

# Exploring domain and task adaptation of LamBERTa models for article retrieval on the Italian Civil Code

Andrea Simeri<sup>1</sup>, Andrea Tagarelli<sup>1,\*</sup>

<sup>1</sup>*Dept. Computer Engineering, Modeling, Electronics, and Systems Engineering (DIMES), University of Calabria, 87036 Rende (CS), Italy*

## Abstract

This paper is concerned with AI-based NLP solutions to the law article retrieval problem, with application to the Italian legal domain and, particularly, to the Italian Civil Code. Based upon the current state-of-the-art on this topic, we revise our early LamBERTa framework in a twofold way relating its domain-adaptation feature: replacing the general-domain pre-trained model with a legal-specific one to fine-tune for the task of article retrieval, and delving into the injection of out-of-vocabulary legal terms into the models' tokenizer. Extensive experimental evaluation based on different collections of query sets, along with qualitative analysis on the models' prediction interpretability, have unveiled interesting findings about the combined effect of domain- and task-adaptation of an Italian BERT model on the task of law article retrieval.

## Keywords

law article retrieval, domain adaptation, legal language models, artificial intelligence and law

## 1. Introduction

Artificial Intelligence (AI) is increasingly used in the legal domain, which finds main motivations in the huge amount of information produced and in the involvement of different actors, such as legal professionals, law courts, legislators, law firms, and even citizens [1].

Starting with BERT [2], deep contextualized *pre-trained language models* (PLMs) have emerged in the NLP field showing outstanding performance in several discriminative and generative tasks. BERT and BERT-like models have also represented a breakthrough for the legal domain, especially concerning classification problems (e.g., [3, 4, 5, 6, 7, 8, 9]).

Early applications of such models to the legal domain include approaches that make PLMs *adaptive* to a specific legal data analysis task, i.e., they directly fine-tune a general-domain pre-trained model to the task at hand. In contrast to such task-adaptive methods, *domain-adaptive pre-training* allows for deeply tailoring a pre-trained model to the domain of the target task [10]. To specialize a pre-trained model on the legal domain, there are two main strategies that stand as alternative to the direct application of an out-of-the-box pre-trained model for the downstream task, namely either to continue pre-training the model on a legal corpus, or to pre-train the model from scratch on a legal corpus.

---

19th IRCDL (*The Conference on Information and Research science Connecting to Digital and Library science*), February 23–24, 2023, Bari, Italy

\*Corresponding author.

✉ [andrea.simeri@dimes.unical.it](mailto:andrea.simeri@dimes.unical.it) (A. Simeri); [andrea.tagarelli@unical.it](mailto:andrea.tagarelli@unical.it) (A. Tagarelli)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

Our study in this paper concerns the above topic contextualized to the Italian legal domain. In this respect, it should be noted that, despite a number of Italian BERT models exist (e.g., [11, 12, 13]), they mostly refer to general-domain language. In particular, no study leveraging BERT for the Italian civil law has been proposed until LambBERTa [14], the first BERT-based framework for law article retrieval as a prediction problem. LambBERTa is in fact designed to learn prediction models by fine-tuning an Italian pre-trained BERT on the Italian Civil Code (ICC), and to answer natural language queries by retrieving the most relevant ICC articles. Much more recently, a new contribution to the Italian legal domain has been offered by the release of the first Italian BERT pre-trained on legal corpora, named ITALIAN-LEGAL-BERT [15].

Given this premise, in this paper we aim to answer the following research questions:

- **RQ1:** How does the behavior of LambBERTa models change when fine-tuning a legal Italian BERT rather than a general-domain Italian BERT?
- **RQ2:** What is the impact of injecting out-of-vocabulary legal terms into LambBERTa models during the fine-tuning stage? Does it depend on how such terms' representation is initialized?
- **RQ3:** What aspects arise from the explanation of the different LambBERTa models through the interpretation of their predictions?
- **RQ4:** Overall, is the combined effect of domain-adaptation and task-adaptation of a pre-trained Italian BERT model helpful to improve performance on the task of article retrieval from the Italian Civil Code?

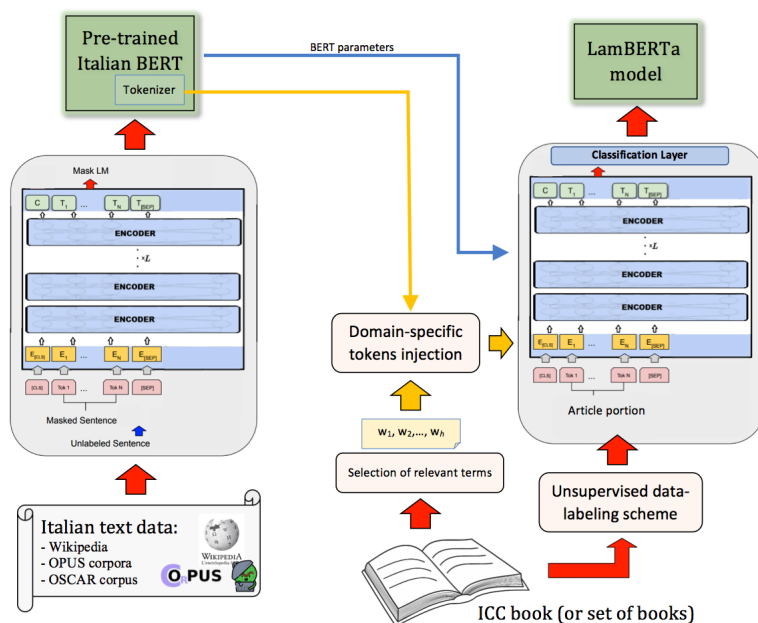
To answer the above questions, we provide the following main contributions. We advance research on AI-based NLP for the Italian legal domain by updating the current state-of-the-art of PLMs for law article retrieval as a prediction task. Starting over our early LambBERTa framework, we develop a new variant of LambBERTa, which makes it domain-adaptive besides task-adaptive; we accomplish this by designing LambBERTa so as to learn ICC article classification models through a fine-tuning of an Italian legal pre-trained BERT on the ICC (Section 4). We further investigate on the domain-adaptation of LambBERTa models by gaining insights into the effect of injecting into them a few domain-specific terms, selected from the target legal corpus, and previously unseen in the pre-trained model's vocabulary (Section 5). Moreover, we perform a qualitative analysis of the different LambBERTa models by explaining their underlying behaviors on a number of query instances (Section 6). We finally provide a discussion on our main findings that were drawn for our LambBERTa model variants based on an extensive collection of query sets at varying degrees of length and lexical complexity (Section 7).

## 2. Background

In this section, we provide background concepts on the Italian Civil Code, the LambBERTa framework [14], and ITALIAN-LEGAL-BERT [15].

### 2.1. The Italian Civil Code

The Italian Civil Code (ICC) is divided into six books, each of which provides rules for a specific theme in civil law. *Book-1* (on Persons and the Family, articles 1-455) contains the discipline of



**Figure 1:** An illustration of the conceptual architecture of the original LamBERTa framework

the juridical capacity of persons, of the rights of the personality, of collective organizations, of the family; *Book-2* (on Successions, articles 456-809) contains the discipline of succession due to death and the donation contract; *Book-3* (on Property, articles 810-1172) contains the discipline of ownership and other real rights; *Book-4* (on Obligations, articles 1173-2059) contains the discipline of obligations and their sources, that is mainly of contracts and illicit facts (civil liability); *Book-5* (on Labor, articles 2060-2642) contains the discipline of the company in general, of subordinate and self-employed work, of profit-making companies and of competition; *Book-6* (on the Protection of Rights, articles 2643-2969) contains the discipline of the transcription, of the proofs, of the debtor's financial liability and of the causes of pre-emption, of the prescription.

For an analysis of the ICC article citation network and relating visualization tool, the interested reader might refer to [16] and [17].

## 2.2. The LamBERTa framework

Figure 1 shows the conceptual architecture of LamBERTa [14, 18]. The starting point is ITALIAN-XXL-UNCASED, a pre-trained Italian BERT model whose data source consists of a large Wikipedia dump, various texts from the OPUS corpora collection, and the Italian part of the OSCAR corpus; the final training corpus has a size of 81GB and 13 138 379 147 tokens.<sup>1</sup>

LamBERTa models are generated by fine-tuning the pre-trained BERT models on a sequence classification task (i.e., BERT with a single linear classification layer on top) given in input the articles of the ICC or a portion of it. This fine-tuning is accomplished by using a typical configuration of BERT for masked language modeling, with 12 attention heads and 12 hidden

<sup>1</sup>bert-base-italian-xxl-uncased, available at <https://huggingface.co/dbmdz/>.

**Table 1**

Main statistics on the ICC books, and additional statistics depending on the domain-specific token injection (number of domain-specific tokens to inject into the pre-trained BERT vocabulary, final vocabulary size, and percentage increase)

ICC portion	# articles	# sentences	# words	# injected tokens	vocabulary size	% increase
<i>Book-1</i>	395	1979	32354	833	31935	2.68%
<i>Book-2</i>	345	1561	24520	698	31800	2.24%
<i>Book-3</i>	364	1619	25971	1072	32174	3.45%
<i>Book-4</i>	891	3595	50509	1383	32485	4.45%
<i>Book-5</i>	713	3937	75764	2048	33150	6.58%
<i>Book-6</i>	331	1453	25937	829	31931	2.66%
All	3039	14131	234945	3993	35095	12.84%

layers, and initial (i.e., pre-trained) vocabulary of 31 102 tokens. Each model is trained for 10 epochs, using cross-entropy as loss function, AdamW optimizer and initial learning rate selected within  $[1e-5, 5e-5]$  on batches of 256 examples.

Notably, LambBERTa is flexible w.r.t. two peculiar modeling aspects: (i) the training-instance labeling scheme for a given set of ICC articles, and (ii) the learning approach. The former will be discussed later in Section 3, whereas the latter concerns the possibility of training models either on the individual books or on the entire ICC corpus; due to space limitations of this paper, we shall focus on the book-specific models.

Another feature of LambBERTa is the **injection of previously unseen legal terms**, selected from the task-specific corpus (i.e., ICC), that are out-of-vocabulary of the Italian pre-trained model. This way, the BERT tokenizer is enabled to recognize those terms appearing in the ICC, while fine-tuning on it, and hence, to avoid breaking them down into subwords. To select such terms to be added as new tokens in LambBERTa, the text of each book in the ICC is processed to remove Italian stopwords and filter out overly frequent terms (as occurring in more than 50% of the articles in  $D$ ) as well as hapax terms. Table 1 reports the number of added tokens and the final number of tokens, for each book of the ICC.

### 2.3. ITALIAN-LEGAL-BERT

ITALIAN-LEGAL-BERT [15] follows the typical BERT architecture, with a language modeling head on top, AdamW Optimizer, initial learning rate  $5e-5$ . ITALIAN-LEGAL-BERT was built upon ITALIAN-XXL-CASED by further pre-training the latter for additional 4 epochs on a corpus extracted from the National Jurisprudential Archive (pst.giustizia.it), a repository containing millions of legal documents, such as decrees, orders, and civil judgments, from Italian courts and courts of appeal. The corpus used to train ITALIAN-LEGAL-BERT is 3.7 GB of text containing above 21M sentences and 498M words. The trained ITALIAN-LEGAL-BERT was evaluated on named entity recognition, sentence classification, and sentence similarity tasks, using 20K civil cases from the National Jurisprudential Archive and above 21K criminal cases from italgireweb (italgiureweb.giustizia.it).

### 3. Training and evaluation data

One important model aspect of LambBERTa corresponds to the **unsupervised article-labeling schemes** that are used to produce a training set for each of the books in the ICC. This is not trivial since two main requirements need to be satisfied: (i) a one-to-one association must hold for classes and articles, since a LambBERTa model is designed to be a classifier at article level, i.e., class labels correspond to the articles in the book(s) covered by the model, and (ii) the entire ICC must be used to fully embed its knowledge. Therefore, a key issue is how to create as many training instances as possible for each article to make LambBERTa learn effectively. To this purpose, in [14], we defined different strategies for selecting and combining portions from each article to build the training set for any specific book, paying also attention to balance the contributions of each article, which are originally varying in length. Given a minimum number of training units per article ( $minTU$ ), by default set to 32, each of the article labeling schemes implements a *round-robin* (RR) method that iterates over replicas of the same group of training units per article until at least  $minTU$  are generated. The most effective scheme turned out to be the *unigram with parameterized emphasis on the title*, which builds the set of training units for each article as comprised of two subsets: the one containing the article’s sentences with round-robin selection, and the other one containing only replicas of the article’s title.

In [14], LambBERTa models are assessed through extensive experiments by considering single-label and multi-label evaluation tasks, based on **different types of queries**, which vary by source, length and lexical characteristics. In this work, we shall use the following query-sets, each defined for any specific book  $B$  of the ICC:

- (QT1) Randomly selected sentences from the articles of book  $B$ ;
- (QT2) Same as QT1, but the sentences are *paraphrased* through an Italian-English-Italian translation of the queries;
- (QT3) *Comments* on the articles of book  $B$ , i.e., annotations about the interpretation of the meanings and law implications associated to an article (laleggepertutti.it);
- (QT4) *Case law decisions* from the civil section of the Italian Court of Cassation that contains jurisprudential sentences associated with the articles of  $B$ .

It should be noted that the above query sets represent different testbeds, whose “difficulty” is highly varying, from lower (QT1) to higher (QT3 and QT4). Due to space limitations of this paper, we refer the reader to [14] for further details on the characteristics of the query sets.

As concerns the assessment criteria, here we consider single-label evaluation criteria only. For each article  $A_i$ , we start by measuring the precision for  $A_i$  ( $P_i$ ), i.e., the number of times (queries)  $A_i$  was correctly predicted out of all predictions of  $A_i$ , the recall for  $A_i$  ( $R_i$ ), i.e., the number of times (queries)  $A_i$  was correctly predicted out of all queries actually pertinent to  $A_i$ , and the F-measure for  $A_i$  ( $F_i$ ). Then, we averaged over all articles to obtain the per-article average *precision* ( $P$ ), *recall* ( $R$ ), *micro-averaged F-measure* ( $F^\mu$ ) as the average over all  $F_i$ s, and *macro-averaged F-measure* ( $F^M$ ) as the harmonic mean of  $P$  and  $R$ . In addition, we account for the top- $k$  predictions and the position (rank) of the correct article in predictions: the former is the fraction of correct article labels that are found in the top- $k$  predictions (i.e., top- $k$ -probability results in response to each query), and averaging over all queries, which is the *recall@k* ( $R@k$ ); the latter is the *mean reciprocal rank* ( $MRR$ ) considering for each query the rank of the correct prediction over the classification probability distribution, and averaging over all queries.

**Table 2**

LamBERTa-V1 vs. LamBERTa-V2 for all books of the ICC on book-sentence-queries (QT1), paraphrased-sentence-queries (QT2), comment-queries (QT3), and case-queries (QT4): Recall, Precision, micro- and macro-averaged F-measures, Recall@ $k$ , and MRR. (Bold values correspond to the best model for each book, evaluation criterion, and query set)

Query type	Book	$R$		$P$		$F^\mu$		$F^M$		$R@3$		$R@10$		$MRR$	
		V1	V2	V1	V2	V1	V2	V1	V2	V1	V2	V1	V2	V1	V2
QT1	$B_1$	0.962	<b>0.975</b>	0.975	<b>0.983</b>	0.961	<b>0.973</b>	0.969	<b>0.979</b>	0.989	<b>0.997</b>	0.999	<b>1.000</b>	0.976	<b>0.983</b>
	$B_2$	0.972	0.972	0.981	<b>0.982</b>	0.971	<b>0.972</b>	0.977	0.977	0.994	0.994	<b>0.999</b>	0.997	0.983	0.983
	$B_3$	<b>0.990</b>	0.935	<b>0.994</b>	0.961	<b>0.990</b>	0.936	<b>0.992</b>	0.948	<b>1.000</b>	0.972	<b>1.000</b>	0.985	<b>0.995</b>	0.957
	$B_4$	<b>0.961</b>	0.665	<b>0.983</b>	0.768	<b>0.964</b>	0.672	<b>0.972</b>	0.712	<b>0.984</b>	0.750	<b>0.990</b>	0.831	<b>0.975</b>	0.721
	$B_5$	<b>0.910</b>	0.562	<b>0.944</b>	0.697	<b>0.909</b>	0.581	<b>0.927</b>	0.622	<b>0.984</b>	0.687	<b>0.998</b>	0.769	<b>0.947</b>	0.639
	$B_6$	0.979	0.979	0.986	0.986	0.978	0.978	0.982	<b>0.983</b>	1.000	1.000	1.000	1.000	0.989	0.989
QT2	$B_1$	<b>0.841</b>	0.815	<b>0.866</b>	0.837	<b>0.828</b>	0.800	<b>0.853</b>	0.826	0.905	<b>0.909</b>	<b>0.949</b>	0.939	<b>0.881</b>	0.865
	$B_2$	<b>0.828</b>	0.799	<b>0.856</b>	0.842	<b>0.814</b>	0.788	<b>0.841</b>	0.820	<b>0.892</b>	0.881	<b>0.941</b>	0.919	<b>0.871</b>	0.845
	$B_3$	<b>0.861</b>	0.740	<b>0.886</b>	0.773	<b>0.851</b>	0.725	<b>0.873</b>	0.756	<b>0.922</b>	0.831	<b>0.942</b>	0.870	<b>0.896</b>	0.794
	$B_4$	<b>0.736</b>	0.463	<b>0.756</b>	0.477	<b>0.713</b>	0.436	<b>0.746</b>	0.470	<b>0.806</b>	0.528	<b>0.861</b>	0.598	<b>0.779</b>	0.512
	$B_5$	<b>0.718</b>	0.401	<b>0.759</b>	0.451	<b>0.710</b>	0.393	<b>0.738</b>	0.425	<b>0.843</b>	0.508	<b>0.908</b>	0.582	<b>0.790</b>	0.472
	$B_6$	0.841	<b>0.852</b>	0.874	<b>0.886</b>	0.833	<b>0.843</b>	0.857	<b>0.869</b>	0.914	<b>0.918</b>	0.941	<b>0.947</b>	0.882	<b>0.889</b>
QT3	$B_1$	<b>0.349</b>	0.297	<b>0.248</b>	0.192	<b>0.274</b>	0.222	<b>0.290</b>	0.233	<b>0.494</b>	0.462	<b>0.675</b>	0.636	<b>0.455</b>	0.411
	$B_2$	0.313	<b>0.320</b>	0.213	<b>0.226</b>	0.239	<b>0.251</b>	0.253	<b>0.265</b>	0.494	<b>0.522</b>	0.655	<b>0.661</b>	<b>0.445</b>	0.440
	$B_3$	<b>0.396</b>	0.260	<b>0.298</b>	0.195	<b>0.325</b>	0.212	<b>0.340</b>	0.223	<b>0.577</b>	0.376	<b>0.704</b>	0.518	<b>0.507</b>	0.346
	$B_4$	<b>0.336</b>	0.078	<b>0.239</b>	0.059	<b>0.264</b>	0.064	<b>0.279</b>	0.067	<b>0.487</b>	0.115	<b>0.622</b>	0.160	<b>0.438</b>	0.107
	$B_5$	<b>0.171</b>	0.019	<b>0.128</b>	0.018	<b>0.137</b>	0.018	<b>0.147</b>	0.019	<b>0.364</b>	0.037	<b>0.562</b>	0.073	<b>0.302</b>	0.041
	$B_6$	0.387	<b>0.442</b>	0.291	<b>0.334</b>	0.317	<b>0.364</b>	0.332	<b>0.381</b>	0.588	<b>0.619</b>	0.745	<b>0.765</b>	0.515	<b>0.549</b>
QT4	$B_1$	<b>0.228</b>	0.182	<b>0.233</b>	0.194	<b>0.210</b>	0.164	<b>0.230</b>	0.187	<b>0.494</b>	0.360	<b>0.813</b>	0.526	<b>0.430</b>	0.328
	$B_2$	<b>0.292</b>	0.246	<b>0.316</b>	0.269	<b>0.274</b>	0.217	<b>0.303</b>	0.257	<b>0.501</b>	0.447	<b>0.726</b>	0.651	<b>0.465</b>	0.396
	$B_3$	<b>0.284</b>	0.197	<b>0.323</b>	0.220	<b>0.273</b>	0.177	<b>0.302</b>	0.208	<b>0.566</b>	0.289	<b>0.856</b>	0.436	<b>0.489</b>	0.268
	$B_4$	<b>0.259</b>	0.050	<b>0.299</b>	0.079	<b>0.250</b>	0.055	<b>0.278</b>	0.061	<b>0.528</b>	0.108	<b>0.803</b>	0.148	<b>0.458</b>	0.092
	$B_5$	<b>0.354</b>	0.019	<b>0.401</b>	0.046	<b>0.342</b>	0.023	<b>0.376</b>	0.027	<b>0.641</b>	0.040	<b>0.884</b>	0.062	<b>0.564</b>	0.040
	$B_6$	<b>0.392</b>	0.378	<b>0.445</b>	0.376	<b>0.372</b>	0.334	<b>0.417</b>	0.377	<b>0.665</b>	0.577	<b>0.885</b>	0.720	<b>0.590</b>	0.514

## 4. Rebuilding LamBERTa based on ITALIAN-LEGAL-BERT

To answer our first research question (RQ1), we develop a new version of LamBERTa by replacing the general-domain pre-trained Italian model (i.e., ITALIAN-XXL-UNCASED) with a legal-specific pre-trained Italian model (i.e., ITALIAN-LEGAL-BERT); recall that the latter model is the result of a further pre-training of the former, although on a cased version. We hereinafter refer to this version of LamBERTa as LamBERTa-V2, to distinguish from the original in [14] hereinafter denoted as LamBERTa-V1. Table 2 summarizes results of the comparison between the two versions based on their evaluation through all query sets. Note that results by original LamBERTa models are borrowed from [14].<sup>2</sup>

At a first glance, it can be noticed that although there is no absolute winner, LamBERTa-V1 generally achieves better performance than LamBERTa-V2. For all query types, LamBERTa-V2 appears to lose more when evaluated on queries pertaining the largest books (i.e., Book-4 and Book-5). Moreover, regardless of the book, the gap of LamBERTa-V2 is particularly evident for the most difficult query sets, i.e., QT3 and QT4, which contain queries that are the most distant, both lexically and semantically, from the language used in the training instances. On

<sup>2</sup>All results shown in Tables 2–4 correspond to the use of same seed setting (for handling computation randomness) and hardware configuration for all LamBERTa models.



**Table 3**

LamBERTa-V1-NoDST vs. LamBERTa-V2-NoDST for all books of the ICC on book-sentence-queries (QT1), paraphrased-sentence-queries (QT2), comment-queries (QT3), and case-queries (QT4): Recall, Precision, micro- and macro-averaged F-measures, Recall@ $k$ , and MRR. (*Bold values correspond to the best model for each book, evaluation criterion, and query set*)

Query type	Book	$R$		$P$		$F^\mu$		$F^M$		$R@3$		$R@10$		$MRR$	
		V1	V2	V1	V2	V1	V2	V1	V2	V1	V2	V1	V2	V1	V2
QT1	$B_1$	0.975	0.975	0.983	0.983	0.973	0.973	0.979	0.979	<b>0.996</b>	0.995	1.000	1.000	0.985	0.985
	$B_2$	0.974	<b>0.975</b>	0.983	<b>0.985</b>	0.973	<b>0.975</b>	0.978	<b>0.980</b>	0.993	<b>0.994</b>	0.996	<b>0.999</b>	0.984	<b>0.985</b>
	$B_3$	0.988	0.988	<b>0.991</b>	0.990	0.987	0.987	0.989	0.989	1.000	1.000	1.000	1.000	0.995	0.995
	$B_4$	<b>0.968</b>	0.967	<b>0.989</b>	0.988	<b>0.971</b>	0.970	<b>0.978</b>	0.977	0.985	0.985	0.990	0.990	0.977	0.977
	$B_5$	<b>0.924</b>	0.922	<b>0.957</b>	0.953	<b>0.924</b>	0.920	<b>0.941</b>	0.937	<b>0.986</b>	0.985	<b>0.999</b>	0.998	<b>0.956</b>	0.954
	$B_6$	0.979	<b>0.980</b>	0.986	<b>0.987</b>	0.978	<b>0.979</b>	0.983	<b>0.984</b>	1.000	1.000	1.000	1.000	0.989	<b>0.990</b>
QT2	$B_1$	<b>0.863</b>	0.851	<b>0.875</b>	0.874	<b>0.849</b>	0.839	<b>0.869</b>	0.862	<b>0.929</b>	0.920	<b>0.962</b>	0.957	<b>0.901</b>	0.890
	$B_2$	<b>0.867</b>	0.845	<b>0.882</b>	0.859	<b>0.854</b>	0.830	<b>0.874</b>	0.852	<b>0.920</b>	0.916	<b>0.959</b>	0.945	<b>0.900</b>	0.885
	$B_3$	<b>0.890</b>	0.867	<b>0.898</b>	0.875	<b>0.876</b>	0.852	<b>0.894</b>	0.871	<b>0.939</b>	0.917	<b>0.963</b>	0.943	<b>0.920</b>	0.898
	$B_4$	<b>0.761</b>	0.753	<b>0.774</b>	0.769	<b>0.738</b>	0.730	<b>0.767</b>	0.760	<b>0.834</b>	0.833	0.882	<b>0.889</b>	<b>0.804</b>	0.801
	$B_5$	<b>0.816</b>	0.778	<b>0.838</b>	0.791	<b>0.801</b>	0.755	<b>0.827</b>	0.785	<b>0.918</b>	0.886	<b>0.952</b>	0.934	<b>0.870</b>	0.839
	$B_6$	<b>0.867</b>	0.864	0.884	<b>0.896</b>	0.856	0.856	0.875	<b>0.880</b>	0.931	0.931	<b>0.962</b>	0.960	<b>0.904</b>	0.900
QT3	$B_1$	<b>0.437</b>	0.388	<b>0.318</b>	0.278	<b>0.351</b>	0.308	<b>0.368</b>	0.324	<b>0.612</b>	0.575	<b>0.740</b>	0.731	<b>0.547</b>	0.510
	$B_2$	<b>0.491</b>	0.478	<b>0.368</b>	0.360	<b>0.401</b>	0.394	<b>0.421</b>	0.411	<b>0.652</b>	0.643	0.792	<b>0.807</b>	<b>0.596</b>	0.589
	$B_3$	<b>0.551</b>	0.477	<b>0.444</b>	0.371	<b>0.475</b>	0.403	<b>0.492</b>	0.418	<b>0.709</b>	0.649	<b>0.805</b>	0.787	<b>0.648</b>	0.591
	$B_4$	0.447	<b>0.449</b>	<b>0.349</b>	0.343	<b>0.377</b>	0.372	<b>0.392</b>	0.389	0.602	<b>0.642</b>	0.720	<b>0.770</b>	0.545	<b>0.566</b>
	$B_5$	<b>0.401</b>	0.322	<b>0.301</b>	0.241	<b>0.327</b>	0.260	<b>0.344</b>	0.276	<b>0.567</b>	0.537	<b>0.712</b>	0.697	<b>0.513</b>	0.460
	$B_6$	0.497	<b>0.553</b>	0.396	<b>0.443</b>	0.425	<b>0.474</b>	0.441	<b>0.492</b>	0.673	<b>0.707</b>	0.803	<b>0.833</b>	0.600	<b>0.653</b>
QT4	$B_1$	<b>0.190</b>	0.175	0.201	<b>0.210</b>	<b>0.174</b>	0.171	<b>0.195</b>	0.191	<b>0.367</b>	0.353	<b>0.581</b>	0.529	<b>0.347</b>	0.313
	$B_2$	0.351	<b>0.357</b>	0.362	<b>0.375</b>	0.318	<b>0.327</b>	0.356	<b>0.365</b>	0.539	<b>0.562</b>	0.697	<b>0.720</b>	0.498	<b>0.502</b>
	$B_3$	0.325	<b>0.329</b>	<b>0.354</b>	0.323	<b>0.303</b>	0.291	<b>0.339</b>	0.326	0.518	<b>0.526</b>	0.675	<b>0.678</b>	0.443	<b>0.455</b>
	$B_4$	0.259	<b>0.301</b>	<b>0.297</b>	0.295	0.245	<b>0.271</b>	0.276	<b>0.298</b>	0.416	<b>0.510</b>	0.577	<b>0.670</b>	0.372	<b>0.446</b>
	$B_5$	<b>0.406</b>	0.371	<b>0.421</b>	0.372	<b>0.379</b>	0.343	<b>0.413</b>	0.371	<b>0.650</b>	0.647	0.768	<b>0.802</b>	0.588	0.558
	$B_6$	0.307	<b>0.470</b>	0.352	<b>0.470</b>	0.297	<b>0.429</b>	0.328	<b>0.470</b>	0.544	<b>0.659</b>	0.698	<b>0.816</b>	0.474	<b>0.600</b>

average over all books, LamBERTa-V2 has indeed a percentage decrease of above 40% on case queries (QT4) and above 27% on comment queries (QT3); remarkably, while this holds for all criteria, the negative peaks are reached for the top-3 and top-10 predictions: -46.4%  $R@3$  and -48.8%  $R@10$ , on QT4, and about 29%  $R@3$  and  $R@10$ , on QT3. In light of the above results, it stands out that using a legal pre-trained model does not bring advantage over a general-domain pre-trained model to fine-tune on the downstream task of ICC article prediction, and actually the legal pre-trained model can often achieve worse performance.

## 5. Investigating on the domain-specific token injection

**Effect of token injection removal.** A major goal of this work is to delve into the effect of the domain-specific token injection into LamBERTa models. To answer our **RQ2**, we first analyze the changes in the behavior of LamBERTa when no out-of-vocabulary tokens are added. We shall use suffix NoDST to distinguish this setting from the original one using token injection.

Results obtained by LamBERTa-V1-NoDST and LamBERTa-V2-NoDST are shown in Table 3. First, we notice that the performance difference of LamBERTa-V2-NoDST w.r.t. LamBERTa-V1-NoDST is reduced, though still remaining negative, with the exception of QT4, where LamBERTa-V2-NoDST achieves average percentage increase of about 3%  $P$  up to 9%  $R$ . More

interesting is to compare the obtained results against those in Table 2. The new setting leads to an improvement of both versions of LamBERTa in most cases, where the ITALIAN-LEGAL-BERT based version takes major benefits. More precisely, the two versions of LamBERTa improves slightly on QT1 and QT2, and more significantly on QT3. By contrast, on QT4, while LamBERTa-V2-NoDST achieves average percentage increase vs. LamBERTa-V2 (from 65% to about 90%), taking light advantage on other models in terms of  $R$ ,  $P$ ,  $F$  criteria, LamBERTa-V1-NoDST tends to be worse than the original LamBERTa-V1 which remains the absolute winner according to the  $R@k$  and  $MRR$  criteria.

**Effect of embedding initialization for token injection.** The above results prompted us to further investigate on the effect of injecting out-of-vocabulary legal terms into LamBERTa models, by focusing now on the *initialization* of the added tokens. In fact, it should be noted that in the original setting of LamBERTa, the selected domain-specific tokens are added to the Italian pre-trained tokenizer using a random initialization. Therefore, to provide a more exhaustive answer to our **RQ2**, we define an enhanced setting for the domain-specific tokens to be added. Our goal is to compute initial embeddings for the new tokens that are not random but incorporate proper knowledge of the legal language. One approach we tried is to initialize each word  $w$  to be added by getting the [CLS] output embedding computed when prompting the Italian pre-trained model, or alternatively ITALIAN-LEGAL-BERT, with just  $w$ . Similarly, we tried by averaging the output embeddings of the tokens corresponding to the subwords of  $w$  detected by the BERT tokenizer. Unfortunately, in both cases, exploiting the output embeddings shows to be inappropriate, which might be due to the fact that these contextualized representations incorporate also the segment and position embeddings. Then, we shifted our attention to vectors extracted from the token embeddings matrix (i.e., the first level of BERT input representation). Given a word  $w$  to be added, we get the token embedding of each of its subwords (excluding [CLS] and [SEP] embeddings) and tried different pooling strategies. The one leading to the best results is initializing the  $w$ 's embedding with the initial embedding of its root subword. This setting is hereinafter referred to as ReDST.

Results for this new setting are reported in Table 4. A first remark that stands out is the benefit brought by this new setting of initialization of the injected tokens w.r.t. a random initialization, for both versions of LamBERTa. This holds always, with the exception of QT4 according to  $R@k$  and  $MRR$  criteria, whereby LamBERTa-V1 is the absolute winner over all models. Besides, LamBERTa-V1-ReDST and LamBERTa-V2-ReDST actually perform comparably or better than LamBERTa-V1-NoDST and LamBERTa-V2-NoDST on QT1, and clearly better than LamBERTa-V1-NoDST and LamBERTa-V2-NoDST on QT4 according to  $R@k$  and  $MRR$ .

## 6. Explainability aspects

Like for any machine and deep learning models, *explainability* of PLMs is central to understand their solutions provided for a given NLP task. This becomes even more crucial when artificial intelligence meets a challenging field like law (e.g., [19, 20]).

Since our earlier study [14], we investigated explainability of our LamBERTa models, with a focus on how they form complex relationships between the textual tokens, and their distinctive attention patterns. In this paper, we take a different perspective, which is more suited for



**Table 4**

LamBERTa-V1-ReDST vs. LamBERTa-V2-ReDST for all books of the ICC on book-sentence-queries (QT1), paraphrased-sentence-queries (QT2), comment-queries (QT3), and case-queries (QT4): Recall, Precision, micro- and macro-averaged F-measures, Recall@ $k$ , and MRR. (*Bold values correspond to the best model for each book, evaluation criterion, and query set*)

Query type	Book	$R$		$P$		$F^\mu$		$F^M$		$R@3$		$R@10$		$MRR$	
		V1	V2	V1	V2	V1	V2	V1	V2	V1	V2	V1	V2	V1	V2
QT1	$B_1$	<b>0.973</b>	0.972	0.981	<b>0.982</b>	<b>0.972</b>	0.971	0.977	0.977	0.996	0.996	1.000	1.000	0.984	0.984
	$B_2$	0.974	0.974	0.983	0.983	0.973	0.973	0.978	0.978	0.994	0.994	0.997	0.997	0.984	0.984
	$B_3$	0.988	0.988	0.991	0.991	0.987	0.987	0.989	0.989	1.000	1.000	1.000	1.000	0.995	0.995
	$B_4$	0.968	0.968	0.989	0.989	0.971	0.971	0.978	0.978	0.985	0.985	0.990	0.990	0.977	0.977
	$B_5$	<b>0.924</b>	0.920	<b>0.957</b>	0.952	<b>0.924</b>	0.919	<b>0.941</b>	0.936	<b>0.986</b>	0.985	0.998	<b>0.999</b>	<b>0.955</b>	0.953
	$B_6$	0.979	0.979	0.986	0.986	0.978	0.978	0.983	0.983	1.000	1.000	1.000	1.000	0.989	0.989
QT2	$B_1$	<b>0.846</b>	0.843	0.861	<b>0.870</b>	0.832	<b>0.833</b>	0.853	<b>0.856</b>	<b>0.928</b>	0.918	<b>0.957</b>	0.951	<b>0.890</b>	0.884
	$B_2$	<b>0.843</b>	0.829	<b>0.859</b>	0.850	<b>0.827</b>	0.814	<b>0.851</b>	0.839	<b>0.913</b>	0.904	<b>0.951</b>	0.943	<b>0.885</b>	0.870
	$B_3$	<b>0.883</b>	0.854	<b>0.889</b>	0.871	<b>0.868</b>	0.841	<b>0.886</b>	0.862	<b>0.924</b>	0.909	<b>0.958</b>	0.939	<b>0.911</b>	0.887
	$B_4$	<b>0.763</b>	0.724	<b>0.778</b>	0.729	<b>0.739</b>	0.696	<b>0.770</b>	0.727	<b>0.840</b>	0.804	<b>0.886</b>	0.864	<b>0.808</b>	0.772
	$B_5$	<b>0.792</b>	0.763	<b>0.813</b>	0.783	<b>0.773</b>	0.742	<b>0.802</b>	0.773	<b>0.898</b>	0.872	<b>0.944</b>	0.921	<b>0.850</b>	0.825
	$B_6$	0.844	<b>0.849</b>	0.875	<b>0.877</b>	0.834	<b>0.838</b>	0.860	<b>0.863</b>	0.921	0.921	0.951	<b>0.957</b>	0.887	<b>0.888</b>
QT3	$B_1$	<b>0.408</b>	0.364	<b>0.308</b>	0.266	<b>0.334</b>	0.292	<b>0.351</b>	0.308	<b>0.578</b>	0.498	<b>0.709</b>	0.670	<b>0.518</b>	0.463
	$B_2$	<b>0.450</b>	0.335	<b>0.344</b>	0.230	<b>0.374</b>	0.259	<b>0.390</b>	0.273	<b>0.637</b>	0.540	<b>0.786</b>	0.702	<b>0.564</b>	0.466
	$B_3$	<b>0.477</b>	0.372	<b>0.379</b>	0.289	<b>0.407</b>	0.311	<b>0.423</b>	0.325	<b>0.642</b>	0.525	<b>0.745</b>	0.695	<b>0.581</b>	0.477
	$B_4$	<b>0.439</b>	0.353	<b>0.341</b>	0.241	<b>0.370</b>	0.272	<b>0.384</b>	0.287	<b>0.594</b>	0.514	<b>0.695</b>	0.663	<b>0.536</b>	0.460
	$B_5$	<b>0.415</b>	0.316	<b>0.312</b>	0.238	<b>0.339</b>	0.258	<b>0.356</b>	0.272	<b>0.575</b>	0.484	<b>0.724</b>	0.636	<b>0.521</b>	0.432
	$B_6$	<b>0.507</b>	0.433	<b>0.406</b>	0.344	<b>0.434</b>	0.365	<b>0.451</b>	0.383	<b>0.656</b>	0.653	0.813	<b>0.823</b>	<b>0.612</b>	0.567
QT4	$B_1$	<b>0.181</b>	0.165	<b>0.212</b>	0.181	<b>0.177</b>	0.152	<b>0.195</b>	0.172	<b>0.351</b>	0.338	<b>0.532</b>	0.523	<b>0.335</b>	0.309
	$B_2$	0.296	<b>0.299</b>	<b>0.336</b>	0.329	<b>0.276</b>	0.274	<b>0.315</b>	0.313	0.530	<b>0.539</b>	0.677	<b>0.715</b>	<b>0.471</b>	0.464
	$B_3$	<b>0.298</b>	0.258	<b>0.298</b>	0.223	<b>0.262</b>	0.218	<b>0.298</b>	0.239	<b>0.496</b>	0.441	<b>0.653</b>	0.629	<b>0.443</b>	0.395
	$B_4$	<b>0.290</b>	0.267	<b>0.291</b>	0.280	<b>0.259</b>	0.244	<b>0.291</b>	0.273	0.455	<b>0.461</b>	<b>0.613</b>	0.597	<b>0.406</b>	0.404
	$B_5$	<b>0.323</b>	0.295	<b>0.337</b>	0.314	<b>0.298</b>	0.271	<b>0.330</b>	0.304	<b>0.620</b>	0.575	<b>0.774</b>	0.740	<b>0.526</b>	0.496
	$B_6$	<b>0.430</b>	0.351	<b>0.471</b>	0.369	<b>0.417</b>	0.329	<b>0.450</b>	0.359	<b>0.658</b>	0.551	<b>0.782</b>	0.747	<b>0.587</b>	0.499

providing interpretation of the models' prediction.

To this purpose, we use LIME - Local Interpretable Model-Agnostic Explanations [21]. LIME aims to explain the behavior of the underlying classifier "around" a query instance: by perturbing interpretable parts of the input query (i.e., words, for textual data), LIME weighs these perturbed data points by their proximity to the original query instance, and observes the associated predictions by the underlying classifier to determine which of those changes will have most impact on the prediction of the original query. To this purpose, the explanation is accomplished by approximating the underlying classifier locally by an interpretable one, such as a linear model. It should be noted that, even if the original classifier deals with non-linear complexities, like our LamBERTa models, in the neighborhood of an instance it behaves roughly as a linear model; therefore the local linear approximation can reasonably be assumed to be correct since LIME looks at a very small region around the query instance.

Figures 2–4 show LIME explanations of the LamBERTa models for various use-case queries. It should be noted that the queries have been defined in the form of questions and do not have strong syntactical patterns matching sentences of the corresponding relevant articles.

In line with the quantitative results discussed in the previous section, the token injection could be unnecessary to get the correct prediction. In this regard, one example is given by the query *Chi può essere escluso dal testamento?* (*Who can be excluded from the testament?*), shown

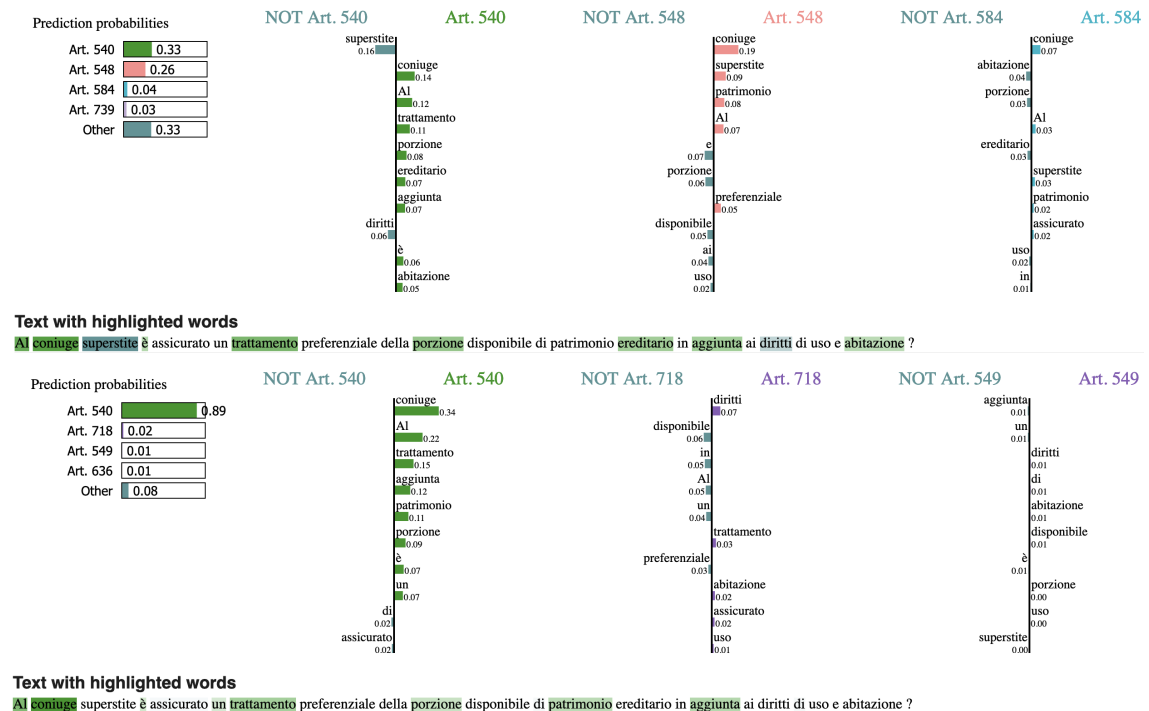


**Figure 2:** Explanations of LambBERTa-V1-NoDST (top) and LambBERTa-V2-NoDST (bottom) predictions for query *Who can be excluded from the testament?*

in Figure 2, which contains all words appearing in the pre-trained Italian-BERT vocabulary. Indeed, the ground-truth article (Art. 463) is correctly predicted by LambBERTa-V2-NoDST and LambBERTa-V1-NoDST, where the latter leverage on more terms as it can be noticed by the LIME-explainer’s highlighting on the important words.

Nonetheless, token injection can still be helpful in some cases. For instance, given query *Al coniuge superstite è assicurato un trattamento preferenziale della porzione disponibile di patrimonio ereditario in aggiunta ai diritti di uso e abitazione?* (*Is the surviving spouse granted preferential treatment of the available portion of the hereditary patrimony in addition to the rights of use and habitation?*), we find this is correctly answered by both LambBERTa-V1 and LambBERTa-V2 models that exploit ICC-specific token injection; for the sake of brevity, in Figure 3 we report explanations for models with random initialization of the injected tokens. This query contains terms ‘superstite’ and ‘ereditario’ that were added to the tokenizer, and hence are fully recognized by such models – otherwise they are subworded by those models equipped with the original pre-trained Italian-BERT vocabulary. Notably, although LambBERTa-V2 achieves higher prediction probability for the ground-truth article (Art. 540), LambBERTa-V1 behaves overall better as it is able to also retrieve the second most relevant article for the query (i.e., Art. 548). In this regard, we notice that while LambBERTa-V1’s predictions are explained also by means of terms ‘superstite’ and ‘ereditario’, these are not recognized as essential by LambBERTa-V2.

The benefit from using token injection is also evident for query *I figli nati sono rappresentati dai genitori congiuntamente?* (*Are the unborn children represented by the parents jointly?*), where ‘nascituri’ is missing from the pre-trained Italian-BERT vocabulary, and hence it is subworded (into 3 tokens). The ground-truth article (Art. 320) is strongly predicted by LambBERTa-V1 and LambBERTa-V1-ReDST, but is not by the counterpart LambBERTa-V2 models: looking at the LIME explanation (Figure 4), the injected token ‘nascituri’ is well-recognized in LambBERTa-V1 as an important feature for prediction along with other neighboring words in the query, but this does



**Figure 3:** Explanations of LambBERTa-V1 (top) and LambBERTa-V2 (bottom) predictions for query *Is the surviving spouse granted preferential treatment of the available portion of the hereditary patrimony in addition to the rights of use and habitation?*

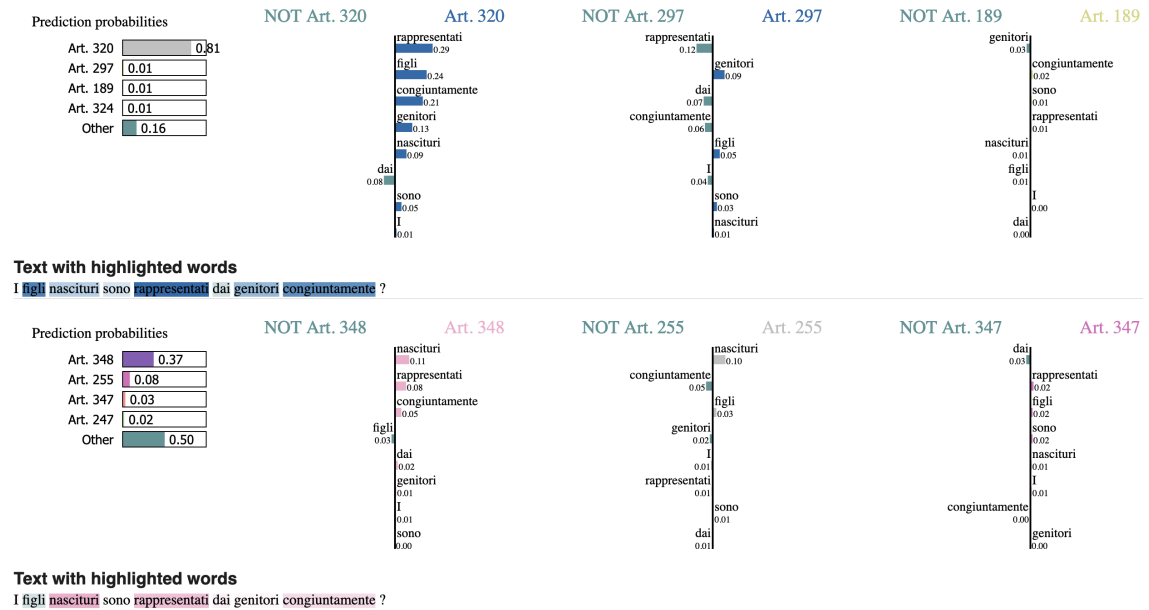
not hold for LambBERTa-V2. This may hint at a phenomenon of fresh-knowledge acquisition, by a model (i.e., LambBERTa-V1) that fine-tunes a general-domain one, against early-knowledge update exhibited by a model (i.e., LambBERTa-V2) that fine-tunes a domain-specific one.

## 7. Discussion

Here we summarize main findings according to our previously stated research questions.

Concerning **RQ1**, the behavior of LambBERTa models changes depending on the Italian pre-trained BERT model, especially on more difficult query testbeds. However, for the task of ICC article retrieval, fine-tuning on Italian legal pre-trained model does not bring particular advantage w.r.t. fine-tuning on Italian general-domain pre-trained model; therefore, original LambBERTa turns out to be preferable in most query scenarios. We point out this should not be surprising — after all, in [22], Legal-BERT was shown to achieve only slightly better performances than BERT, on both classification and entity recognition tasks. More importantly, our result is in line with some recent studies that demonstrated how domain adaptive pre-training leads to significant improvements only with low-resource downstream tasks [23].

Injecting domain-specific tokens, i.e., out-of-vocabulary legal terms (**RQ2**) can provide low or no benefits in most cases; however, a non-random initialization of the new tokens to be injected significantly improves LambBERTa performance. Moreover, remarks drawn from the explanation



**Figure 4:** Explanations of LambBERTa-V1 (top) and LambBERTa-V2-NoDST (bottom) predictions for query *Are the unborn children represented by the parents jointly?*

of the LambBERTa models’ predictions (RQ3) have confirmed that different domain-specific token-injection settings might be required to successfully address different query scenarios.

To sum up, thus answering our RQ4, domain adaptation reveals to be less determinant than task adaptation for the task of article retrieval w.r.t. the Italian Civil Code. Nonetheless, there are specific situations relating to the underlying lexical/semantic aspects of the input queries that require to be handled by different variants of LambBERTa for successfully accomplishing the retrieval task.

## 8. Conclusions

We presented a follow up of our research on Italian BERT models for the task of law article retrieval, with application to the Italian Civil Code (ICC), which updates the current state-of-the-art of PLMs for prediction tasks in the Italian legal domain. Our goal was to delve into the effects of domain-adaptation in combination with the task-adaptation to learn Italian BERT models (LambBERTa) for the task of ICC article prediction. To this purpose, we investigated the role of a recently defined Italian legal BERT into our framework, as well as the effects of enhancing the tokenizer with new terms selected from the target legal corpus, by varying the setting for the initial token embeddings of such terms.

We expect that our work can pave the way for further exploration of domain-/task-adaptation for (Italian) legal BERT models. On our side, we plan to define effective methods to integrate task-oriented knowledge on a pre-trained (domain-adaptive) model. Also, we believe that a from-scratch pre-trained Italian legal BERT model is worthy being developed.

## References

- [1] H. Surden, Artificial intelligence and law: An overview, 35 GA. ST. U. L. REV. 1305 (2019).
- [2] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proc. NAACL-HLT, 2019, pp. 4171–4186.
- [3] J. Rabelo, M. Kim, R. Goebel, Combining similarity and transformer methods for case law entailment, in: Proc. Int. Conf. on Artificial Intelligence and Law (ICAIL), 2019, pp. 290–296.
- [4] I. Chalkidis, I. Androutopoulos, N. Aletras, Neural legal judgment prediction in english, in: Proc. ACL, Association for Computational Linguistics, 2019, pp. 4317–4323.
- [5] L. Sanchez, J. He, J. Manotumruksa, D. Albakour, M. Martinez, A. Lipani, Easing legal news monitoring with learning to rank and BERT, in: Proc. ECIR, volume 12036 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 336–343.
- [6] Y. Shao, J. Mao, Y. Liu, W. Ma, K. Satoh, M. Zhang, S. Ma, BERT-PLI: modeling paragraph-level interactions for legal case retrieval, in: Proc. IJCAI, 2020, pp. 3501–3507.
- [7] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutopoulos, LEGAL-BERT: the muppets straight out of law school, CoRR abs/2010.02559 (2020).
- [8] H. Nguyen, P. M. Nguyen, T. Vuong, Q. M. Bui, C. M. Nguyen, T. B. Dang, V. Tran, M. L. Nguyen, K. Satoh, JNLP team: Deep learning approaches for legal processing tasks in COLIEE 2021, CoRR abs/2106.13405 (2021). URL: <https://arxiv.org/abs/2106.13405>. arXiv:2106.13405.
- [9] M. Yoshioka, Y. Aoki, Y. Suzuki, BERT-based ensemble methods with data augmentation for legal textual entailment in COLIEE statute law task, in: Proc. Int. Conf. on Artificial Intelligence and Law (ICAIL), ACM, 2021, pp. 278–284.
- [10] S. Gururangan, A. Marasovic, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don’t stop pretraining: Adapt language models to domains and tasks, in: Proc. Annual Meeting of the Association for Computational Linguistics (ACL), ACL, 2020, pp. 8342–8360.
- [11] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, V. Basile, ALBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets, in: Proc. 6th Italian Conf. on Computational Linguistics (CLiC-it), volume 2481 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019.
- [12] D. Puccinelli, S. Demartini, R. E. D’Aoust, Fixing comma splices in italian with BERT, in: Proc. 6th Italian Conf. on Computational Linguistics (CLiC-it), volume 2481 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019.
- [13] F. Tamburini, How “BERTology” Changed the State-of-the-Art also for Italian NLP, in: Proc. 7th Italian Conf. on Computational Linguistics (CLiC-it), volume 2769 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020.
- [14] A. Tagarelli, A. Simeri, Unsupervised law article mining based on deep pre-trained language representation models with application to the Italian civil code, *Artif. Intell. Law* 30(3) (2022) 417–473. Published: 15 September 2021. doi:10.1007/s10506-021-09301-8.
- [15] D. Licari, G. Comandè, ITALIAN-LEGAL-BERT: A Pre-trained Transformer Language Model for Italian Law, in: Proc. KM4LAW Workshop with the 23rd Int. Conf on Knowledge Engineering and Knowledge Management, volume 3256 of *CEUR Workshop Proceedings*, CEUR, 2022.

- [16] L. La Cava, A. Simeri, A. Tagarelli, The Italian Civil Code network analysis, in: Proc. RELATED – Relations in the Legal Domain Workshop @ICAIL 2021, volume 2896 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 3–16.
- [17] L. La Cava, A. Simeri, A. Tagarelli, LawNet-Viz: A Web-based System to Visually Explore Networks of Law Article References, in: Proc. 45th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, ACM, 2022, pp. 3300–3305. doi:10.1145/3477495.3531668.
- [18] A. Tagarelli, A. Simeri, LamBERTa: Law Article Mining Based on Bert Architecture for the Italian Civil Code, in: Proc. 18th Italian Research Conference on Digital Libraries, volume 3160 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.
- [19] K. Branting, B. Weiss, B. Brown, C. Pfeifer, A. Chakraborty, L. Ferro, M. Pfaff, A. S. Yeh, Semi-supervised methods for explainable legal prediction, in: Proc. Int. Conf. on Artificial Intelligence and Law (ICAIL), 2019, pp. 22–31.
- [20] P. Hacker, R. Krestel, S. Grundmann, F. Naumann, Explainable AI under contract and tort law: legal incentives and technical challenges, *Artif. Intell. Law* 28 (2020) 415–439.
- [21] M. T. Ribeiro, S. Singh, C. Guestrin, “Why Should I Trust You?”: Explaining the Predictions of Any Classifier, in: Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1135–1144. doi:10.1145/2939672.2939778.
- [22] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, I. Androutsopoulos, LEGAL-BERT: the muppets straight out of law school, *CoRR abs/2010.02559* (2020).
- [23] S. Geng, R. Lebrecht, K. Aberer, Legal Transformer Models May Not Always Help, *CoRR abs/2109.06862* (2021).