Final Report on Proof of Concept: Context Aware Legal Verification (in LEOS)

This study was carried out for European Commission, Directorate-General for Digital Services by:
Monica Palmirani[1*], Fabio Vitali[2]

[1]*University of Bologna, ALMA-AI, via Galliera 3, Bologna, 40121, Italy*
[2] *University of Bologna, DISI, via Mura Anteo Zamboni 7, Bologna, 40126, Italy*

This project has been developed thanks a team of UNIBO composed by: Michele Corazza, Generoso Longo, Emanuele Di Sante, Aurora Brega, Andrea D'Arpa.
The work has been reported on in three thesis: Aurora Brega (on inter alia testing the ACM), Generoso Longo, and Emanuele De Santis (developed partially the advanced functionalities of this project) are also important results for disseminating the culture of LEOS. Colleagues in the Commission participated in the assessment.

Disclaimer

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the Commission.
The Commission does not guarantee the accuracy of the data included in this study. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use which may be made of the information contained therein.

1. Introduction

The legal drafting activity is a crucial task in the legislative procedure in any deliberative assembly. The goals of this task are many: i) to support the political decision-makers; ii) to standardize the language with the legal tradition, adopting multilingual translations when necessary; iii) to apply drafting rules to improve quality, and clearness; iv) to guarantee the Rule of Law and the theory of law principles; v) to track the modifications happening over time due to the the legislative process. In the last 15 years many specialized editors have been developed [13],[5],[3],[1], in order to support these important goals using Natural language processing technology [6]. Among the proposed solutions some use the Semantic Web approach [2], while others apply Symbolic AI based on rules [12]. LEOS [5], [10] is one of the most promising web editors for legal drafting, it has been developed by the EU Commission to support the internal legal drafting activities but also with the aim to serve the Member States as well.

LEOS is an open-source web editor specific for legal drafting, it is written in Angular and it is oriented to manage all the law-making process [15].

The aim of this work is to develop a framework architecture that is capable of enhancing LEOS with add-ons, developed with AI technologies, that improve the quality of the legal content, help the legal drafters, and manage the law-making process. The two add-ons provide the following features [7],[4][4],[14]:

i. Suggest the pertinent normative definitions using similarity with the bill topic;
ii. Suggest the pertinent normative reference using the thematic similarity with the bill;
iii. Take into consideration the temporal information and the nested normative references;
iv. Use the metadata of ELI[1] and EUROVOC[2] to improve the similarity.

The aim is also to create a user interface capable of:

i. Reduce manual/error-prone work typing the normative references, also avoiding repetitions in legislative citations;
ii. Maximising reuse of similar legal concepts (e.g., definition);
iii. Increasing transparency and searchability of the existing legal knowledge included in the corpora.

[1] ELI: https://eur-lex.europa.eu/eli-register/about.html
[2] EUROVOC: https://eur-lex.europa.eu/browse/eurovoc.html?locale=it

2. Methodology

The adopted methodology is based on hybrid AI [11], and it uses multiple techniques for achieving its goals. We do not generate new text (e.g., using LLM o generative AI), but we intend to suggest pertinent, contextual, and significant existing legal knowledge extracted by the legal corpora, using a similarity index according to the bill parameters that the legal drafter is writing. We also use the EUROVOC classification and other contextual information provided by the experts during the drafting process (e.g., type of provision).

Secondly, the approach takes into consideration the temporal validity of the normative provisions, excluding those that are repealed, or suggesting the appropriate versions of the consolidated text according to the view date typed by the end-user. If the author seeks the normative definition of "privacy" before the GDPR, they can set the date of view before the 5 May of 2016 (the date of entering into force of the act) and the system will respect this setting.

Thirdly, we resolve the normative references in order to include in the model of indexing the text cited in the recursive way as well (only the first level), allowing us to grasp more information, especially when the definition is limited in the text and it consists only of normative citations to another provision (e.g., "For the purposes of this Directive, the definitions laid down in Article 2 of Directive 2000/60/EC shall apply").

Fourthly, the context is important for providing the relevant output of the suggestion. A definition depends on the topic of the bill. For example, we have many definitions of 'accuracy' and it depends on the topic of the document.

Fifthly, the user interface is a fundamental pillar for guaranteeing good usability, transparency, and explicability of the AI behaviors and output [8].

Finally, we use Akoma Ntoso [9] serialization for fostering the structure of the legal documents, the normative references, the metadata of the lifecycle of the document, the date of entry into force, into operation, and the date of repeal.

3. Dataset

The dataset used is composed by 10 years of European legislation (2010-2021), about 15.000 regulations and directives. It was provided by the European Publication Office in Formex 3.0 XML format. We have converted all the documents in Akoma Ntoso, and using a natural language processing approach we have annotated the definitions and the normative references.

The dataset includes about 899 documents with definitions. For definitions, we have considered only the explicit provisions usually titled "Definitions" or where a regular pattern can surely identify the relationship between a term (*definiens*) and description (*definiendum*) (e.g., 'definiens' means definiendum, "'domain' means one or several data sets that cover specific topics;"). The definitions that include

normative references are managed by navigating the link to include the complete information (e.g. 'personal data' means personal data as defined in point (1) of Article 4 of Regulation (EU) 2016/679).

4. Use Cases

4.1. Normative References

Normative references are qualified citations used for mentioning other documents or provisions relevant for the normative discourse. The errors during the typing of the normative references produce incorrect links and additional effort in the control phases.

The system permits to type incomplete normative references and to retrieve and rank the existing and into force references which are similar to the information requested by the end-user. In the case a citation of the form "Regulation 406", for example, the system returns all the Regulation which are valid, into force, numbered 406 and pertinent to the EUROVOC of the bill. The system completes the reference (e.g., Regulation 406/2010) and returns the title of the document and other information for identifying the act as well.

Due to the evolution of the European institutions, the references have changed syntax and patterns over time. For this reason, the end-user can easily make a mistake in the citation format. Our tool helps the end-user to compose the reference according to the historical period of the document cited. For example, a Regulation before 1968 is cited using number/yy/EEC (e.g., Regulation No 1009/67/EEC); after 1968 we have number/yy (e.g. Regulation (EEC) No 2195/91) and after 2009 we have yyyy/number (e.g., Regulation (EU) 2016/679).

4.2. Legal Definitions

Legal definitions are a sensitive part of the law because they define new legal concepts, new terminologies, equivalences between different other definitions, and exceptions in the case of specific cases. In EU legislation, we usually have a clear article called "Definitions", but sometimes we could also find technical definitions in the last part of the act or in the annexes.

Additionally, we could have definitions organized in a long list of points, which might be connected to each other. Definitions are composed of three main parts: definiens (term); definiendum (description); legal concept (abstract class of concept). The use of the same term for multiple definitions is not infrequent, and the term might have completely different meaning in different domains (e.g., pollution has different definitions according to the domain like water, energy, industry, etc.).

For this reason, the tool calculates the similarity of a given term (which can also be composed of multiple words) with the existing, valid, and updated (present in

consolidated versions of documents) definitions in the legal corpus, using the similarity index as a criterion.

4.3. Legal Reporting Requirements Clauses

In the context of the EU Legislation is fundamental to classify and extract the so called "Reporting Requirements Clauses". The reporting requesting (RR) legislative provisions are particular obligations to perform reports addressed to an institution with the goal of monitoring specific legislative domains over time. The SORTIS (Study on Regulatory Reporting Standards)[3] project promoted by the European Commission Digital-Ready Policymaker unit aims to develop a solution for detecting a particular type of legal obligation called "reporting requests" inside of the European legislation. As a result, regulatory reporting metadata vocabulary (RRMV)[4] has been developed. This ontology can be used to structure 'requests' in legal provisions. The EU legislation obliges many different stakeholders to regularly report data such as financial transactions, capital liquidity, data breaches, etc. to present reports and make monitoring activities in order to improve the legislation. The amount of required data is constantly increasing and there is often a lack of knowledge on what can be reused. To avoid overlapping or missing reporting, these obligations are monitored over time according to European policies. Moreover, they also need to be updated as the legislation changes over time. For this reason, we use ontology and AI for detecting ex-post the RR. However, it is important also to prevent convoluted norms and to have a template for drafting these provisions in a standard way, in simple text and machine-readable. This permits to:

- simplifying identification of existing requirements;
- tracking them over time;
- making complex queries jointly with other legislative metadata (e.g., temporal metadata);
- measuring the policy effect.

An example is produced here.

[3] Streamlining regulatory reporting: the SORTIS project results | Joinup (europa.eu).
[4] https://code.europa.eu/regulatory-reporting/rrmv.git

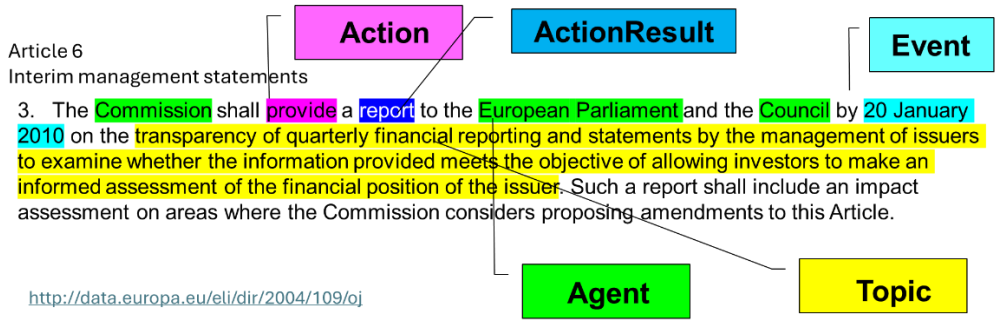

Figure 1 – Example of RR.

We have integrated inside of LEOS the actions for:

1. qualify during the drafting of the provision as "RR" with a special button;

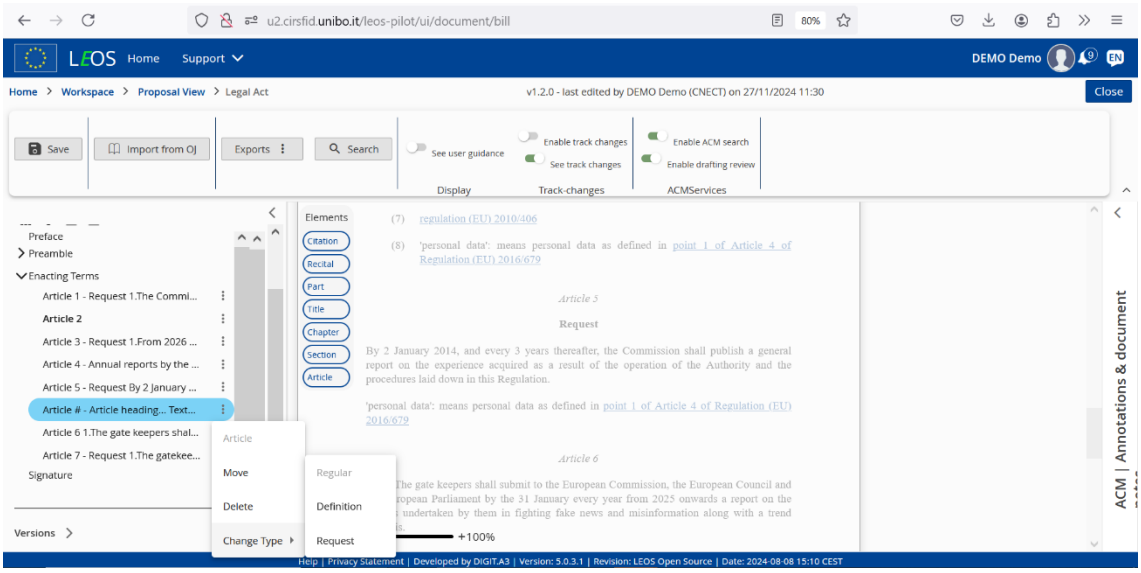

Figure 2 – Creation of a "Request" Article.

2. to save the request and to call the AI algorithm for detecting the components;

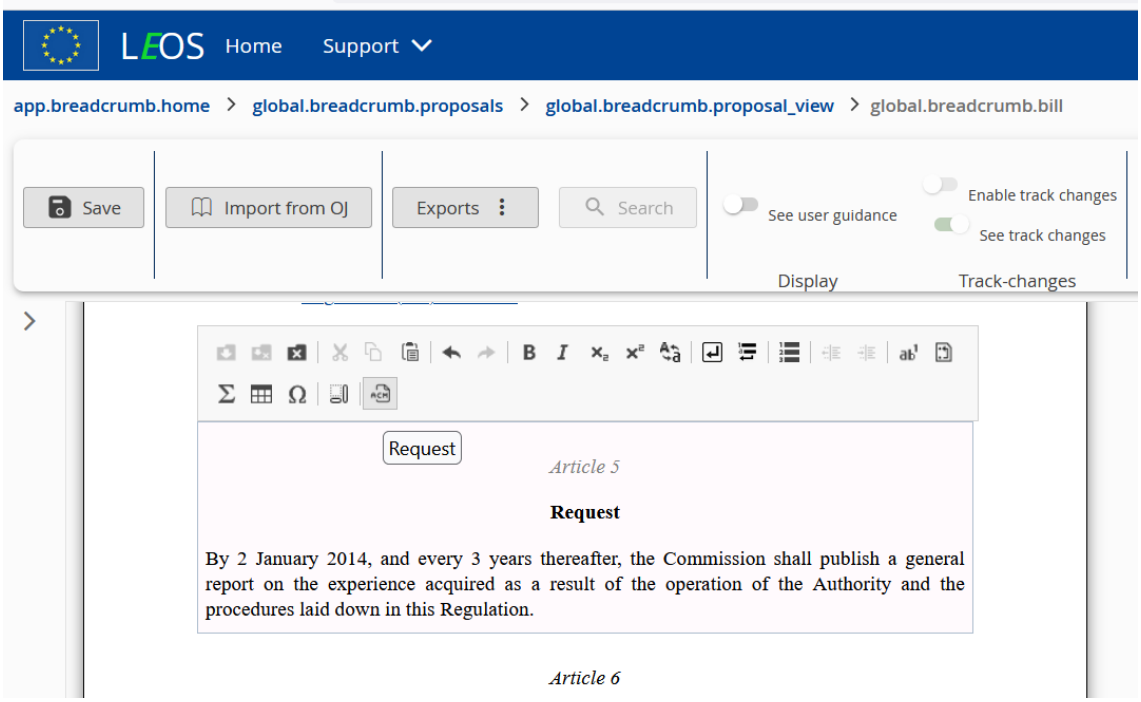

Figure 3 – Generation of the parsing of the text by clicking on the button "ACM".

3. to extract the parameters and model them in RDF using the RRVM ontology;
4. to return the information to the text in bold for permitting to the end-user to check the correctness of the detection;

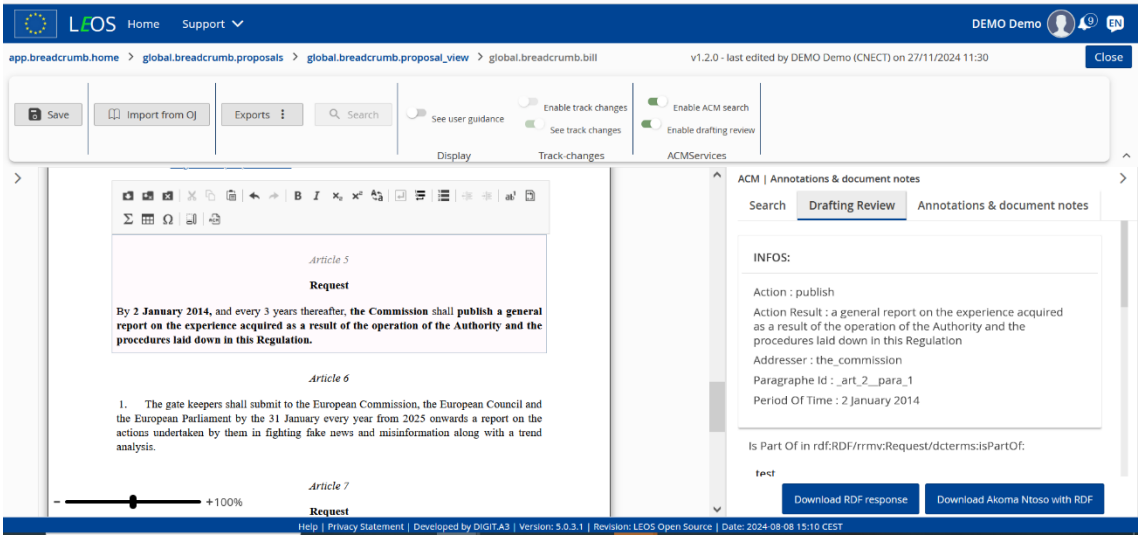

Figure 4 – Extraction of the parameters (right window) and returning of the information in the text in bold.

5. to download the RDF or the AKN+RDF included in the <meta> block (precisely the <otherAnalysis> block).

Figure 5 – AKN+RDF in the metadata block.

5. Architecture

The overall architecture is composed of an XML database that includes the Akoma Ntoso XML documents and an SQL database containing the correspondence between each document and its EUROVOC categorization. Each EUROVOC is associated with an average of the Word2Vec [16] embeddings of the words composing it. The eXist database including all the AKN-XML documents5 can also use Lucene Java library to calculate the index of the document text and in particular to the definitions (defBody elements). When a new document enters the eXist database it is also indexed in the SQLDB and the Word2Vec representation of its definitions is stored. If the document does not have EUROVOC tags, we extract them from CELLAR and we serialize the information in the metadata of the Akoma Ntoso documents.

[55] eXist is an XML database that is indexed using Lucine and querable with XQuery.

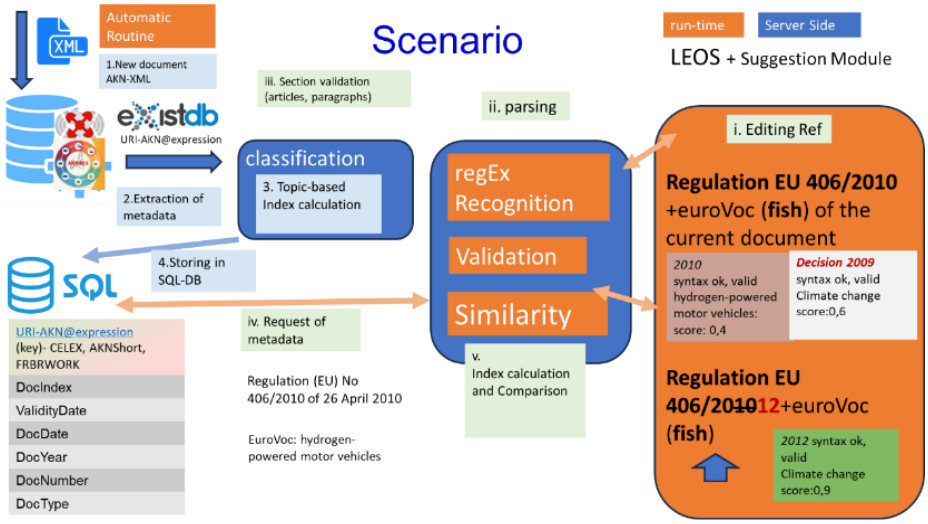

Figure 6 – Architecture of the system.

During legal drafting, if the end-user wants to get a suggestion (e.g., normative reference or definition), they need to provide some parameters as inputs, in order to calculate the corresponding indexes like the title and the EUROVOC keywords of the bill (proposal of law). The dynamic input typed by the end user (e.g., incomplete normative reference or definition keywords) is parsed to compare the content with the existing document collection in eXist. After a first filter using traditional Information Retrieval techniques for grasping the relevant documents, the similarity score is calculated based on the text retrieved and compared with the embeddings of the input parameters stored in the SQL DB (for EUROVOC values) and using the similarity algorithm of Lucine for the definitions. The ranking is based on the index score, the temporal parameters, considering the normative citations included in the normative provision retrieved as well.

Lucene Similarity class implements the scoring model. The library offers several already-built implementations of the Similarity class, which reflect different scoring models developed in the field of Information Retrieval. Our implementation adopts Default Similarity class, which combines the Boolean model, adopted to filter documents matching the query, and a readjustment of the Vector Space model, based on TF-IDF weights, for scoring results. In particular, VSM is refined by Lucene taking into account the corpus statistics contained in the inverted index, the number of terms that correspond to the query, and the multiplying enhancement factors expressed in the research. This class is also exploited by the process chain of indexing, since it deals with the calculation of the normalization factors, which depend on the length of the fields and the boost factors specified in the configuration(Similarity (Lucene 3.6.1 API) (apache.org)).

Additional module was developed for the “Reporting Requirements” clauses.

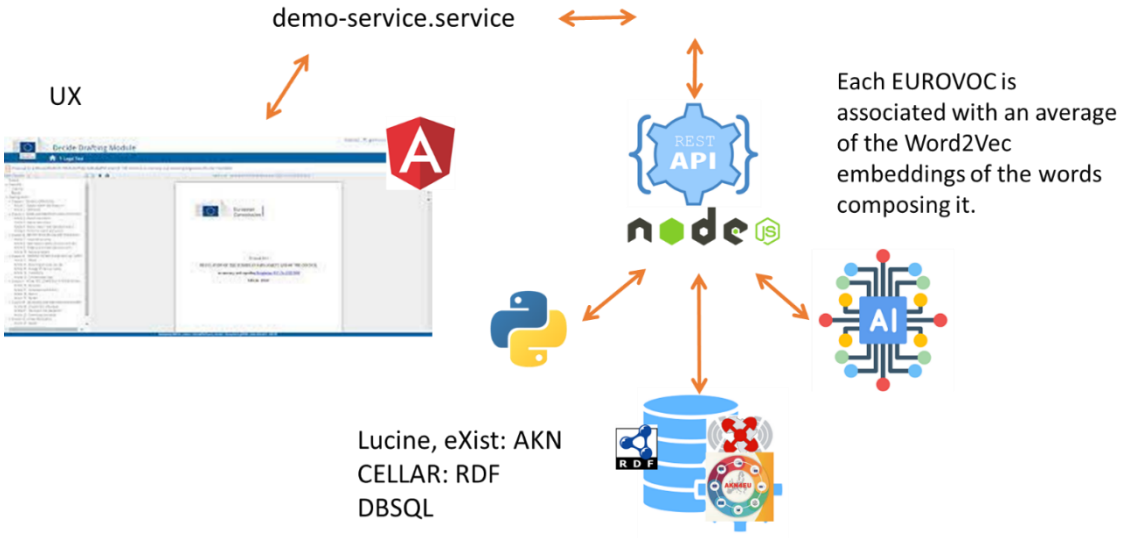

Figure 7 – Architecture of the system.

6. User-interface

The user interface is a fundamental part of this application. LEOS is enriched with an add-on that enables these functionalities in a selective way. The suggestions are offered in a portion of the window that allows the end-user to confirm or discard the output, or to integrate the results in the drafting text.

Our custom components are organised in a dedicated application folder, comprising new components (stored in .component.ts, .component.html, and .component.scss files), new classes (.ts files), and service (in a .service.ts file). This service manages the essential methods and global variables used by our approach.

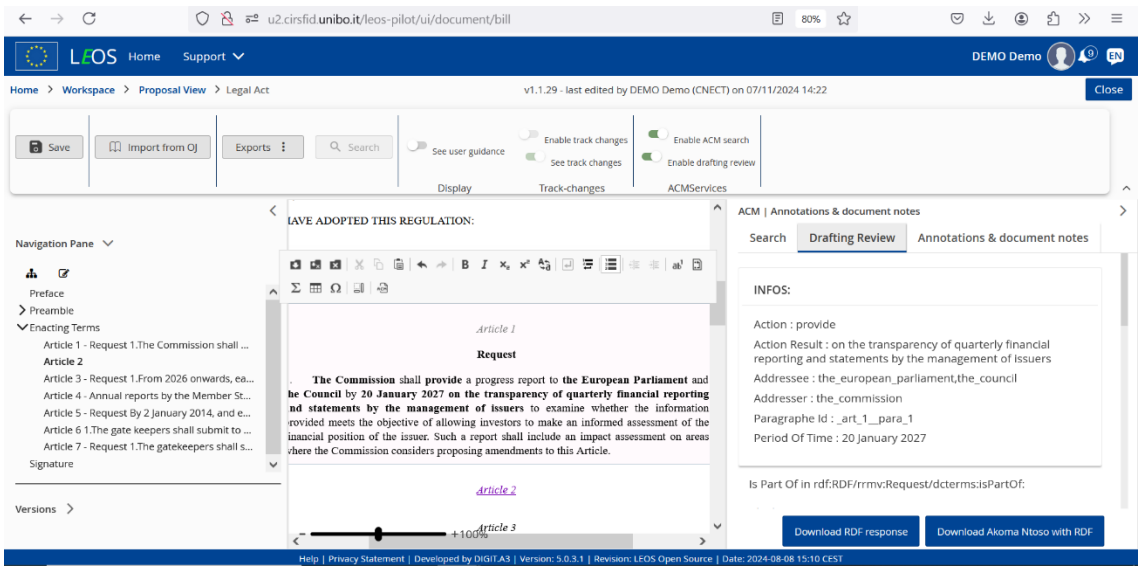

Figure 8 – Interface of LEOS with the add-on.

To maintain consistency, we adopted a style for our extension that closely imitates the original application's design. Many of the components used were taken from the eUI library, and we followed the guidelines and suggestions provided by the eUI framework. The version of the eUI library used is 14, the same one adopted by LEOS and used in its native components. Therefore, both the shape and color of the interface elements are consistent with those indicated by the framework.

The components we added, we always provide feedback to the user, displaying results when generated, an error message if the service responses raise an issue, and an alert if the user's request is not executed correctly, accordingly with the functionality we aim to provide. We designed it so that the user knows the reasons for an incomplete or incorrect request and is given the opportunity to make any necessary corrections. We also strive to maintain consistency in the terms used in the labels, ensuring that each element is identified by a unique name and avoiding multiple elements with the same name.

The end user of the service is an expert in legislative matters, so we prioritised making the interface simple and intuitive but also very specific for professional tasks in drafting, considering that the user has clear knowledge of the subject matter being addressed. We created mockups of the interface to evaluate it before implementation, ensuring that it is indeed usable and effective. The end-user is constantly involved in the evaluation with regular meetings where the usability is tested and feedback is incorporated in the software.

We use also drag&drop for moving the suggestion from the right window to the editor text.

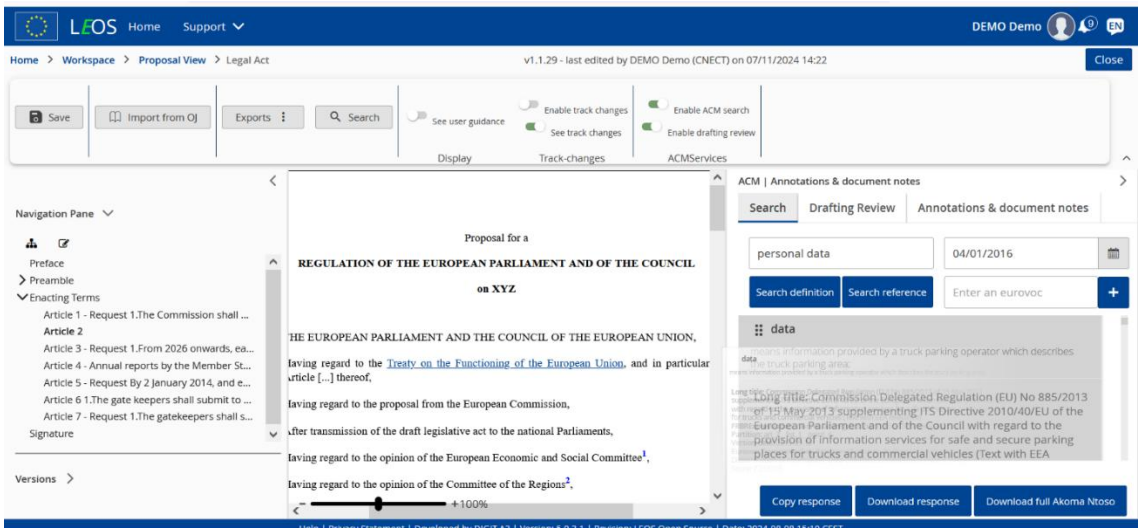

Figure 3 – Interface of LEOS with the add-on results.

7. Conclusions

The current PoC demonstrated the integration of add-ons integrated into LEOS web editor to enhance legal drafting tasks using AI applications. The user interface is a fundamental component of this PoC that is designed to incorporate the principles of transparency, accessibility, user experience, and explicability. The methodology is to not generate new text (e.g., like LLMs) to avoid hallucinations, which could affect the democratic rules of the law-making process. In the future we could consider to use LLM mitigated by RAG considering that now we have embedding vector DB, AKN4EU XML repository.

We aim to extract and offer to the legal drafters the legal knowledge stored in the corpus, which is sometimes difficult to find due to the large volume of documents, and to return the relevant information accompanied with a particular index score based on temporal parameters, similarity of text using qualified legal provisions like definitions and normative references. The first results were evaluated by legal experts and they are promising and pertinent to the drafting text. Moreover, the end-users appreciated the provided suggestions, which could retrieve pertinent information using topic similarity, cutting repetitive work and focusing on higher-level tasks.

8. Recommendations

Based on the study results the following recommendations have been arrive at:

- To continue the work on the current PoC's and ready the technology for large scale take up with further emphasis on increasing user interface and special attention to synergise the 'Human AI interaction'
- To test within the Commission the results of the current PoC more widely
- To use the opportunity of the finalisation of the study to solicit input from practioners in Commission on to use of AI in drafting legislation.
- To expand the set of PoC's inter alia considering the results of the study on 'AI based solutions legislative drafting in the EU' see AI-based solutions for legislative drafting in the EU - Publications Office of the EU
- To explore the use of Large Language Models within the context set out in previous study including on 'Drafting Legislation in the ear of AI and digitisation', the SORTIS project (see Streamlining regulatory reporting: the SORTIS project results | Interoperable Europe Portal) and the context of Digital Ready Policy Making (see Digital-ready Policymaking | Interoperable Europe Portal)

Moreover, the following recommendations made in the study on 'Drafting legislation in the era of AI and digitisation' are reiterated:

- Secure a continued high-level management support for the direction of travel, embracing innovation while proceeding with caution. Embracing innovation, beyond the status quo implies accepting an increased level of risk, a willingness to explore and engage with new ideas and technologies and to take on a leadership role, where necessary changing our culture and ways of working.
- Advocate a pondered and thoughtful use of AI in law-making, seeing that the potential contribution this may make to the ongoing digital transformation could be very significant, translating to improved quality and increased efficiency.
- Now is the right time to act, e.g., by doing more experimentation and piloting 'closer to business' and 'at scale'.
- Adopt a multi-skill, multi-domain team devoted to implement an ambitious agenda on the use of AI for drafting legislation.
- Use the opportunity of the results of this study to illustrate the high potential of AI in drafting law within the Commission, with EU institutions, Member States and the wider GovTech community.
- Continue to liaise with and work to obtain buy-in from lawyers and policymakers in the Commission.

Code and technical material

The code is here: https://gitlab.com/CIRSFID/leosplus
The project produced also a thesis of Aurora Brega.
The project produced also a paper in open source: https://ceur-ws.org/Vol-3762/590.pdf

References

[1] Agnoloni Tommaso, E Francesconi, P Spinosa, xmLegesEditor: an opensource visual XML editor for supporting legal national standards Proceedings of the V legislative XML workshop, 239-251, 2007
[2] Casanovas, Pompeu; Palmirani, Monica; Peroni, Silvio; Van Engers, Tom; Vitali, Fabio, Semantic Web for the Legal Domain: The'next step, «SEMANTIC WEB», 2016, 7, pp. 213 - 227
[3] Grant Vergottini, https://xcential.com/legispro/standards/
[4] Griglio Elena - Marchetti Carlo, La "specialità" delle sfide tecnologiche applicate al drafting parlamentare : dal quadro comparato all'esperienza del Senato italiano / Elena Griglio, Carlo Marchetti, Osservatorio sulle fonti. - 2022, n. 3, p. 361-386
[5] LEOS https://joinup.ec.europa.eu/collection/justice-law-and-security/solution/leos-open-source-software-editing-legislation
[6] Lesmo Leonardo, Alessandro Mazzei, Monica Palmirani, Daniele Paolo Radicioni: TULSI: an NLP system for extracting legal modificatory provisions. Artif. Intell. Law 21(2): 139-172 (2013)
[7] Lorello Laura, La qualità della legislazione diventa un'esigenza bicamerale : considerazioni sul nuovo Comitato per la legislazione del Senato della Repubblica nel primo semestre di attività, Federalismi.it : rivista di diritto pubblico italiano, comunitario e comparato. - 2023, n. 27, p. 17-45
[8] Palmirani M., Vitali F, Legislative Drafting Systems, in: Usability in Government Systems, NEW YORK, Morgan Kaufmann, 2012, pp. 133 - 151

[9] Palmirani Monica (2011). Legislative change management with Akoma Ntoso. In Legislative XML for the semantic Web, 101–130. Springer
[10] Palmirani Monica et. al. Drafting legislation in the era of AI and digitisation, https://joinup.ec.europa.eu/sites/default/files/document/2022-06/Drafting%20legislation%20in%20the%20era%20of%20AI%20and%20digitisation%20%E2%80%93%20study.pdf
[11] Palmirani Monica, F Sovrano, D Liga, S Sapienza, F Vitali, Hybrid AI Framework for Legal Analysis of the EU Legislation Corrigenda, Legal Knowledge and Information Systems, 68-75
[12] Palmirani Monica, Luca Cervone, Octavian Bujor, Marco Chiappetta: RAWE: An Editor for Rule Markup of Legal Texts. RuleML (2) 2013
[13] Palmirani Monica, Raffaella Brighi: An XML Editor for Legal Information Management. EGOV 2003: 421-429
[14] Tafani Laura, Federico Ponte, Le tecniche legislative statali, regionali e dell'Unione europea a confronto : per un auspicabile ravvicinamento, Osservatorio sulle fonti. - 2022, n. 1, p. 447-497
[15] Leos Manual https://joinup.ec.europa.eu/collection/justice-law-and-security/solution/leos-open-source-software-editing-legislation/releases
[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.
[17] Palmirani Monica, Fabio Vitali, Generoso Longo, Emanuele Di Sante, Aurora Brega, Andrea D'Arpa, Michele Corazza, Legal Drafting supported by AI: enhancing LEOS, https://ceur-ws.org/Vol-3762/590.pdf