

Hybrid AI Framework for Legal Analysis of the EU Legislation Corrigenda

Monica PALMIRANI^{a1}, Francesco SOVRANO^b, Davide LIGA^a, Salvatore SAPIENZA^a and Fabio VITALI^b

^a *CIRSFID-ALMA-AI, University of Bologna*

^b *DISI, University of Bologna*

Abstract. This paper presents an AI use-case developed in the project “Study on legislation in the era of artificial intelligence and digitization” promoted by the EU Commission Directorate-General for Informatics. We propose a hybrid technical framework where AI techniques, Data Analytics, Semantic Web approaches and LegalXML modelisation produce benefits in legal drafting activity. This paper aims to classify the corrigenda of the EU legislation with the goal to detect some criteria that could prevent errors during the drafting or during the publication process. We use a pipeline of different techniques combining AI, NLP, Data Analytics, Semantic annotation and LegalXML instruments for enriching the non-symbolic AI tools with legal knowledge interpretation to offer to the legal experts.

Keywords. Akoma Ntoso, Classification AI, NLP, legal drafting techniques.

1. Introduction: AI for legislative drafting process

The scope of the “Study on legislation in the era of artificial intelligence and digitization”, promoted by the EU Commission Directorate-General for Informatics, is part of the digital transformation agenda supported by the EU Commission, particularly relevant in this historical moment where the Rules of Law changes quickly due to the COVID-19 special regulation. Companies and society require legal certainty and it is fundamental to implement policies such as “Better regulations”², “Fit for the Future”³, in conjunction with the “evidence-based legislation”⁴ methodology and the “digital-ready policymaking”⁵ approach. With this study we intend to improve the quality of the law-making process and of the content of each legislative regulation by investigating the following features: i) text clarity supporting legal drafters and end-user presentation; ii) linguistic variants and temporal versions management; iii) law-making/policy development process in decision making of the Commission supporting also amendments and consolidation; iv) metadata integration (ELI, ECLI, AKN, CDM, etc.) in the different steps of the law-making process; iv) modelling legal norms expressed in the legislative document; v) facilitation of the implementation of law by the Member

¹ E-mail: {monica.palmirani, francesco.sovrano2, davide.liga, salvatore.sapienza, fabio.vitali}@unibo.it

² https://ec.europa.eu/info/sites/default/files/better_regulation_joining_forces_to_make_better_laws_en_0.pdf

³ https://ec.europa.eu/info/law/better-regulation/have-your-say-simplify_en

⁴ <https://ec.europa.eu/info/sites/default/files/better-regulation-toolbox.pdf>

⁵ <https://joinup.ec.europa.eu/collection/better-legislation-smoother-implementation/digital-ready-policy-making>

States. The study has the following goals: 1) reducing manual/error-prone work using patterns (e.g., corrigenda) and best practices templates during the legal drafting, to automatize as much as possible consolidation and semantic annotation, using legal ontologies and thesauri (e.g., EuroVoc); 2) maximising the reuse of similar legal concept detected using Machine Learning and legal data analytics applied to the whole legal system (e.g., definition, derogation); 3) favouring the implementation of some policies in the legislation (e.g., digital-ready, gender neutrality); 4) increasing transparency up to publication, thus increasing the searchability. In this light we have isolated⁶, three main use-case scenarios and this paper aims to present the preliminary results of the first use case. The first use-case focuses on corrigenda and provides a clustering of them to understand which patterns could help the informatics tools (e.g., LEOS editor) to develop new relevant features to minimize errors and to improve the quality of legislation. This use-case also provides more information to the legal drafter.

2. Corrigenda in the EU Legislation and preliminary taxonomy

Corrigenda is a special modification necessary due to an error occurred in the official publication process. Since under theory of law it is a material error, not substantial, it has immediate efficacy since the beginning of the legislative act. The modifications of corrigenda are thus inserted in the first emission of the text, as if it had never been published differently. Corrigenda involve Directives, Regulation, and Decisions. For this reason, corrigenda need an immediate publication of the modificatory instructions on the official EU Official Journal and they are immediately implemented in the original text. Making a query to CELLAR⁷ we get about 24.000 triples that connect each corrigendum to the document corrected, involving all the 24 official languages of the EU institutions, but only about 8.500 of them are connected to the English language variant. The corrigendum actions can be numerous, sparse across different points of the destinations, and they can also play a different semantic role, not only textual. The aim of this study is to isolate better the portion of the text involved (more granularity), to understand the legal role of the modification (e.g., temporal modification), to evaluate why they are frequent. We have prepared a light taxonomy of the quality of the modificatory instructions (25 classes) grouped in five *macro-areas*:

i) **Structure modifications** (e.g., provisions, annexes, footnotes, recitals, preamble, etc.)

On page 1, footnote 1:

for: '(1) OJ L 145, 13.6.1977, p. 1. Directive as last amended by Directive 2006/98/EC (OJ L 221, 12.8.2006, p. 9).',
 read: '(1) OJ L 145, 13.6.1977, p. 1. Directive as last amended by Directive 2006/98/EC (OJ L 363, 20.12.2006, p. 129).'

ii) **Legal temporal information** (e.g., date of efficacy, date of adoption)

On the cover page, on page 11 and page 12, adoption date:

for: '15 March 2021',
 read: '15 February 2021'.

iii) **Qualified portion of text** (e.g., definitions, references, modification of modifications)

On page 257, point (b) of the first paragraph of Article 112:
 for: '(b) Article 10 and points (a) and (b) of Article 12(1) of Directive 98/79/EC, and ...',

⁶ Two focus groups composed by EU Commission legislative drafting offices, Open Data Office, Parliament of EU, Publication Office of EU were held using a questionnaire.

⁷ <http://publications.europa.eu/webapi/rdf/sparql>

read: '(b) Article 10, points (a) and (b) of Article 12(1) and Article 15(5) of Directive 98/79/EC, and ...'.

On page 98, Article 2(1)(18):
 for: '(18) "competent authority" means a competent authority as defined in Article 2(1)(26) of Directive 2014/65/EU;',
 read: '(18) "competent authority" means a competent authority as defined in Article 4(1)(26) of Directive 2014/65/EU;'.

iv) **Entities** (e.g., role, places, number, organization, etc.)

On page 10, in the column 'COUNTRY OF ISSUE':
 for: 'CZECH REPUBLIC',
 read: 'CZECHOSLOVAKIA'.

v) **Presentational information** (e.g., images, punctuation, publishing information)

On page 89, in the Annex, on the 12th line 'Austria', in the second column:
 for: '343 405 392',
 read: '343 473 407'.

3. Dataset

The first step of the experiment was the dataset selection: all the corrigenda files in Formex 4.0, in English language, with the corresponding original file. The total number of corrigenda files is 2.513 documents, 3.478 pairs of modifier and modified text. The words in the old text are 87.906 and the words in the new text are 100.416. The average of the modifications for each correcting document is 1,81, but even corrigenda with 77 instructions of modification can be found. The second step was to convert these files in Akoma Ntoso including the CELLAR RDF information inside of a unique XML file that, despite not perfectly marked-up, is valid against the AKN-XSD schema or matches perfectly the AKN4EU specifications. This second step allows to have context, normative references, temporal parameters, metadata (e.g., ELI), modifications annotation qualifications in a unique consistent XML format. Publication Office supported the team of University of Bologna with the extraction operations.

4. Methodology

The methodology used in this work combines unsupervised clustering K-means enriched with Akoma Ntoso annotation and light-taxonomy information. At the end it is a mix of annotated text and unsupervised classification. Differently to many other research in the same field, we want to foster the structure information of the legal document (e.g., articles) and the light-taxonomy extracted using classic NLP techniques. Machine Learning (ML) approaches can classify a part of the legal text as 'definition', or a 'modification', or detect the 'date' included in the sentence but connecting all this information in a meaningful manner is quite difficult. Additionally, the same corrigendum could be classified in different ways: it can be a temporal modification, a table modification, or a definition modification. We intend to go beyond a pure classification methodology and to group in cluster the corrigenda modifications using the destination type (table, annex, normative provision, footnote, etc.), the type of modification (substitution, insertion, repeal), the text modified in relation with the old

text (when it is present), the role of the text modified (e.g., definition) and the temporal parameters (e.g., date of application). For this reason, the methodology is called hybrid and it mixes annotated validated information and unsupervised AI techniques. The mix of the two could permit to obtain a more semantic clustering that can be closer to the legal needs of the domain. The clustering may help the end-user and the tools to avoid the mistakes that produced the corrigenda. For permitting the interpretation we used KNIME as Data Analytics tool for comparing the clustering with some parameters: similarity distance, typology, granularity of the text of destination involved in the modification, and the typology of the document.

5. Hybrid Pipeline

The pipeline uses a hybrid approach, and it is composed by following steps:

a) **Preliminary light-taxonomy of the corrigenda**: legal experts have analysed a random sample of corrigenda with a good balance between years and then they have created an agnostic taxonomy of the main modificatory events that is used by the technical team as the light-taxonomy needed for the classification. Legal experts have identified also good signals in the text for classifying the corrigenda using regular expressions. We have identified 25 classes; b) **Conversion in Akoma Ntoso**: we have converted corrigenda documents from Formex 4.0 in Akoma Ntoso using Python and RegEx; c) **Classification of the Corrigenda**: using simple NLP signatures we have classified the corrigenda using a light-taxonomy and the metadata of Akoma Ntoso. In this way we have assigned the qualification of each modification (e.g., substitution, insertion, repeal); d) **Clustering of the Corrigenda**: we have created clusters of the corrigenda using K-means algorithm techniques; e) **Distance of the text calculation**: we have calculated the distance between the old text and the new text using the Levenshtein distance; f) **Data Analytics**: this step combines the results of the previous ones with AKN information to explain by user interfaces some interpretations, statistics, analyses using KNIME; g) **Evaluation**: we set up a legal expert team composed by three members: two members check, and the third supervises them and resolves conflicting interpretations. The goal of this step is to evaluate the results of the clustering and of the Data Analytics work; h) **Legal interpretation**: the legal experts use the diff-text and the graphs of the user-interface for providing a legal interpretation. In this step we also refine the light-taxonomy adding legal meaning. The same error could have different meanings and semantics depending also on the topic, so the legal interpretation is a fundamental part of the research.

6. Related Work

We have already converted several pre-existing document collections [10][14] in Akoma Ntoso, developed different NLP tools using patterns and RegEx rules [6][7] for extracting legal knowledge from the text (e.g., normative citations), classified legal text using ML or Deep Learning (DL) techniques [8][15]. Other researches have demonstrated the effectiveness of the ML/DL in the legal documentation fields [3][16][17][18][11] but without including the necessary semantic information for completing the context. The innovative approach in this work is to use hybrid architecture that uses unsupervised approach adding semantics [4][5] to the clustering results using light-taxonomy, NLP extraction, Data Analytics. The aim is to interpret the

output with the legal knowledge information supported by other techniques. We use data analytics tool (KNIME platform⁸) for providing information necessary to detect some best practices to suggest to legal drafters and software designers.

7. Akoma Ntoso Conversion of the corrigenda

We have converted Formex 4.0 in Akoma Ntoso in order to reach the following goals: a) **to detect the granular citations of the destination**. In Formex 4.0 this information is not present, and we have parsed the normative citations for representing the correct destination (e.g., article 23, paragraph 3, point a). This is relevant in order to provide the context of the semantic action of corrigendum. b) **to convert the modifications** in metadata that are not represented in Formex 4.0. The attributes `@period` that qualifies the span of time when the modification is valid, `@old` and `@new` that are also present in Formex 4.0 and the `@destination` with a precise specification.

8. Unsupervised Corrigenda Clustering

The pipeline we adopted to analyse the corrigenda consists in three main phases: i) *Feature Identification*; ii) *Dimensionality Reduction*; iii) and *Clustering*. During the *Feature Identification* phase, we selected the pieces of information to be considered for clustering. In the present case, we opted for the following features, deemed to bear enough information to (unsupervisedly) push the clustering algorithm towards the structure of our taxonomy: a) the difference between the embeddings (representative numerical vectors obtained via [2]) of the corrigens and the corrigenda. This is crucial to the clustering on the semantic contents of the modifications; b) a set of booleans that indicate whether the description contains the keywords 'table', 'annex', 'recital', 'title', 'note'. This is used for clustering on basis of the modification's description. Considering that the resulting number of features may be large, depending on the characteristics of the embedding function that is used, we then perform one step of *Dimensionality Reduction*. *Dimensionality Reduction* is quite commonly used in conjunction with further clustering techniques, to foster better clusters. In our specific case, the number of features (about 773 in total, between features *a* and *b*) is arbitrarily reduced (to 50), removing the less significant ones through Principal Component Analysis⁹. Finally, after the *Dimensionality Reduction* we perform one step of automated *Clustering*. In our case, K-Means¹⁰ is applied in an attempt to extract 25 different classes. The number is 25 because our reference taxonomy consists of 25 classes and the goal is to extract a clustering that is possibly aligned to it (see [Figure 1](#)). Twenty-four clusters are detected, and the most numerous clusters are C4 and C19.

9. Levenshtein Distance

We noticed also that the corrigenda often use significant portions of text, usually structured in hierarchical normative provision (e.g., article, paragraph, point), even if the real modification is limited to a few characters. For this reason, we have calculated the

⁸ <https://www.knime.com/knime-analytics-platform>

⁹ PCA is not necessarily the best technique to use, other can be envisaged.

¹⁰ DBSCAN, OPTICS and many other clustering techniques appeared to not work very well in our case, making impossible to specify the final number of clusters to extract.

Levenshtein Distance (LD)¹¹ and we have discovered that than 81,4% involves parts of the text in excess respect the real needs (between 0,6 and 1). To evaluate the correctness of the Levenshtein distance the legal experts checked the text using a naïf diff algorithm written in Python for permitting a correct visualization to the legal expert team in agnostic way and not influenced by the previous tool of classification. We have also taken the Levenshtein distance, and we have made a comparison with other parameters including the type of provision of the text modified. Ultimately, we have noticed that the big partitions like article, table, annex, recital have a high index of LD (higher than 80%) with respect to little portions of text such as heading, number, reference.

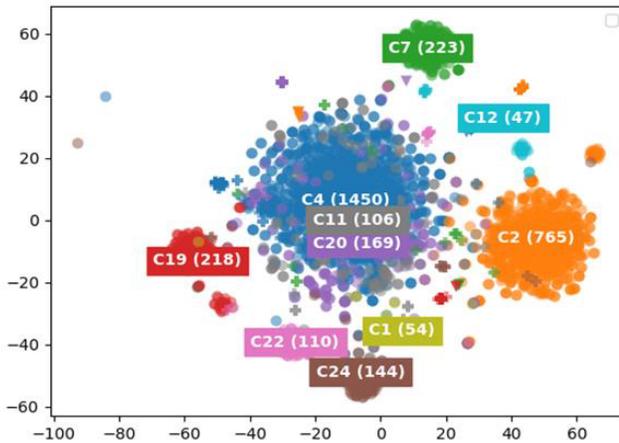


Figure 1 – Visualisation of the clusters we automatically extracted. This visualisation is obtained with tSNE.

10. Data Analytics

We added also other parameters of the data analysis with the goal to interpret the clustering made by the unsupervised algorithm. We have analysed the type of the modifications, and we have noticed a relevant concentration in the period 2004-2009, in correspondence of some of the most intensive period of the EU institutions (e.g., 2004 enlargement to ten new countries, 2009 Lisbon Treaty). We have also investigated the relationship between clusters, partition type and type of document and we have found a relevance between partition. It is contrariwise not influenced by the type of document even if “Regulation” is the higher for occurrences. For instance, cluster C01 seems to be aligned on footnotes. Since this interpretation was not entirely satisfactory, we opted to make the supervised follow-up annotation experiment.

11. Supervised Experiment

We built a dataset of 199 annotated corrigenda, according to the 25 identified classes. The corrigenda were randomly selected by one legal expert and then manually cross-annotated by two legal experts by relying on the 25 classes. The resulting dataset defines a multi-label text classification task. Considering the plethora of existing classifiers and the complexity of finding the right one, with the right configuration, we

¹¹ 0 means that old and new text diverge, 1 means that old and new are identical.

designed a tool that automatically searches for the best classifier within a pre-defined search space. This tool evaluates each possible classifier with a k-fold cross-validation (in our case, $k=4$). Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split. Each classifier is trained for a maximum of 10 epochs (it sees the same data a maximum of 10 times). The classifier with the highest F1-Macro (average calculated on k -folds) is kept as the best. The features considered by the classifier are: i) the difference between the embedding of the corrigenda and the corrigens; ii) a vector of 6 booleans that indicates if particular keywords can be found in the description: amend, recital, title, note, annex and table. The embeddings of the corrigenda and the corrigens were obtained through a deep language model called paraphrase-mpnet-base-v2 [19]. As classifier, we tried 1500 configurations of hyper-parameters of a shallow neural network with one regularised hidden layer of u units. We used a shallow neural network and a $k=4$ because the size of the dataset was small, therefore using a too large k would have resulted in very small test-sets, while a deep neural network would have clearly overfitted. These configurations were tested with an Async HyperBand Scheduler [20] performing a grid search on the following hyper-parameters: 1) batch size (2,3,4); ii) units (4,6,8,10,12); iii) activation function (None, relu, leaky_relu, selu, tanh); iv) learning rate (0.3, 0.1, 0.03, 0.01); v) regularisation strength (0.01, 0.003, 0.001, 0.0003, 0.0001). The best results were given by the following configuration: batch size: 3; units: 4; activation function: None; learning rate: 0.03; regularisation strength: 0.0001. This means that a linear classifier (activation function: None) suffices with the feature we used, and no complex deep learning models are needed. This linear classifier produced the following average results over the 4 folds:

- 1) F1-macro (the average F1-score for each class): 0.076 ± 0.001 ;
- 2) F1-weighted (the average F1-score for each class, weighted by its representativeness): 0.904 ± 0.007 .

These results show that the dataset is unbalanced, meaning that some classes do not have enough datapoints, so the algorithm is not capable to recognize them. In fact, 12 of the 25 classes have less than 10 samples in the dataset, being significantly under-represented. Nonetheless, the algorithm can classify correctly the most represented classes.

12. Conclusions

Our conclusions¹² can be summarised as follows: 1) too much text is involved in the corrigenda that could produce new errors and it is then very difficult for the end-user to detect the new part of the text involved in the corrigendum. Also, the consolidated text offered by the EUR-LEX service is not granularly annotated and the legal expert needs to read in comparative manner the two texts; 2) the clustering operates on the basis of the type of provision involved in the modification and the type of modifications (e.g., C4 is mostly modifications at article level and with modification of the meaning); 3) the statistics detected an intense period of modifications between 2004 and 2009 and it is also natural considering the relative figures of the total number of legal documents emitted in this interval of time. We need to elaborate these findings to transform the

¹² See the dataset, the software, the output in <https://gitlab.com/CIRSFID/AI4LegalDrafting>

outputs in a policy to be provided to the legal drafters, decision-makers and to the technical team for improving the quality of the legislation. This work underlines also the difficulty to provide an interpretation and sound evidence of the meaning of the results coming from unsupervised ML and confirmed the hypothesis that a supervised hybrid architecture could help also in the task of explaining AI for a better transparency.

Acknowledgements. We thank Digit for Informatics of the European Commission - Study on ‘Drafting legislation in the era of AI and digitization’- for supporting our research, and the European Publication Office for the extraction of data using CELLAR.

References

- [1] AKN4EU https://ec.europa.eu/isa2/news/akoma-ntoso-eu-akn4eu-version-30-has-been-published_en
- [2] Akoma Ntoso Version 1.0. Part 1: XML Vocabulary <http://docs.oasis-open.org/legaldocml/akn-core/v1.0/akn-core-v1.0-part1-vocabulary.html>
- [3] Branting K., B. Weiss, B. Brown, C. Pfeifer, A. Chakraborty, L. Ferro, M. Pfaff, and A. Yeh. (2019). Semi-Supervised Methods for Explainable Legal Prediction. In *ICAIL '19*. Association for Computing Machinery, New York, NY, USA, 22–31.
- [4] Casanovas P. et al. (2016). Semantic Web for the Legal Domain: The Next Step. 1 Jan. 2016: 213-227.
- [5] Francesconi E., Küster M., Gratz P., Thelen S. (2015). The Ontology-Based Approach of the Publications Office of the EU for Document Accessibility and Open Data Services. *EGOVIS2015*: 29-39.
- [6] Gianfelice D., Lesmo L., Palmirani M., Perlo D., Radicioni D. P. (2013). Modificatory Provision Detection: a Hybrid NLP Approach, in: *ICAIL2013 Proceedings*, ACM New York: 43-52.
- [7] Liga D., Palmirani M. (2019). Classifying argumentative stances of opposition using Tree Kernels, in: *ACAI 2019*, New York, ACM: 17-22.
- [8] Liga D., Palmirani M. (2020). Combining Tree Kernels and Tree Representations to Classify Argumentative Stances. *ASLD@ISWC 2020*: 12-23.
- [9] Mandal A., Ghosh K., Ghosh S. et al. (2021). Unsupervised approaches for measuring textual similarity between legal court case reports. *Artif Intell Law* 29, 417–451 (2021).
- [10] Palmirani M. (2018). Akoma Ntoso for Making FAO Resolutions Accessible. *Law via the Internet 2018*: 159-169.
- [11] Robaldo L., Villata S., Wyner A. et al. (2019) Introduction for artificial intelligence and law: special issue “natural language processing for legal texts”. *Artif Intell Law* 27, 113–115.
- [12] Rossi A., Ducato R., Haapio H., Passera S., Palmirani M. (2019). Legal Design Patterns: Towards a New Language for Legal Information Design, in: *Internet of Things. Proceedings of the 22nd International Legal Informatics Symposium IRIS 2019*, Bern, Editions Weblaw, 2019, pp. 517 – 526.
- [13] Song K, Tan X, Qin T, Lu J, Liu TY. Mpnet (2020). Masked and permuted pre-training for language understanding. arXiv preprint arXiv:2004.09297. 2020 Apr 20.
- [14] Sovrano F., Palmirani M., and Vitali F. (2020) Deep learning based multi-label text classification of UNGA resolutions. In *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance (ICE-GOV2020)*.
- [15] Sovrano F., Palmirani M., Vitali F. (2020). Legal Knowledge Extraction for Knowledge Graph Based Question-Answering. *JURIX 2020*: 143-153.
- [16] Tagarelli A., Simeri A. (2021). Unsupervised law article mining based on deep pre-trained language representation models with application to the Italian civil code. *Artif Intell Law*.
- [17] Waltl B., Bonczek G., Scepankova E. et al. (2019). Semantic types of legal norms in German laws: classification and analysis using local linear explanations. *Artif Intell Law* 27, 43–71.
- [18] Huang Z., Low C., Teng M., Zhang H., Ho D. E., Krass M. S., and Grabmair M. (2021). Context-aware legal citation recommendation using deep learning. In *ICAIL21*. ACM, New York, 79–88.
- [19] Reimers N., and Gurevych I. (2020). Making monolingual sentence embeddings multilingual using knowledge distillation. arXiv preprint arXiv:2004.09813.
- [20] Li L., et al. (2018). A system for massively parallel hyperparameter tuning. arXiv preprint arXiv:1810.05934.