Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200859

Natural Language Processing Applications in Case-Law Text Publishing

Francesco TARASCONI [a,1], Milad BOTROS [a], Matteo CASERIO [a], Gianpiero SPORTELLI [a], Giuseppe GIACALONE [b], Carlotta UTTINI [b], Luca VIGNATI [b] and Fabrizio ZANETTA [b]

[a] *CELI Language Technology, Turin, Italy*
[b] *Giuffrè Francis Lefebvre, Milan, Italy*

Abstract. Processing case-law contents for electronic publishing purposes is a time-consuming activity that encompasses several sub-tasks and usually involves adding annotations to the original text. On the other hand, recent trends in Artificial Intelligence and Natural Language Processing enable the automatic and efficient analysis of big textual data. In this paper we present our Machine Learning solution to three specific business problems, regularly met by a real world Italian publisher in their day-to-day work: recognition of legal references in text spans, new content ranking by relevance, and text classification according to a given tree of topics. Different approaches based on BERT language model were experimented with, together with alternatives, typically based on Bag-of-Words. The optimal solution, deployed in a controlled production environment, was in two out of three cases based on fine-tuned BERT (for the extraction of legal references and text classification), while, in the case of relevance ranking, a Random Forest model, with hand-crafted features, was preferred. We will conclude by discussing the concrete impact, as perceived by the publisher, of the developed prototypes.

Keywords. natural language processing, applications, transfer learning, language models, text classification, information extraction, publishing industry, machine learning, BERT fine-tuning, random forest, Italian language

1. Introduction

Processing case-law contents, such as court judgements, for electronic publishing purposes is a time-consuming activity that encompasses several sub-tasks and usually involves adding annotations to the original text. Some operations, such as ranking new documents by their relevance, are required to determine which ones are worthy of publication. Other annotations are incorporated in products or services for the final customers, for example to facilitate search and exploration of related contents. Annotating legal texts requires specific knowledge, usually provided by domain experts or coded in a software component. On the other hand, recent trends in Artificial Intelligence and Natural Language Processing enable the automatic and efficient analysis of big textual data. These methods usually must be adapted for a specific domain. We will present our solution to three different business problems in the context of an Italian publisher of legal texts

[1] Corresponding Author: Francesco Tarasconi, CELI Language Technology, Via San Quintino 31, 10121 Turin, Italy; E-mail: tarasconi@celi.it.

and related products, in particular concerning the automatic annotation of Italian court judgements (mostly common or criminal law), originally provided in XML format :

1. **recognizing legal references**, distinguishing between references to legislation or to other judgements (Section 4);
2. **ranking by potential relevance** for the editors, to help assessing whether the content should be published or not (Section 5);
3. **labeling according to given topics**, described by a hierarchy of three classification levels, containing nodes such as "personal freedom" or "extortion" (Section 6).

For each problem, we developed a Machine Learning prototype that was deemed viable by the Business (i.e. the publisher's managers and decision-makers), and successfully deployed in a controlled production environment for inference on new data and further fine-tuning. The availability of high-quality training data, collected by the Business over the course of the years, enabled the successful experimentation of supervised methods. Before describing in detail the developed prototypes, we will summarize some previous work to better contextualize our research (Section 2); we will also provide essential details about the pre-existing annotation process of the publisher (Section 3). We conclude by discussing the business impact of the developed prototypes, together with their limitations and further work (Section 7).

2. Related work

A problem we will investigate in Section 4 is the automatic extraction of legal references, which can been solved without the help of Machine Learning through top-down approaches, as shown in [1] and [2]. However, our goal is also to classify different types of references according to their roles in the examined judgement (see [3] for a similar business case); we will frame the problem as a Named Entity Recognition one and solve it with Machine Learning methods, in order to better use context information and generalize. Named Entity Recognition for Italian language using Deep Learning is tackled with interesting results in [4]. Similar applications in role classification, that involve a Machine Learning approach, can be found in [5]. Text classification methods are within the scope of our research in Section 5 (binary classification problem) and Section 6 (multi-class); they have been successfully applied to a number of use cases ranging from plagiarism [6] to estimating the period in which a text was published [7]. Overall, Machine Learning overcomes the limits of manually compiling classification rules, when enough training data are available. Successful experiments in predicting law areas from text, using the Support Vector Machine model class, are described in [8]. Deep Learning approaches for the legal domain, using Convolutional Neural Networks, are described in [9]. More context to the problem of Extreme multi-label text classification (XMTC) and relative applications of Deep Learning techniques is provided in [10]. A larger amount of training examples was traditionally required in order to reach satisfying results through Deep Learning. Human-labeled data, domain-specific, are still necessary to conduct successful experiments, but in smaller amounts, thanks to transfer learning and pre-trained language models. One the most effective architectures developed over the last few years is Google BERT [11], a transformer model that leverages upon the self-attention mechanism. BERT can be fine-tuned for specific tasks such as Named Entity Recognition and text classification. Chalkidis and Kampas [12] noted that self-attention does not only lead to performance improvements in legal text classification, but might also provide useful evidence for the predictions. However, Deep Learning models can be computationally expensive and sometimes the apparent performance gain over other Machine Learning methods is negligible or spurious, as discussed for example in [13]. NLP–based metadata extraction for Italian legal texts is described in [14] and [15], but they are focused on the legislative act life-cycle and consolidation.

3. Business Context: A Real-World Publishing Process of Legal Texts

We will briefly describe in this section the business context where our research took place, in particular the electronic publishing workflow where NLP was applied. We won't provide information on other operations that are outside the scope of these applications. The original contents are judgements released by Italian courts and, after a pre-publishing phase, provided in XML format (*documents*). Each document is assigned a unique *ID* and stored in a database with its metadata, such as an identifier of the corresponding source, called *Authority*, and the *Date* of the judgement. XML documents are divided in three different sections: an introductory *Preamble* providing contextual information to the judgement; a main part containing factual and legal information (called *FactsLaw*); a final part containing the verdict (called *PQM*, acronym for the Italian expression ”per questi motivi”, meaning ”for these reasons”). Each section is further divided into *Paragraphs*, of variable length (from hundreds to thousands of characters).
The following Steps are performed on each document, enriching the original XML:

1. **extraction of legal references**: contiguous spans within the same Paragraph, that contain a reference, are tagged. Prior to this work, it was accomplished through top-down rules and regular expressions. See Section 4 for more details;
2. **linking of legal references**: hyperlinks to external documents are added, containing the judgement or legislation mentioned in the text. This is accomplished through a custom search engine that is outside the scope of this paper;
3. **relevance classification**: documents are labeled as relevant or irrelevant. Relevant ones are considered for further editing and publication. This operation is historically performed by domain experts and content curators. See Section 5 for more details;
4. **topic classification**: each relevant document is labeled by domain experts, according to what the examined judgement is about and a pre-existing topic tree. See Section 6;
5. **holding formulation**: one or more holdings are compiled by domain experts, summarizing the law principles expressed in the judgement. Through adoption of attention-based models, this task is related to the topic classification one step and briefly discussed in Section 6.
6. **reference classification**: references to other judgements that were previously extracted are classified by domain experts as ”according to” / ”different from” / ”related to”, based on the relation between the two verdicts; errors in reference extraction are also manually corrected.

Topics, holdings and legal references form the backbone of several of the publisher's electronic products, for attorneys and other Law professionals. Given the current state-of-the-art, outlined in Section 2, A.I. potential and limitations, the following best practices were agreed upon with the Business:

(i) to carefully frame the use cases/business problems;
(ii) to identify meaningful datasets for Machine Learning model development, together with the appropriate error metrics;
(iii) to evaluate different models according to chosen metrics, and also in terms of computational cost and explainability, so that an informed decision can be taken by the Business;

(iv) to perform error analysis of each prototype, educating the Business on the limits of A.I. and understanding where the human must intervene.

4. Application: Recognition of Legal References

Our goal is to identify in a judgement all the spans of text that refer to a specific law or to another judgement. References to other judgements must also be classified as ”according to” / ”different from” / ”related to” the examined judgement. Developing a single Machine Learning system that performs both operations allows to automate Steps 1 and 6 described in Section 3. This simple distinction between reference roles is used downstream in several publishing products.

4.1. Methodology

The proposed solution is based on a fine-tuned version of multi-language BERT[2] for Named Entity Recognition [11]. Our setup is similar to the one for Portuguese language described in [16], but we do not use the CRF layer that is described in the paper. The final layer performs token-level classification with one predicted class among the following target list, defined in manner consistent with common IOB practices in NER [4]:

1. O: the token is outside / not part of a reference;
2. B-L: the token is the beginning of a legislative reference e.g. to a specific law article;
3. B-J-ACC: the token is the beginning of a reference to a judgement, that is in accordance with the examined judgement;
4. B-J-DIF: as B-J-ACC, but the referred verdict was different from the examined one;
5. B-J-REL: as B-J-ACC, but the two judgements are simply related; from a legal standpoint, it’s a weaker relation compared to B-J-ACC and B-J-DIF;
6. I-R: the token is the continuation of a reference (any kind).

The chosen metrics to evaluate the system, agreed upon with the Business, are the F1-Scores of ”proper” reference classes, excluding the O class from the list above.
Original input comes in the form of XML Paragraphs where free text references (i.e. spans of text) are tagged accordingly. Through a custom version of the standard BERT *wordpiece* tokenizer, a preprocessing phase prepares each Paragraph for analysis, associating target classes to BERT tokens, and removing all XML markup. Data are split in a Training Set (70%), Development Set (15%) and Test Set (15%). BERT fine-tuning is conducted by adding a final feedforward layer with softmax, and minimizing cross-entropy loss function over training data. Development data are used to perform model evaluation and selection by maximizing the weighted average of F1-Scores, calculated over all target classes, barring the O class. A postprocessing function, used for integration with the publisher’s pipeline, is made available for re-aligning BERT output to the original text. At the moment of inference on new documents, all Paragraphs are classified separately, in conformity with model training.

Implementation Details. The described methodology was implemented using TensorFlow 1.12, in particular the *estimator* API for training, evaluation, prediction and export for serving [17].

[2] BERT original code from: `https://tfhub.dev/google/bert_multi_cased_L-12_H-768_A-12/1`

4.2. Prototype Data

When our research started, the publisher's information concerning the type of reference to other judgements (necessary to discriminate between B-J-ACC, B-J-DIF, B-J-REL classes) was not available at the level of text spans, but stored only at document level. Therefore, domain experts were involved to further annotate, add the precise classes to text spans, and provide the required input. For this reason, only a small subset of the publisher's documents could be used, for the development of this application. We worked on criminal and common law judgements of the Italian Highest Courts of Appeal. The resulting dataset is composed of 6,133 Paragraphs from 150 documents, with 13,657 total references.

4.3. Results

Table 1. Breakdown of Test error metrics for fine-tuned BERT model in legal reference recognition.

Type	Test Cases	Precision	Recall	F1-Score
B-L	692	0.940	0.957	0.948
B-J-ACC	77	0.535	0.494	0.514
B-J-DIF	15	0.200	1.000	0.333
B-J-REL	776	0.883	0.930	0.906
I-R	16,118	0.969	0.985	0.977

Breakdown of performance on Test Set is reported in **Table 1**. The system achieved a weighted F1-Score on classes of interest of 0.970 (including continuations I-R), 0.900 (counting only beginnings of references B).

Error Analysis. Several errors were in delimiting text spans containing references, exactly as the original data, but the model proposals were found to be often acceptable as well. Only in 6 cases serious errors were committed: confusing laws with judgements, or B-J-ACC references with B-J-DIF. Despite lower performances on less frequent classes, the prototype was considered viable by the Business, given also the partially subjective nature of the task; more experiments will be conducted with additional data.

Other Experiments. Different setups, for solving the problem with BERT, were experimented with, such as breaking down the problem into related subtasks (e.g. distinguishing B-L and B-J, plus distinguishing between B-J-ACC, B-J-DIF and B-J-REL). These approaches yielded slightly lower performances (between 0.01 and 0.02 drop in weighted F1-Score) and found more difficult to correctly assign the less frequent labels. Other experiments, without pre-training for the Italian language (e.g. analyzing windows of texts as shown in [4]), saw a larger performance drop, especially in discriminating between B-J-ACC, B-J-DIF and B-J-REL.

5. Application: Ranking by Relevance

The goal of this application is to identify the potential relevance of documents, in order to select the ones that will be annotated further and eventually published (see Step 3 in Section 3). A model that formulates such predictions should implement, explicitly or implicitly, the criteria employed by humans; a supervised approach, based upon pre-classifed relevant documents, seems therefore promising . Because the output of Machine Learning models can usually be expressed as a probability or a score, our idea, agreed upon with the Business, was to provide the end-user with a ranking of documents, to review model suggestions in order of relevance.

5.1. Methodology

Our solution is based on a Random Forest model [18] that uses hand-crafted features, defined together with the editors, and is trained on a binary classification problem, to distinguish between relevant and irrelevant documents. The probability of belonging to the relevant class is provided as output and it's used as relevance ranking. The features are:

a) number of references to legislation (see Section 4) in the document;
b) number of references to other judgements (see Section 4) in the document;
c) length (number of characters) of FactsLaw XML section (see Section 3), after removing XML markup;
d) number of legal quotes, delimited by quotation marks and containing more than one word;
e) binary features corresponding to presence or absence or specific expressions in the PQM XML section.

Coding these features involves an NLP preprocessing step, not only to remove XML markup, but also to perform lemmatization and be able to match variants of the original expression, e.g. "**declares** the **appeals** inadmissible" should match the given expression "**declare** the **appeal** inadmissible".

Data are split in a Training Set (60%), Development Set (20%) and Test Set (20%). A grid search is performed in order to maximize the weighted F1-Score on the development set and identify the optimal number of estimators, minimum samples in each leaf and maximum depth of each tree. According to the importance of listed variables in the resulting model, calculated through permutations [18], they are all useful to the task.

Implementation Details. The procedure was coded in Python and implemented using Scikit Learn 0.22.1 [19].

5.2. Prototype Data

The dataset, that was determined in accordance with the Business, represents a sample of stored data from all the Authorities which are currently managed. The dataset is composed of 4,958 documents: 64% relevant and 36% irrelevant. It is largely composed (70%) of judgements from the Highest Courts of Appeal (criminal and common law), but also contains documents from the T.A.R. Administrative Regional Tribunal (5%), Italian Constitutional Court (4%) and E.U. courts (4%). Remaining documents come from other Italian courts. Irrelevant documents are likely to be *more frequent* in the real-world execution of this task, as not all the historical ones were stored and available. At the same time, it was not possible to determine an average distribution of "relevant vs irrelevant" documents. This fact will be considered in analyzing the performance of the optimal solution; strong bias towards the relevant class should be avoided.

Finally, working on this dataset, through Machine Learning methods, allowed us to find human mistakes in the original classification.

5.3. Results

Table 2. Breakdown of Test error metrics for Random Forest model in relevance classification.

Class	Precision	Recall	F1-Score
Relevant	0.84	0.90	0.87
Irrelevant	0.81	0.70	0.75
Weighted average	0.83	0.83	0.83

The model achieves a weighted F1-Score of 0.83 in Test. Breakdown between relevant and irrelevant classes is reported in **Table** 2.

As we have seen in the *Prototype Data* Subsection, irrelevant data are likely underrepresented in our dataset, so it's important that the performance on the irrelevant class is checked carefully, as its weight in the real-world application is higher. We will evaluate further fine-tuning of the model and re-balancing of the training data, as information from the production environment is collected.

Error Analysis. Human analysis of 50 errors showed that, in 64% of cases (32 documents), the model picked the wrong class, but in a borderline situation; several irrelevant documents were considered ”acceptable” (i.e. relevant) by some of the domain experts. The remaining 18 documents, actual mistakes, had a lower ranking associated with them, indicating lower model confidence. There were cases, difficult to treat with this approach, where a judgement was labeled as ”irrelevant”, because the annotator knew pertained a topic, well covered by the publisher, and with very similar judgements already analyzed.

Other Experiments. A single Classification Tree, based upon the same features, achieved a weighted F1-Score of 0.78 on the same task. Adding features, based on frequent words or specific references, found in the document, didn't improve the performance of Random Forest or Classification Tree models.

An implementation of BERT for binary classification of judgements, similar to the one described in Section 6, was used to test an approach entirely based on free text analysis, and achieved a weighted F1-Score of 0.75.

6. Application: Classification by Topic

Our goal is to label each document as related to none, one or more topics. Topics belong to a proprietary resource of the publisher's: a classification tree of three levels, with 12,066 nodes. The majority of documents (75%) are associated to a single topic; more than 99% documents possess between 1 and 5 labels.

After conducting an exploratory analysis, the original problem was transformed in a more tractable one; for what concerns the prototype, object of this research, target topics must possess a minimum number $F = 200$ of training examples. In case a node is discarded because of its frequency, lower than F, documents belonging to that node are assigned to the parent node (corresponding to a more generic topic), if possible.

This restriction allowed us to build a working prototype and show its usefulness to the Business. Adding data, reducing F and managing more topics will be treated as further evolution of the developed system.

6.1. Methodology

The proposed solution is based on a fine-tuned version of multi-language BERT [11] for multi-label text classification. Our setup is similar to the one proposed in [20] for multi-label text classification on EU Legislation and we exploit the multi-label attention mechanism through an architecture similar to the one described in [21]. The main obstacle in adapting BERT to this application is the limitation of the length of documents that the model can analyze (512 tokens). We fix a constant N, and, for each document, N different Paragraphs are randomly sampled from the FactsLaw XML section and processed individually through the attention layers. The N different outputs from these layers are combined to produce a unified document representation, passed to the final fully connected (and output) layer. Random sampling is more effective, on this dataset, than considering the first N Paragraphs. Data are split in a Training Set (80%) and a Test Set (20%). Fine-tuning is conducted on Training data, by minimizing sigmoid cross-entropy loss function.

Output is provided in two formats: all labels with score > 0.5 or the top K labels, regardless of their minimum probability. While the first format is used to evaluate and compare different models through F1-Scores and their weighted average, the second format is used in production environment for end-users (domain experts and editors), when performing inference on new data.

Implementation Details. The described approach was implemented in the same framework employed in Section 4, using TensorFlow 1.12. N was fixed at 40 for computational reasons. K was fixed at 5 after evaluating the prototype's performance.

6.2. Prototype Data

The dataset is composed of 44,413 documents from the Highest Courts of Appeal (Criminal and Common Law), collected by the publisher over the last five years.

After a preliminary analysis, having fixed F at 200, 81 topics were considered during development. In spite of considering a small subset of the full classification tree, 64% of documents have at least one valid (i.e. frequent) topic associated. The most frequent topic is *contracts and obligations*, with 1,248 examples.

6.3. Results

The described solution achieves a weighted F1-Score of 0.505 over the 81 examined Topics. It was verified that the correct (i.e. originally assigned by human) labels are found 90% of the times in the first 5 predictions.

The output of attention layers, as suggested in [12], is currently being examined by domain experts to assess its usefulness in highlighting the most important Paragraphs and in the holding definition phase (Step 5 of Section 3).

Error Analysis. Examining the top K predictions for some documents, domain experts verified that they are usually related and that there was in fact a certain degree of freedom in choosing the original classification itself.

Other Experiments. The best performing Bag-of-Words, no pre-training, experiment, was an XGBoost ensemble model [22], using a combination of frequent words and frequent legislation references as features. It achieved a weighted F1-Score of 0.370.

7. Conclusions and Future Work

We have first introduced the annotation process of court judgements by a real-world Italian publisher, highlighting areas where amount of human effort and availability of training data motivated the experimentation of Machine Learning automatic approaches. We then described the developed solutions to three specific problems, showing how Natural Language Processing could in fact reach satisfying performances where training data was sufficient. Employing a model architecture based on BERT, fine-tuned for the specific tasks of Named Entity Recognition and Extreme Multi-label Text Classification, provided the best results in the most complex problems, where free text understanding was crucial. In the case of ranking by relevance, the importance of hand-crafted features (in capturing the differences between relevant and irrelevant documents) explains why a simpler, faster Random Forest model obtained better results and was chosen for deployment.

7.1. Business Impact

Working on the described prototypes required several skills, ranging from Natural Language Processing development to in-depth knowledge of the legal domain for problem framing, data selection and error analysis. The resulting team-mix was deemed successful and can be adopted in new projects. Communication between the Business and the developers was constant during the research and effective: the added value of Deep Learning was shared and understood, not taken for granted. The developed prototypes are performing inference on a subset of new real-world data, in a controlled production environment, before further fine-tuning and integration. The current integration model is asynchronous and employs Apache Kafka (`kafka.apache.org`) for handling data feeds. Each Machine Learning module is exposed as a synchronous RESTful Service. A JSON data exchange format was agreed for integration in the rest of the publishing pipeline. This system currently helps the editors and reduces the amount of human effort by pre-annotating documents which can then be reviewed more quickly by the domain expert. The model for relevance ranking mirrors closely human decision-making and actually allows to correct some mistakes in the original classification.

7.2. Limits and Further Developments

The models for extracting legal references and topic classification will require new cycles of annotated data gathering, training and test, in order to increment the coverage of less frequent classes. Instead, the main limit of ranking by relevance is its being based upon intrinsic features of the documents. Adding features based on the similarity to previous judgements could help in dealing with particular or difficult cases.

Once the users have acquired trust in the system and the machine behavior mirrors more closely the human's in edge cases, a deeper integration in the publishing process will be possible. To this end, advances in zero-shot learning should also be followed closely and tested. Finally, monitoring how these modules work on new data and carefully reviewing user's feedback will help in identifying unknown issues and making the solution more robust over time.

References

[1] Agnoloni T, Bacci L, Peruginelli G, van Opijnen M, van den Oever J, Palmirani M, Cervone L, Bujor O, Lecuona AA, García AB, Di Caro L, Siragusa S. Linking European Case Law: BO-ECLI Parser, an Open Framework for the Automatic Extraction of Legal Links. In: Frontiers in Artificial Intelligence and Applications. Volume 302: Legal Knowledge and Information Systems. 2017. Pages 113-118.

[2] Gheewala A, Turner C, Maistre JR. Extraction of Legal Citations using Natural Language Processing. In: Proceedings of the 15th International Conference on Web Information Systems and Technologies. 2019. Pages 202-209.

[3] Winkels R, Boer A, de Maat E, van Engers T, Breebaart M, Melger H. Constructing a semantic network for legal content. In: Gardner, A (ed.) Proceedings of the Tenth International Conference on Artificial Intelligence and Law (ICAIL 2005). ACM Press, New York (2005). Pages 125-140.

[4] Bonadiman D, Severyn A, Moschitti A. Deep Neural Networks for Named Entity Recognition in Italian. Italian Conference on Computational Linguistics (CLiC it). 2015.

[5] Winkels R, Boer A, Vredebregt B, Someren, A. Towards a Legal Recommender System. JURIX. 2014.

[6] Barrón-Cedeño A, Vila M, Martí MA, Rosso P. Plagiarism meets paraphrasing: Insights for the next generation in automatic plagiarism detection. In: Computational Linguistics 39, 4 (2013). Pages 917–947.

[7] Niculae V, Zampieri M, Dinu L, Ciobanu AM. Temporal Text Ranking and Automatic Dating of Texts. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers. Association for Computational Linguistics. 2014. Pages 17–21.

[8] Şulea OM, Zampieri M, Malmasi S, Vela M, Dinu LP, van Genabith J. 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.

[9] Wei F, Qin H, Ye S, Zhao H. Empirical Study of Deep Learning for Text Classification in Legal document Review. 2018 IEEE International Conference on Big Data (Big Data). Seattle, WA, USA. 2018. Pages 3317-3320.

[10] Liu J, Chang WC, Wu Y, Yang Y. Deep Learning for Extreme Multi-label Text Classification. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17). Association for Computing Machinery, New York, NY, USA. 2017. Pages: 115–124.

[11] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. Association for Computational Linguistics. 2019. Pages 4171–4186.

[12] Chalkidis I, Kampas D. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. In: Artificial Intelligence and Law 27. 2019. Pages 171–198.

[13] Niven T, Kao HY. Probing Neural Network Comprehension of Natural Language Arguments. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. Pages 4658-4664.

[14] Bolioli A, Dini L, Mercatali P, Romano F. For the Automated Mark-Up of Italian Legislative Texts in XML. Legal Knowledge and Information Systems. JURIX. 2002.

[15] Spinosa P, Giardiello G, Cherubini M, Marchi S, Venturi G, Montemagni S. NLP-based metadata extraction for legal text consolidation. In: The 12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference. 2009. Pages 40-49.

[16] Souza F, Nogueira R, Lotufo R. Portuguese Named Entity Recognition using BERT-CRF. Preprint on arxiv.org. Last revised 27 Feb 2020.

[17] Abadi M et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. White Paper. Software available from tensorflow.org.

[18] Breiman, L. Random Forests. In: Machine Learning 45. 2001. Pages 5–32.

[19] Pedregosa F et al. Scikit-learn: Machine Learning in Python, JMLR 12. 2011. Pages 2825-2830.

[20] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Ion Androutsopoulos Large-Scale Multi-Label Text Classification on EU Legislation In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. Pages: 6314–6322.

[21] You R, Zhang Z, Wang Z, Dai S, Mamitsuka H, Zhu S. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada. 2019.

[22] Friedman J. Greedy Function Approximation: A Gradient Boosting Machine. In: The Annals of Statistics, Vol. 29, No. 5. 2001.