Weakly Supervised One-shot Classification using Recurrent Neural Networks with Attention: Application to Claim Acceptance Detection

Charles Condevaux, Sébastien Harispe, Stéphane Mussard, Guillaume Zambrano

September 2019

Weakly Supervised One-shot Classification using Recurrent Neural Networks with Attention: Application to Claim Acceptance Detection

Charles Condevaux [a,1] Sébastien Harispe [b] Stéphane Mussard [a] Guillaume Zambrano [a]
[a] *CHROME, Univ. Nîmes, France*
[b] *LGI2P, IMT Mines Alès, Univ. Montpellier, Alès, France*

Abstract. Determining if a claim is accepted given judge arguments is an important non-trivial task in court decisions analyses. Application of recent efficient machine learning techniques may however be inappropriate for tackling this problem since, in the Legal domain, labelled datasets are most often small, scarce and expensive. This paper presents a deep learning model and a methodology for solving such complex classification tasks with only few labelled examples. We show in particular that mixing one-shot learning with recurrent neural networks and an attention mechanism enables obtaining efficient models while preserving some form of interpretability and limiting potential overfit. Results obtained on several types of claims in French court decisions, using different vectorization processes, are presented.

Keywords. Classification, legal analysis, one-shot learning, deep learning.

1. Introduction

Text classification has long been identified as an important topic for Computer Science applications in the Legal domain [1]. A large diversity of applications can indeed be framed around classifying legal *entities* that are, or can be represented as, sequences of words, e.g. cases, decisions, claims, contracts. Interest for text classification has for instance be motivated by the recurrent need expressed by lawyers to find the most relevant cases according to specific contexts of interest [2] – text classification can indeed be used to structure case corpora by populating a predefined organization of cases that will further be used to improve case retrieval. Such classification techniques can also be used for filtering cases, and deciding whether it is relevant or not for a law firm to accept or reject a new case [3]. Recent applications for analyzing the impact of legal change through case classification have also been proposed [4]. Other examples of applications in the legal domain are: legal norms classification [5], detection of the semantic type of

[1] Corresponding author: charles.condevaux@unimes.fr. Granted by Région Occitanie: project PREMAT-TAJ.

legal sentences [6,7], detection of clause vagueness [8], or prediction of supreme court rule and the law area to which a case belongs to [9].

This paper presents our work on the definition of a deep learning model and a methodology for solving complex text classification tasks with only few labelled examples. The selected application is a general court decision classification setting applied on a French corpus of court decisions - court decision classification is an important non-trivial task in court decisions analyses. Court decisions have been labelled based on the acceptance of claims. In that context, we study in particular how mixing one-shot learning with recurrent neural networks and attention mechanisms in order to obtain efficient models.

The paper is organized as follows. Section 2 reviews some related works. Section 3 presents the model. Section 4 presents the datasets and the word vectorizations used in our experiment. Finally, Section 5 discusses the results.

2. Related works

Rule-based approaches and Machine Learning (ML) approaches are generally distinguished in text classification [5]. Rule-based classification systems rely on predefined domain expert rules, e.g. if the decision contains the utterance "*Article 700*" then label it with class `Damage`. The broad literature related to these approaches cannot be reduced to this simple example. Nevertheless, even if effective and relevant in specific cases[2], rule-based approaches *de facto* suffer from the need to express rules and to manage rule interactions for ensuring good performance. ML approaches may be used to overcome this limitation by implicitly inferring the decision rules of interest to drive efficient classification.[3] From a labelled dataset composed of numerous classification examples, learning algorithms are used for building predictive models. These approaches are today often preferred and have proven successful in numerous application contexts.

A large literature in Machine Learning, Natural Language Processing (NLP) and Computational Linguistics studies (text) classification. Among the most popular models widely used for text classification since the past two decades, we can cite: Naive Bayes Classifier, Logistic Regression, Random Forest, Support Vector Machine (SVM) and Multilayer Perceptron. These models require defining a vector representation of a text that will later be considered as model input. Specific and often *ad hoc* features of interest are sometimes used for building these representations – e.g. a boolean feature could be "*Does the text contains an utterance of* '*Article 700*?' ". To overcome the limitation of manually defining features, information related to the words composing the vocabulary used in the text corpora is often used as features. From simple bag-of-words and vector space models from the late 60s, to more refined weighting scheme modelling word relevance for classification, e.g. TF-IDF, these approaches have led to the definition of efficient models able to automatically solve interesting problems framed within text classification [10]. For instance, SVM have been used to perform legal norm classification with an accuracy of more that 90% for more than 13 different classes [5]. Using the same model, with TF-IDF vector representations, accuracy rates up to 94% have also

[2] Interpretability and the fact that these approaches do not rely on datasets may be interesting advantages.

[3] We focus on supervised ML, the traditional setting considered for text classification.

been obtained in a sentence classification task [11]. Despite these encouraging successes, more complex problems related to text classification are still out of reach.

The recent developments in Deep Learning have led to a fruitful diversity of radically new efficient neural network-based classification models, among which specific developments are of particular interest for text classification. Recurrent neural networks, such as Long Short-Term Memory (LSTM), are very useful for processing sequential data (e.g. such as texts, sequences of words) [12]. Embedding techniques have been developed and refined for encoding entities of interest, such as words, sentences, or texts, in law dimension spaces (e.g., BERT, ELMo, FastText) – these representations can next be used for classification or other ML pipelines [13,7]; note that these representations do not need to be defined explicitly through feature definition, as done in traditional Machine Learning approaches.

Attention mechanisms are also developed for better identifying and incorporating important information during the decision process. Technical aspects related to these approaches and techniques will later be introduced. They have led to very interesting performance improvements in various popular challenges offered to text classification [14,15,16]. Nevertheless, due to their intrinsic properties, deep learning models require large (labelled) datasets to be trained. This is an important issue for their use in the legal domain since it is most often difficult to mobilize experts in this domain, generally leading to data scarcity with only expensive and small labelled corpora available [4]. This limitation contributes to explaining the reduced amount of works on the use of Deep Learning for text classification in the legal domain. Active researches in ML focus on reducing the need of labelled data using (i) approaches to reuse models trained in related contexts (e.g., transfer learning, fine-tuning), (ii) by exploiting unlabeled data (e.g. via embeddings), or (iii) by exploiting as much as possible the information expressed in labelled data (e.g, one-shot learning, siamese neural networks).

Applying advanced deep learning techniques on small datasets is indeed possible given the right setup while avoiding overfitting. A strategy experimented in this paper is to implement one-shot learning aiming at solving classification tasks only using few examples [17]. This approach is today mainly used in computer vision [18] with memory-augmented networks [19] but can be adapted to NLP [20], or even to estimate word embeddings [21]. Instead of learning to directly map an input to an output class, the one-shot approach implemented using siamese networks aims at estimating a similarity function between pairs of observations [22]. This problem can be reduced to a binary classification task by setting a given label if both inputs are *similar* (i.e. share the same original label). Using such a discriminant approach, a model can be learned from a single example per class. However, since this task is non trivial in NLP due to the sequential aspect of language, entire datasets are generally used instead.

Applications and development such advanced deep learning techniques have to be encouraged in the legal domain in order to fully benefit from recent advances in Machine Learning. This paper presents how they can be used to classify judge decisions.

3. One-shot learning using a siamese recurrent network with attention

The proposed model implements one-shot learning using a siamese recurrent network with an attention mechanism to fine tune sentence representations. These embeddings are next reused jointly with selected features to solve a classification task.

3.1. General architecture

The general architecture of our model is presented in Figure 1.

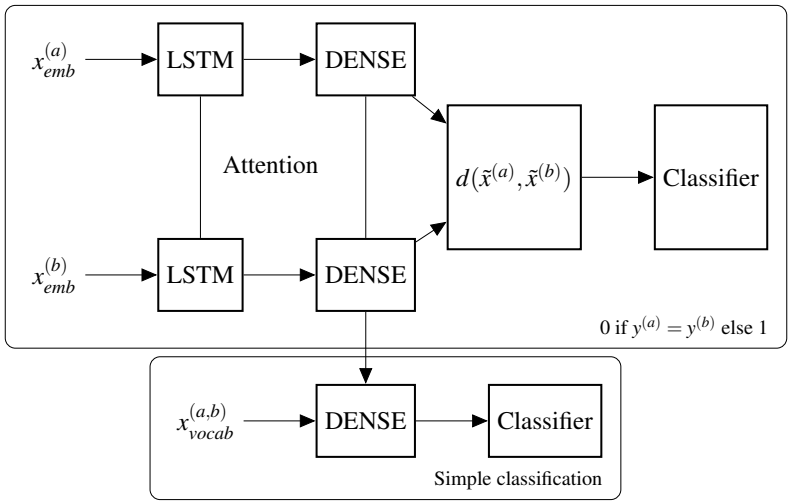

Figure 1. Proposed siamese network architecture

A siamese network composed of two symmetric sub-networks sharing the same weights but taking different inputs (sentences) is considered (see Section 4.2 for word representations). As we need to process sequences of words, the architecture relies on a bidirectional Long-Short Term Memory (LSTM) cell which takes pre-vectorized sentences as inputs ($x^{(a)}_{emb}$ and $x^{(b)}_{emb}$). In order for the network to focus on specific areas, an attention mechanism is added on top of the recurrent layer: to each part of the sentence is assigned a specific weight which denotes how important a word is, relative to other ones. These weights are then used to compute a weighted sum over the output of the LSTM, yielding a fixed 2-dimensional tensor encoding the sentence – this is defined in the literature has a many-to-one attention mechanism. Depending on the dataset and the setup, we use two different variants of this mechanism: the concatened version and the general Luong dot product attention [15] which rely on slightly different sub-networks for computation. A fully connected layer (dense) is also applied on the 2-dimensional tensor; this projection is later reused for the second classification task (bottom of Figure 1).

One-shot learning requires to compute a distance function between two samples that go through the siamese network. As we are using a binary cross entropy loss, the output is squashed using a dense layer with a sigmoid activation. Two distance functions ($d(\tilde{x}^{(a)}, \tilde{x}^{(b)})$) yield better performances given specific setups (details presented in Section 4). Those functions are applied feature-wise: the first one is based on the absolute difference between the two projections and the second one on a modified cosine distance. If

the weighted sum of distances is minimal, the sigmoid outputs a 0 and both samples will share the same label.

As datasets used for this task are very small (less than 100 labelled sentences), the network is compact with small layer sizes and dropout (0.25) to prevent overfitting. The LSTM has a (16 x 2) hidden size and the attention added on top is composed of 16 units.[4]

3.2. Classification

The top part of the network can be used independently for prediction. Given a new sample, it can be compared to known and precomputed samples from the training set. The same label as the closest or top closest sentences can then be associated. Using a second classifier however is often more reliable, stable and provides better performances.

The output of the dense layer preceding the distance function can be seen as a sentence embedding with a fixed size. This representation is transferred and concatenated with a selected set of discriminant words for this task. As classification is done directly on the original dataset (no couples involved), the number of features must be small to prevent overfitting. Word selection has been done comparing frequencies between classes. We employ a simple absolute distance metric which shows the best performances in our case. As we are in a binary case (the judge accepts or rejects a claim), frequencies have been defined as follows: with f_c^w the frequency of word w in sentences from category c, the discriminant words are those maximizing the difference $|f_i^w - f_j^w|$.

4. Datasets and words representations

In this section we describe the 5 datasets used for the classification task and the experiment setup. The preprocessing pipeline and the way specialized words embeddings are trained are also presented.

4.1. Datasets

Five datasets are chosen to cover different types of claims. They have been manually annotated by lawyers who labelled the part where the judge gives its argument for the specific claim and the result associated (accept or reject). This result is not straightforward to infer as French legal language has very specific vocabulary and expressions which are mostly unknown and extremely ambiguous for nonexpert people. The datasets are balanced and relative to name change requests (600.NOM, 74 observations), unpaid debts (600.DEC, 96 observations), lawyers' liability (500.RES, 400.RES, 100 observations each) and damage and interest claims for serious injuries (300.DOM, 98 observations).

4.2. Representation

Representing words and sentences is a challenging task and has a strong impact on classifiers performance. We compare different vectorization approaches from the simple TF-IDF to state-of-the-art models like BERT [23].

[4] This number is doubled when the data augmentation strategy later introduced is applied.

We built specialized word embeddings ranging from 32 up to 128 dimensions. All of them have been trained on a large corpus composed of 670 millions tokens from (French) court decisions and written laws. As French legal texts are very sensitive to case and punctuation (e.g. semicolons are important separators), these symbols have been kept. We estimate 3 different word embeddings: FastText [24], ELMo [25] and Flair [26]; BERT has however not been trained on our corpus as it requires massive computation power – the Bert-base multilingual cased pretrained model has been used instead.[5] On the one hand, all of them can handle out of vocabulary words and have specific characteristics: FastText and BERT use n-grams, Flair focuses on characters, while ELMo considers words and characters at the same time. On the other hand, FastText is static while all others are contextual, meaning they can change the word representation with respect to a specific context for disambiguation purpose. This is done by using recurrent layers or multi-head attention from the transformer architecture [16]. Training requires a few hours for FastText while ELMo and Flair need days to converge. As legal texts tend to have similar structures, a niche vocabulary and redundant expressions, very compact models can achieve low perplexity (< 20 for ELMo with 64 base dimensions) – this shows that French legal language is predictable.

4.3. Data augmentation

Data augmentation is a common way to deal with small datasets. Its main purpose is to artificially enrich and increase the number of observations (words) lying in the texts of the dataset. This is a challenging task in NLP as modifying one single word can drastically change the meaning of a sentence, which implies bias in the prediction over tiny samples. We investigate different approaches to see whether this technique can be done on legal language. First, words are randomly replaced by their synonyms using a thesaurus. This creates poor quality sentences as standard synonyms are not suitable for juridical specific vocabularies. Second, a random noise has been added on vectorized sentences with and without random word permutations. This does not yield any improvement, even leading to worse generalization capacities. Last, a translation tool is employed by going through several translations until returning to French language. This yields interesting new sentences *relevant* to the original ones. Augmenting data this way significantly improves the performance of the classifiers (see Section 5) allowing the models to be trained deeper while avoiding overfitting.

5. Experiment and results

We compare different approaches to find how fine tuning word embeddings and vocabulary selection can improve performances on small datasets (Figure 2). We start by investigating standard algorithms coupled with simple vectorization processes: the first one is based on TF-IDF while the second one relies on a selected vocabulary and a naive sentence embedding based on the average word representation (Table 1). We then show how fine-tuning using one-shot learning (Table 2) and data augmentation (Table 3) yield significant improvements. All results are averaged with 10-fold cross validation.

[5] https://github.com/google-research/bert

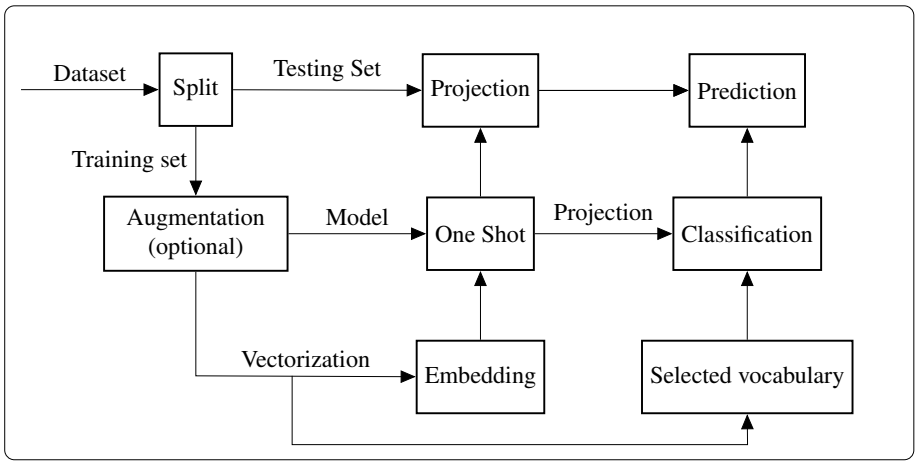

Figure 2. Train-test one-shot learning

5.1. Standard algorithms

	SVM			Random Forest			Logistic		
	P	R	F	P	R	F	P	R	F
600.NOM*	0.798	0.740	0.748	0.782	0.822	**0.786**	0.743	0.740	0.728
600.NOM**	0.820	0.820	0.781	0.867	0.780	**0.813**	0.752	0.748	0.739
600.DEC*	0.669	1.000	0.788	0.747	0.872	**0.795**	0.6578	0.96	0.767
600.DEC**	1.000	0.842	0.908	0.931	1.000	**0.959**	0.889	0.934	0.902
500.RES*	0.591	0.583	0.568	0.651	0.731	0.619	0.674	0.700	**0.645**
500.RES**	0.716	0.712	**0.659**	0.623	0.733	0.636	0.639	0.546	0.570
400.RES*	0.943	0.710	0.795	0.826	0.890	**0.837**	0.810	0.882	0.828
400.RES**	0.783	0.848	0.789	0.924	0.876	**0.893**	0.709	0.820	0.736
300.DOM*	0.827	0.923	0.834	0.847	0.888	**0.854**	0.847	0.903	0.825
300.DOM**	0.963	0.916	0.931	1.000	0.925	**0.952**	1.000	0.857	0.910

* TF-IDF vectorization — Precision (P), Recall (R) and F-measure (F)

** Selected vocabulary + mean embedding

Table 1. Comparing classification performances with different inputs

The random forest better performs on each demand category except for lawyers' liability (500.RES) for which SVM and logistic classifiers provide better F-measures. Words representations are averaged to provide a sentence embedding which is concatenated with the selected vocabulary. This yields significant gains compared with TF-IDF which

is obviously overfitting, for instance a F-measure gap of 0.164 is recorded on unpaid debts (600.DEC). TF-IDF shows poor performances for two main reasons: vocabulary is large and fixed, leading to a sparse representation; it is unable to handle variations consistently (e.g plural) unlike word embeddings. Selecting a subset of discriminant words often achieves similar performances with far less parameters and computation.

5.2. One-shot siamese recurrent network

The results of the one-shot siamese recurrent network are presented in Table 2. In this case, the mean embedding is replaced by the one-shot strategy which acts as a powerful sentence embedding model (fine-tuned weighted sum). The classifier outperforms the random forest on each demand category with the aid of ELMo. BERT shows lower performances as it has not been trained on a large legal corpus. Models are trained over embeddings with 32 dimensions, concatenated attention, and the ℓ_1 distance function.

	FastText			ELMo			Flair			BERT		
	P	R	F	P	R	F	P	R	F	P	R	F
600.NOM	0.842	0.810	0.818	0.867	0.830	**0.846**	0.842	0.850	0.843	0.755	0.923	0.789
600.DEC	0.945	1.000	0.967	0.975	1.000	0.986	0.986	1.000	**0.992**	0.986	0.983	0.983
500.RES	0.756	0.690	0.701	0.774	0.714	**0.734**	0.805	0.665	0.709	0.610	0.863	0.686
400.RES	0.868	0.916	0.882	0.907	0.921	**0.908**	0.828	0.901	0.854	0.921	0.843	0.872
300.DOM	1.000	1.000	**1.000**	1.000	0.983	0.990	0.983	0.983	0.982	0.963	0.983	0.971

Table 2. Comparing embeddings on one-shot classification task

These overall improvements come from the fact that we now rely on a model able to take advantage of the sequential aspect and long-short term dependencies (using LSTM and attention). This is fundamental as french legal language tends to be extremely ambiguous with double negatives, references, implicit reasoning...

	With augmentation			Without augmentation			Overall gains	
	P	R	F	P	R	F	ΔF*	ΔF**
600.NOM	0.870	0.960	**0.880**	0.867	0.830	0.846	+0.034	+0.094
600.DEC	0.986	1.000	**0.992**	0.986	1.000	**0.992**	+0.000	+0.197
500.RES	0.794	0.903	**0.817**	0.774	0.714	0.734	+0.083	+0.172
400.RES	0.918	0.969	**0.940**	0.907	0.921	0.908	+0.032	+0.107
300.DOM	1.000	1.000	**1.000**	1.000	1.000	**1.000**	+0.000	+0.146
Average							+0.030	+0.142

* F-measure difference with and without augmentation

** F-measure difference with augmentation and naive TF-IDF

Table 3. Best overall models with and without augmentation

Finally, Table 3 presents the contribution of the text augmentation. Improvements on 3 categories are observed. As we have access to more examples, we can deepen our model architecture by increasing layer sizes (they are all doubled) and using larger word embeddings (64 dimensions). Coupled with Luong attention and a cosine distance function, we achieve better generalization given the extra flexibility yields by more parameters. Further increasing embeddings size does not provide additional gain.

5.3. Attention for interpretability

Vocabulary selection provides a way to extract discriminant words but fails to take into account less frequent expressions or variations (e.g plural). Attention is a soft selection mechanism linking each input to a specific score given the context. As we feed words, we can find out which part of the sentence has high weights and where the network is focusing. The output of attention is a weighted sum over the temporal dimension, this leads to a more accurate and fine grained sentence embedding compared with a simple word average (see Table 1 and 2). Adding this mechanism also helps dealing with long term dependencies as it is insensitive to sequence length, even LSTM cells can suffer and forget large information parts from long sequences (> 30 words).

6. Conclusion

The one-shot siamese recurrent network proposed in this paper outperforms traditional algorithms of the literature for the purpose of predicting decisions outcome given highly ambiguous judge arguments. The results obtained with attention mechanisms as well as data augmentation seem to be promising; they illustrate how the Legal domain could benefit from advanced deep learning techniques suited for contexts in which only small labelled datasets are available. This work also opens the way on the employ of recent network architectures in jurimetrics such as adversarial networks, which provide some good potential to find discriminant words and expressions.

References

[1] Teresa Gonçalves and Paulo Quaresma. Is linguistic information relevant for the classification of legal texts? In *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, ICAIL '05, pages 168–176, New York, NY, USA, 2005. ACM.

[2] Stefanie Brüninghaus and Kevin D. Ashley. Toward adding knowledge to learning algorithms for indexing legal cases. In *Proceedings of the 7th International Conference on Artificial Intelligence and Law*, ICAIL '99, pages 9–17, New York, NY, USA, 1999. ACM.

[3] Robert Bevan, Alessandro Torrisi, Danushka Bollegala, Katie Atkinson, and Frans Coenen. Efficient and effective case reject-accept filtering: A study using machine learning. In *Proc. of the 31st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 171–175, 2018.

[4] Roos Slingerland, Alexander Boer, and Radboud Winkels. Analysing the impact of legal change through case classification. In *Proc. of the 31st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 121–1230, 2018.

[5] Bernhard Waltl, Johannes Muhr, Ingo Glaser, Elena Scepankova Georg Bonczek, and Florian Matthes. Classifying legal norms with active machine learning. In *Proc. of the 30st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 11– 20, 2017.

[6] Emile de Maat, Kai Krabben, and Radboud Winkels. Machine learning versus knowledge based classification of legal texts. In *Proc. of the 23th International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 87– 96, 2010.

[7] Ingo Glaser, Elena Scepankova, and Florian Matthes. Classifying semantic types of legal sentences: Portability of machine learning models. In *Proc. of the 31st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 121–1230, 2018.

[8] Giuseppe Contissa, Koen Docter, Francesca Lagioia, Marco Lippi, Hans-Wolfgang Micklitz, Przemyslaw Palka, Giovanni Sartor, and Paolo Torroni. Automated processing of privacy policies under the eu general data protection regulation. In *Proc. of the 31st International Conference on Legal Knoweledge and Information Systems (JURIX)*, pages 51–60, 2018.

[9] Octavia-Maria, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P. Dinu, and Josef van Genabith. Exploring the use of text classification in the legal domain. In *Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL)*, London, United Kingdom, 2017.

[10] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[11] Emile de Maat and Radboud Winkels. A next step towards automated modelling of sources of law. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 31–39, New York, NY, USA, 2009. ACM.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[13] Iosif Angelidis, Ilias Chalkidis, and Manolis Koubarakis. Named entity recognition, linking and generation for greek legislation. In *Legal Knowledge and Information Systems - JURIX 2018: The Thirty-first Annual Conference, Groningen, The Netherlands, 12-14 December 2018.*, pages 1–10, 2018.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Y Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

[15] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025, 2015.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[17] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594–611, April 2006.

[18] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. One-shot learning with memory-augmented neural networks. *CoRR*, abs/1605.06065, 2016.

[19] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1842–1850, 2016.

[20] Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[21] Andrew K. Lampinen and James L. McClelland. One-shot and few-shot learning of word embeddings. *CoRR*, abs/1710.10280, 2017.

[22] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems 6*, pages 737–744. Morgan-Kaufmann, 1994.

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[24] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[25] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018.

[26] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.