Few-Shot Legal Text Segmentation via Rewiring Conditional Random Fields: a Preliminary Study

Alfio Ferrara[1 [0000−0002−4991−4984]], Sergio Picascia[1 [0000−0001−6863−0082]], and Davide Riva[1 [0009−0003−9681−9423]]

Università degli Studi di Milano
Department of Computer Science
Via Celoria, 18, 20133 Milano, Italy

Abstract. Functional Text Segmentation is the task of partitioning a textual document in segments that play a certain function. In the legal domain, this is important to support downstream tasks, but it faces also challenges of segment discontinuity, few-shot scenario, and domain specificity. We propose an approach that, revisiting the underlying graph structure of a Conditional Random Field and relying on a combination of neural embeddings and engineered features, is capable of addressing these challenges. Evaluation on a dataset of Italian case law decisions yields promising results.

Keywords: Text Segmentation · Legal Document Processing · Conditional Random Fields.

1 Introduction

Several legal systems around the world are undergoing a complex process of digital transformation, in which a pivotal role is played by the digitization and automated processing of legal documents. Court decisions and law codes are known to be long and articulated documents, lacking of standard rules and conventions defining their structure.

Text Segmentation is concerned with the ex-post recognition of the structure of a textual document, which may be related to the discussed topics, *topical segmentation*, or to the functions that each segment plays in the text, *functional segmentation*. Here we look particularly at the latter, as an important contribution in the legal domain to provide valuable information to several downstream tasks, thus becoming a critical component of a complete Information Extraction pipeline [2]. However, we acknowledge the need to overcome 3 key challenges that concern the legal domain: (a) the articulated structure of documents comprises the possibility that parts of text which perform a single function are discontinuous within the text; (b) models need to operate in a *few-shot* scenario, characterized by the scarcity of annotated data and strong label imbalance; (c) specificity of the legal jargon, with its own semantics and syntax deviating from common language, requires expert knowledge for proper understanding.

Given the scarce availability of annotated data, in this study we explore 3 solutions that modify a Linear-Chain Conditional Random Field (CRF) model to address the 3 challenges outlined above. In particular, we propose adding connections between non-consecutive portions of text to deal with (a) while keeping the number of parameters feasible. To tackle (b), we inject prior statistical information into the model. Finally, we measure the impact of domain-specific feature extraction models to confront (c). The resulting model is evaluated on a dataset of Italian court decisions which underwent manual segmentation by part of legal experts in the context of the *Next Generation UPP* Project, funded by the Italian Ministry of Justice.

The paper is organized as follows: Section 2 describes previous work on Text Segmentation with a focus on the legal domain; Section 3 presents the proposed approach; Section 4 contains an evaluation of the approach on the dataset of Italian court decisions; and Section 5 discusses the conclusions and future work.

2 Related Work

Text Segmentation (TS) consists in partitioning a text into coherent portions called *segments*. Coherence may be interpreted from a topical, semantic, structural, or functional perspective, and criteria for partitioning are highly dependent on the given interpretation. TS approaches can be categorized as *linear* or *hierarchical* [5], where a linear approach sees a textual document as a sequence of segments [7], while a hierarchical approach splits segments at several levels, down to a predefined granularity [4]. From another point of view, we can distinguish between *region-oriented* approaches, that aim at detecting segment boundary position [12], and *class-oriented* approaches that classify each text unit (be it a paragraph, sentence, clause, or even a single word) into a segment type [11].

In the legal domain, Aumiller et al. [1] advocated for topical TS as a way to improve downstream applications, such as information retrieval and document summarization. However, we argue that *functional* TS may often be more relevant than topical TS in the legal domain, since the same information may be more valuable when found in certain parts of the document (e.g. highly argumentative parts) rather than others (e.g. introductory parts). Functional TS of legal documents has been addressed in the context of judgements by the US Security and Exchange Commission, using CRFs [11], as well as Italian court decisions, adding a sentence-level mean-pooling layer and feed-forward neural network on top of a BERT model fine-tuned for Italian legal language [9]. Both approaches, as well as ours, fall in the *linear*, *class-oriented* category; however, it must be noticed that approaches are hardly portable from one application to another, due to the development of different segmentation schemas for different legal documents and the subsequent need to annotate a sufficient amount of documents to re-train models. Furthermore, supervised approaches like the ones in this category often have to deal with a few-shot scenario with significant label imbalance.

Inspired by GraphSeg [5], our model builds a sentence relatedness graph and trains a classifier on it. However, to capture functional instead of semantic relatedness, we adopt a supervised approach to graph construction. In the same way, we train a CRF to classify graph nodes with manually defined labels instead of relying solely on an unsupervised clustering algorithm, which would produce an unlabeled classification. Thus, our work is, to the best of our knowledge, the first to explore functional TS of legal documents in a few-shot scenario, simultaneously addressing domain specificity, segment discontinuity, as well as class imbalance.

3 Methodology

The proposed approach to Text Segmentation is displayed in Figure 1.[1]

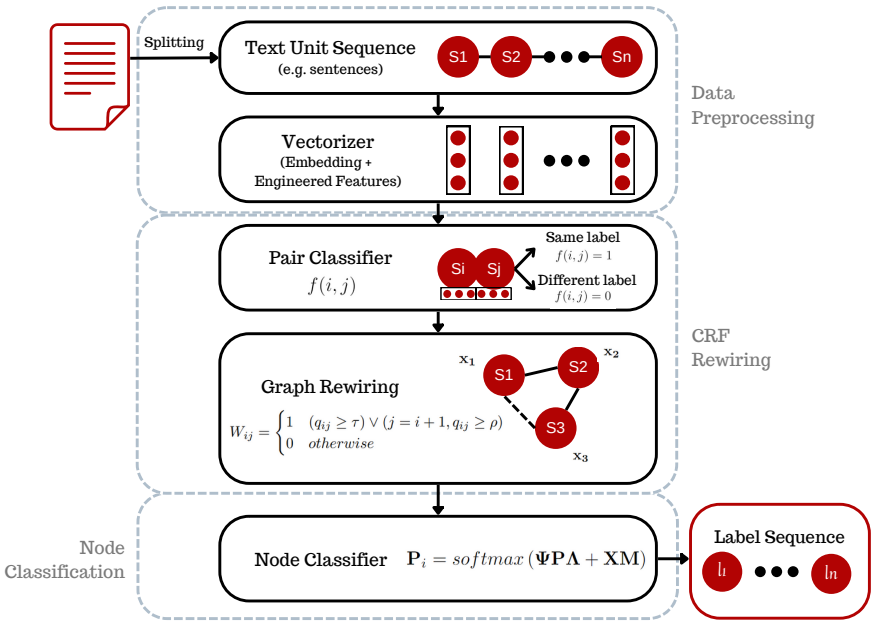

Fig. 1: Workflow of the proposed approach.

3.1 Data Preprocessing

TS models typically operate on sequences of text units, which can be single words, phrases, clauses, sequences or even paragraphs according to the granularity level that is appropriate for the application at hand.

[1] A Python package will be made available at `https://github.com/umilISLab/TextSegmentation`

After being split, each text unit s_i is represented as a d-dimensional vector $\mathbf{x}_i$ using a Vectorizer. The Vectorizer concatenates the embedding vectors generated by a neural embedding model (e.g. BERT [3]) with additional, engineered features that have been found to be informative in legal text structure [6]. Such variables are the counts of verb moods and tenses, the number of references to law articles and previous judgements, the sentence length, and the sentence position in the document[2].

3.2 CRF Rewiring

Given a graph $G = (V, E)$ with N nodes where each node v has an associated feature vector $\mathbf{x}_v$ and a label y_v with value in label set $L = \{l_1, \ldots, l_K\}$, a CRF [8] is a discriminative graphical model whose general form is written as:

$$\mathbb{P}(y_i = l_k|\mathbf{X}, G) = \frac{1}{Z} exp\left(\sum_{e \in E} \lambda_k t_k(e, \mathbf{X}, \mathbf{y}) + \sum_{v \in V} \mu_k s_k(\mathbf{x}_v, y_v)\right) \quad (1)$$

where $t_k(e, \mathbf{X}, \mathbf{y})$ are *transition functions* over edges e of the graph, possibly depending on the labels of all nodes $\mathbf{y} = (y_1, \ldots, y_N)$ as well as their feature vectors $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, $s_k(\mathbf{x}_v, y_v)$ are *state functions* over node v, depending solely on the node features, λ_k and μ_k are weights to be learnt and Z is a normalization factor.

Our approach starts from an undirected, Linear-Chain CRF, which is typically employed in sequential labelling problems as it assumes a graph structure with edges only between consecutive nodes, i.e. $E = \{(v_i, v_{i+1}) : i = 1, \ldots, N\}$. The hypothesis underlying our work is that modifying such graph structure by adding/removing edges between appropriately chosen nodes can improve performance on TS tasks characterized by segment discontinuity.

To do so, we employ a *Pair Classifier*, which is a binary classifier trained to predict the probability that a pair of text units (s_i, s_j) share the same segment label, based on the concatenation of their vector representations $(\mathbf{x}_i, \mathbf{x}_j)$. Formally, given the function:

$$f(i, j) = \begin{cases} 1 & y_i = y_j \\ 0 & otherwise \end{cases} \quad (2)$$

we estimate the probability $q_{ij} = \mathbb{P}(f(i, j) = 1|\mathbf{x}_i, \mathbf{x}_j)$.

A natural advantage of employing such a model rather than a model that directly aims at predicting segment labels is that the Pair Classifier has at its disposal $\binom{N}{2}$ data for training from a single document made of N text units, thus mitigating possible overfitting issues.

The probabilities q_{ij} predicted by the Pair Classifier are then exploited to rewire the linear-chain graph structure based on the following two rules:

[2] Notice that the proposed model is capable of working with any vector representation of the analyzed text units.

– an edge is added between two non-consecutive nodes (v_i, v_j) if $q_{ij} \geq \tau$;
– an edge between two consecutive nodes (v_i, v_{i+1}) is removed if $q_{i,i+1} < \rho$.

An example of the procedure is provided in Figure 2.

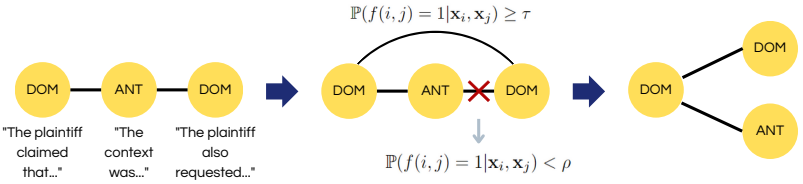

Fig. 2: Example of rewiring with 3 text units having labels DOM, ANT and DOM again (see Section 4.1 for details). First, we have a linear-chain structure, then the Pair Classifier evaluates the probability of each connection, and finally one new connection is added between the two DOM-labelled units and an existing one is removed, complying with probability thresholds.

In order to favor the flow of information in the sequence of units $(s_1, \ldots, s_N)$ with respect to "residual" connections between non-consecutive ones, we recommend $\rho \leq \tau$. The larger the difference, the more sequential connections are favored. A stricter constraint on residual connections makes passage of information possible between text units that belong to the same segment with a high degree of certainty. Finally, self-loops are added by default for each node.

The graph $\tilde{G}$ resulting from this rewiring operation will have an adjacency matrix $\mathbf{W}$ such that:

$$W_{ij} = \begin{cases} 1 & (q_{ij} \geq \tau) \vee (j = i+1, q_{ij} \geq \rho) \\ 0 & otherwise \end{cases} \tag{3}$$

Fringe cases are $\tau = \rho = 0$, which yields a fully connected graph, and $\tau = \rho = 1$, for which a link appears only for completely certain connections.

3.3 Node Classification

Once the rewired graph $\tilde{G}$ has been constructed, a CRF classifier is applied for node classification:

$$\mathbf{P}_i = softmax\left(\mathbf{\Psi}\mathbf{P}\mathbf{\Lambda} + \mathbf{X}\mathbf{M}\right) \tag{4}$$

where P_{ik} denotes the probability of label l_k for node v_i, $\mathbf{\Psi} \in [0,1]^{N \times N}$ is the transition matrix of the rewired graph, obtained by row-normalization of the adjacency matrix, $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the feature matrix of all nodes, and $\mathbf{\Lambda} \in \mathbb{R}^{K \times K}$ and $\mathbf{M} \in \mathbb{R}^{d \times K}$ are the weight matrices to be learnt. Since Eq. 4 is

implicit, having $\mathbf{P}$ on both the left-hand and right-hand sides, we initialize $\mathbf{P}$ and compute it iteratively relying on row-stochasticity of $\mathbf{\Psi}$ to ensure convergence.

To address the problem of class imbalance, we propose a combination of two solutions. The first is *loss weighting*, which consists in weighting each segment label l_k in the training phase by a factor $\frac{N}{N_k K}$, where K denotes the cardinality of the label set and N_k is the number of text units in the training set labelled with l_k. The second solution is to inject prior *positional knowledge* of labels, where available, in the initialization of $\mathbf{P}$.

4 Evaluation

In this section we evaluate our approach to Text Segmentation on a real dataset of Italian court decisions, which underwent manual annotation in the context of the *Next Generation UPP* Project, funded by the Italian Ministry of Justice.

4.1 Dataset

The dataset employed for the evaluation consists of 50 Italian case law decisions, retrieved from 12 Courts in Northern Italy and concerning first degree civil law judgements on the matter of unfair competition.[3] The documents have been manually annotated with functional segments by a group of 9 legal experts through an interactive annotation activity. We refer to [13] for further details. The annotation schema, presented in Table 1, comprises five segment labels, representing the functional role a segment plays in the legal document. A NULL label is automatically added to indicate segments with no specific function.

Labels	**Italian**	**Explanation**
COR	Corte e Parti	Court, judicial panel, parties
ANT	Antefatto	Background information
DOM	Domande	Claim(s) and argumentation of the parties
MOT	Motivazione	Reason(s) for the final decision(s)
DEC	Decisione	Final decision(s)

Table 1: The annotation schema for Italian court judgements of civil proceedings.

The dataset contains annotated segments in the form of quintuples (document ID, start, end, text, label), where start and end are integers indicating a character position in the document, while label belongs to the set of labels. Segments cannot overlap with one another. Moreover, since at the present moment there is no rule strictly imposing the structure of a court decision in Italian judicial system, segment labels have no semantic nor order relationship with one another. The only two types of segments for which we have reasonable a-priori

[3] The dataset cannot be made public at the moment due to privacy-related constraints. An anonymization process to make the data public is ongoing.

positional knowledge are COR, expected at the beginning of a document, and DEC, expected at the end.

For the purpose of our experiments, among the annotated 50 documents, we filter the 38 court decisions with a complete annotation. Each document is split in sentences, which are the text units considered by our model. Since segment boundaries resulting from the manual annotation activity do not always match sentence boundaries, we adopt a heuristic approach to label sentences by assigning label y to a sentence s, if s overlaps by at least $1/2$ of its length (as number of characters) with a segment labeled with y. No sentence in the dataset is equally shared between two segments with different labels, and, in case a sentence overlaps with two such segments, the difference in overlapping lengths is never under 20% of the sentence length. The resulting dataset, which undergoes no further preprocessing, contains documents having an average of ~ 121 sentences (s.d. ~ 54) and ~ 4114 words (s.d. ~ 1821). Label imbalance is evident in that an average of ~ 56 sentences per document are labelled as MOT, while only ~ 7 with DOM, with counts of other labels in between.

Since Inter-Annotator Agreement metrics showed the heterogeneity of annotations, hinting at its intrinsic complexity and subjectivity, we regard a perspectivist approach as preferable. Perspectivist approaches to ground-truthing consist in refraining from the definition of a univocal ground truth so to capture the perspective of different annotators, insofar as they are deemed reliable. As a consequence, in case a single segment is assigned two different labels by two different annotators, we do not aim at solving discrepancies. As an example, the sentence *"All issues are outweighed by the current term of protection (70 years after the author's death)"* [4] is labelled by one annotator as ANT and by another as MOT, thus the two labels are assigned 50% probability each.

4.2 Setting

We implement our model adopting a feed-forward neural network with a single hidden layer as Pair Classifier, and train both the Pair Classifier and the Node Classifier with the AdamW algorithm [10] to minimize cross-entropy loss.

We study the impact of domain-specific features by experimenting with 3 different embedding models: a transformer model fine-tuned on Italian legal documents called `Italian-Legal-BERT`[5] [9] with mean pooling (ILB), a Sentence-BERT model pre-trained on Italian language (SBERT)[6], and a non-neural model based on TFIDF with ICA dimension reduction so to match the output dimensionality of the other models, i.e. 768. We experiment with a sparse ($\tau = 0.95$), a dense ($\tau = 0.65$), and an intermediate rewiring ($\tau = 0.80$), keeping in all cases $\rho = \frac{\tau}{2}$. Positional knowledge injection for addressing label imbalance was obtained by initializing $\mathbf{P}$ so that the probability of label COR for the first sentence

[4] Traslated from Italian.

[5] Model available at `huggingface.co/dlicari/Italian-Legal-BERT`

[6] Model available at `huggingface.co/nickprock/sentence-bert-base-italian-xxl-uncased`

and that of label DEC for the last sentence of each document is set to 1. To evaluate how our approach deals with segment discontinuity and class imbalance, we compare it against two baseline models in a 4-fold Cross Validation framework:

- a ONE-BY-ONE classifier, which takes the vector representation of each sentence individually and trains a shallow classifier (e.g. Random Forest) to predict its label;
- a SEQUENTIAL model, equivalent to our model before rewiring, i.e. a Linear-Chain CRF[7].

4.3 Results

Results, in terms of average F_1 scores, are presented in Table 2.

		Embedding			Embedding + Features		
		ILB	SBERT	TFIDF	ILB	SBERT	TFIDF
ONE-BY-ONE		0.547	0.523	0.505	0.582	0.558	0.599
SEQUENTIAL		0.622	0.547	0.555	**0.656**	0.607	0.601
REWIRED (Ours)	Sparse	**0.650**	0.572	**0.556**	**0.656**	0.559	**0.616**
	Intermediate	0.627	0.560	0.521	0.530	**0.621**	0.591
	Dense	0.620	**0.587**	0.518	0.554	0.543	0.587

Table 2: 4-fold Cross Validation average F_1 scores.

F_1 scores are naturally informative for what concerns (i) the capacity of a model to deal with segment discontinuity and (ii) the impact of domain-specific features.

For (i), we notice little improvement from the rewiring operation, whose outcome is closely comparable with the one of the SEQUENTIAL model. In general, rewiring that produces a more sparse graph is preferable, while a dense structure is even detrimental, up to a 0.1 difference in F_1 scores, and performs worse than ONE-BY-ONE models.

For (ii), results show a consistent improvement when an model fine-tuned on domain data (Italian-Legal-BERT) is used. Nevertheless, including the additional, engineered features discussed in 4.2 seem to provide an equally consistent improvement, contributing with information that may not be captured by the embedding model. Indeed, while transition functions $\mathbf{\Psi P \Lambda}$ have higher coefficients (in absolute value) with respect to state functions $\mathbf{XM}$, additional features play an important role. For instance, verbs at future tense and verbs at subjunctive mood are always among the 10 features with the strongest influence on the prediction of DOM.

As error analysis, Figure 3 shows the confusion matrix between predicted and ground truth labels.

[7] We employed CRFSuite implementation for this model, available at `www.chokkan.org/software/crfsuite`

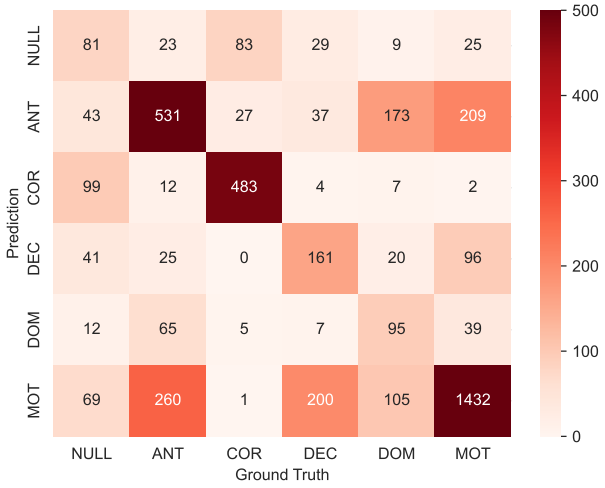

Fig. 3: Confusion matrix.

Besides the label imbalance, it can be appreciated that confusion between segments labelled with COR and DEC, which are expected at the opposite ends of a document, is minimal. Conversely, ANT, MOT and DEC-labelled segments tend to intertwine the most, since final decisions are often introduced and background information is often recalled within the reasoning section. Indeed, even annotators had the highest disagreement on label pairs (MOT, DEC) and (DOM, ANT). [13]

5 Conclusion and Future Work

In this paper we addressed the Text Segmentation task proposing a model that, revisiting the underlying graph structure of a Conditional Random Field, aims at handling segment discontinuity and operates in a few-shot scenario with scarcity of data. Moreover, we experimented with domain-specific embedding models and engineered features in order to capture more information than we would have with a general embedding model. The preliminary results confirm our hypothesis that domain-specific, engineered features provide useful information to the model, and that our methods to tackle label imbalance are effective.

This work will serve as basis for the integration of a segmentation module in a complete information retrieval pipeline tailored to the legal domain, for instance for document retrieval and building (see [2] for more). To achieve such objective, we aim at constructing a model which has generalization, scalability and domain specificity as its key characteristics.

Acknowledgements This work is partially supported by the Next Generation UPP project within the PON programme of the Italian Ministry of Justice and by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NextGenerationEU.

References

1. Aumiller, D., Almasian, S., Lackner, S., Gertz, M.: Structural text segmentation of legal documents. Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law (2020)
2. Castano, S., Ferrara, A., Montanelli, S., Picascia, S., Riva, D.: A knowlegde-based service architecture for legal document building. In: 2nd International Workshop on Knowledge Management and Process Mining for Law. Sherbrooke, Quebec, Canada (2023)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
4. Glavas, G., Ganesh, A., Somasundaran, S.: Training and domain adaptation for supervised text segmentation. In: Workshop on Innovative Use of NLP for Building Educational Applications (2021)
5. Glavas, G., Nanni, F., Ponzetto, S.P.: Unsupervised text segmentation using semantic relatedness graphs. In: International Workshop on Semantic Evaluation (2016)
6. Grover, C., Hachey, B., Korycinski, C.: Summarising legal texts: Sentential tense and argumentative roles. In: HLT-NAACL 2003 (2003)
7. Koshorek, O., Cohen, A., Mor, N., Rotman, M., Berant, J.: Text segmentation as a supervised learning task. In: North American Chapter of the Association for Computational Linguistics (2018)
8. Lafferty, J.D., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: International Conference on Machine Learning (2001)
9. Licari, D., Comandè, G.: Italian-legal-bert: A pre-trained transformer language model for italian law. In: CEUR Workshop Proceedings (Ed.), The Knowledge Management for Law Workshop (KM4LAW) (2022)
10. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2017)
11. Savelka, J., Ashley, K.D.: Segmenting u.s. court decisions into functional and issue specific parts. In: International Conference on Legal Knowledge and Information Systems (2018)
12. Solbiati, A., Heffernan, K., Damaskinos, G., Poddar, S., Modi, S., Calì, J.: Unsupervised topic segmentation of meetings with bert embeddings. ArXiv **abs/2106.12978** (2021)
13. Zanoli, E., Barbini, M., Riva, D., Picascia, S., Furiosi, E., D'Ancona, S., Chesi, C.: Annotators-in-the-loop: Testing a novel annotation procedure on Italian case law. In: Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII). pp. 118–128. Association for Computational Linguistics, Toronto, Canada (2023), https://aclanthology.org/2023.law-1.12