectively. The accuracy of the proposed neural method was lower when the model used generic embeddings (accuracy: 86%) or randomly initialized embeddings (accuracy: 83%).

3.3.7 Discussion

As we observed in the articles of the current subsection, CNN-based models have been widely adopted in order to tackle information retrieval tasks. In the last 3 years, the research community is moving from intensive feature engineering towards more simplified networks that encode the text inputs by using standalone CNN s or combined with LSTM s. Researchers improve the performance by adding additional features produced by various methods (e.g., LDA, LSI, BM25 and well-known word distances). This development aligns with the current trends in information retrieval.

4 Conclusions

In this paper, we presented the early adaptation of Deep Learning in the legal domain. We discussed the concept of word embeddings and their importance in natural language processing. We highlighted methodologies tailored to legal analytics and we presented topic-wise the published works. We studied the related work from three main perspectives: (1) Text feature representation, (2) Neural Network architecture, and (3) the performance and outcomes. Furthermore, we enclosed significant observations derived from the results. Finally, we shared publicly domain-specific legal word embeddings, named Law2Vec, in order to support future experimentation for research purposes and beyond. We are planning to augment the legal corpora that were used with data coming from additional legal sources in order to broaden the semantic representation of the Law2Vec model.

References

Bojanowski P, Grave E, Joulin A, Mikolov T (2016) Enriching word vectors with subword information. arXiv preprint arXiv:160704606

Branting LK, Yeh A, Weiss B, Merkhofer E, Brown B (2017) Inducing predictive models for decision support in administrative adjudication. In: Proceedings of the MIREL Workshop on the The 16th International Conference on Artificial Intelligence and Law, London, UK

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

Chalkidis I, Androutsopoulos I, Michos A (2017) Extracting contract elements. In: Proceedings of The 16th International Conference on Artificial Intelligence and Law, London, UK, pp 19–28

Chalkidis I, Androutsopoulos I, Michos A (2018) Obligation and prohibition extraction using hierarchical rnns. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), Melbourne, Australia

Do PK, Nguyen HT, Tran CX, Nguyen MT, Nguyen ML (2017) Legal Question Answering using Ranking SVM and Deep Convolutional Neural Network. CoRR abs/1703.0. arXiv:1703.05320

Firth JR (1935) The technique of semantics. Trans Philos Soc 34(1):36–73

Goldberg Y (2017) Neural network methods in natural language processing. Morgan and Claypool Publishers, San Rafael

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, pp 328–339

Kim My, Xu Y, Goebel R (2015) A convolutional neural network in legal question answering. In: Ninth International Workshop on Juris-informatics (JURISIN)

Kudo T, Yamamoto K, Matsumoto Y (2004) Applying conditional random fields to Japanese morphological analysis. In: Proceedings of the Empirical Methods for Natural Language Processing, Barcelona, Spain, pp 230–237

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, Stateline, NV

Morimoto A, Kubo D, Sato M, Shindo H, Matsumoto Y (2017) Legal question answering system using neural attention. In: Proceedings of COLIEE - International Conference on Artificial Intelligence and Law, London, UK

Nanda R, John AK, Caro LD, Boella G, Robaldo L (2017) Legal information retrieval using topic clustering and neural networks. In: 4th competition on legal information extraction and entailment, 16th international conference on artificial intelligence and law vol. 47, pp 68–78

Nejadgholi I, Bougueng R, Witherspoon S (2017) A semi-supervised training method for semantic search of legal facts in Canadian immigration cases. Front Artif Intell Appl 302:125–134

Nguyen T, Nguyen L, Tojo S, Satoh K, Shimazu A (2017) Single and multiple layer BI-LSTM-CRF for recognizing requisite and effectuation parts in legal texts. In: 2nd ASAIL Workshop on 16th 16th international conference on artificial intelligence and law, London, UK

Nguyen T, Nguyen L, Tojo S, Satoh K, Shimazu A (2018) Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. Artif Intell Law 26(2):169–199

O'Neill J, Buitelaar P, Robin C, Brien LO (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, London, UK, pp 159–168

Pennington J, Socher R, Manning CD (2014) GloVe: Global vectors for word representation. In: Proceedings of the conference on empirical methods in natural language processing, Doha, Qatar, pp 1532–1543

Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Conference of NA chapter of the association for computational linguistics, New Orleans, Louisiana, USA

Tang G, Guo H, Guo Z, Xu S (2016) Matching law cases and reference law provision with a neural attention model. IBM China Research, Beijing

Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E (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, San Diego, CA, USA, pp 1480–1489