Information Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Lynx: A knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain

Julián Moreno Schneider [a,*], Georg Rehm [a], Elena Montiel-Ponsoda [b], Víctor Rodríguez-Doncel [b], Patricia Martín-Chozas [b], María Navas-Loro [b], Martin Kaltenböck [c], Artem Revenko [c], Sotirios Karampatakis [c], Christian Sageder [d], Jorge Gracia [e], Filippo Maganza [f], Ilan Kernerman [g], Dorielle Lonke [g], Andis Lagzdins [h], Julia Bosque Gil [e], Pieter Verhoeven [j], Elsa Gomez Diaz [i], Pascual Boil Ballesteros [i]

[a] DFKI GmbH, Germany
[b] Universidad Politécnica de Madrid, Spain
[c] Semantic Web Company GmbH, Austria
[d] Cybly GmbH, Austria
[e] Universidad de Zaragoza, Spain
[f] Alpenite, Italy
[g] KDictionaries, Israel
[h] Tilde, Latvia
[i] Cuatrecasas, Spain
[j] DNV GL, The Netherlands

ARTICLE INFO

Article history:
Received 31 October 2020
Received in revised form 1 November 2021
Accepted 2 November 2021
Available online xxxx
Recommended by Andrea Tagarelli

MSC:
00-01
99-00

Keywords:
Text analytics
Tools
Systems
Applications
Knowledge discovery/representation

ABSTRACT

The EU-funded project Lynx focuses on the creation of a knowledge graph for the legal domain (Legal Knowledge Graph, LKG) and its use for the semantic processing, analysis and enrichment of documents from the legal domain. This article describes the use cases covered in the project, the entire developed platform and the semantic analysis services that operate on the documents.

1. Introduction

Systems for the processing of content from the legal domain have, in recent years, received a lot of attention as breakthroughs in Artificial Intelligence (AI) and Data Science (DS). They have brought about a plethora of new opportunities and possibilities, among others, in the area of compliance checking.

Due to the high demand for transparency in public institutions and administrations and as a direct consequence of legislation passed in this regard, the situation has resulted in many public institutions publishing and sharing their data according to open standards and best practices, especially to enable easy data access and data reuse. In Europe, this has been driven by the *Directive on open data and the re-use of public sector information*, also known as the Open Data Directive,[1] which entered into force on 16 July 2019. European institutions and national governments are currently publishing law, policies and recommendations in their own official languages. Accessing, understanding, comparing and making use of these documents poses serious challenges to all stakeholders, including companies, that are already or that are about to become active internationally, i. e., on new national markets.

* Corresponding author.
E-mail address: julian.moreno_schneider@dfki.de (J. Moreno Schneider).

[1] Directive (EU) 2019/1024

https://doi.org/10.1016/j.is.2021.101966
0306-4379/

This contribution is the result of the EU-funded innovation project Lynx, in which a knowledge-based AI service platform for content processing, enrichment and analysis for the legal domain has been developed. The specific focus of the platform is to assist companies in researching and successfully addressing compliance issues in a multilingual and multi-jurisdictional scenario. The Lynx Service Platform (LynxSP) relies on a data model to structure and link documents and entities in a Legal Knowledge Graph (LKG), and on document and workflow managers that enables the flexible orchestration of a set of Natural Language Processing (NLP) and Information Retrieval (IR) services that process legal documents. The document manager plays a central role in the Lynx Service Platform, since it is in charge of structuring, annotating and storing documents that will then be consumed by the NLP and IR services. The order and combination of services is determined by the workflow manager to implement and realise tasks such as cross-lingual search or question answering.

This article is structured as follows. Section 2 presents the use cases defined and covered in the project. Section 3 outlines the general architecture and provides an overview of the Lynx Service Platform and its main components. Section 4 introduces the linguistic resources created for and included in the LKG. Section 5 describes the semantic processing services implemented in the Lynx project and provides more details on individual components. Section 6 contextualises our approach with related work for the processing of documents or information from the legal domain. Finally, Section 7 concludes the article and sketches directions for future work.

2. Use cases

The Lynx project consists of three use cases which demonstrate the usefulness of the LKG, the Lynx services and the service platform. Each of these use cases focuses on different functionalities of the Lynx Service Platform.

2.1. Geothermal Energy (GTE)

Governments play a crucial role in legislating and assuring compliance to mitigate safety and environmental risks, in all sectors and the energy industry in particular due to the transition it is currently in. With the expected growth in sustainable energy alternatives, continuous standardisation of technology to bring down costs and risks can be expected. Most countries will, either individually or together with others, develop policies and laws. Governments will seek balance in the use of subsidy schemes to accelerate growth and develop regulation or legislation to mitigate safety and environmental risks to guide the sustainable growth of technologies and markets. Companies active in these supply chains are likely to seek cross-border growth in order to develop economies of scale and bring costs down. If cross-border growth is envisioned, keeping up with the most recent legal and regulatory rules is likely to become a challenge as country-specific clauses and local languages complicate getting an overview.

In Lynx, this specific context and challenge is explored for the geothermal energy domain as a proxy of the wider renewable energy domain.

What is Geothermal Energy (GTE)? GTE is heat generated in the sub-surface of the Earth. A geothermal fluid or steam carries the geothermal energy to the Earth's surface. Geothermal energy operators drill a production and an injection well (also known as a doublet) to a certain depth (between 100 m and 4000 m) to circulate fluid to produce "heat".[2] Depending on the temperature, this fluid can be used to produce clean electricity, or as a baseload for municipal district or industry heating or cooling. GTE is seen as a promising sustainable energy alternative and the industry (supply) and its users (demand) is at the dawn of accelerated growth [1].

Geothermal energy challenges. To prove the value of the technical approaches, use cases were designed to explore solving the typical problems and challenges in this domain using the services developed in Lynx, for example:

1. National actors in the GTE supply chain facing regulatory risks, missing potential opportunities, are taking poor decisions due to compliance information being fragmented over multiple information sources. The *first GTE challenge* is "Can value be generated by connecting machine-readable regulatory information resources for GTE?"
2. International actors in the GTE supply chain struggle with a lack of understanding of country-specific regulatory frameworks (which is a competitive disadvantage) which limits international competition and the potential benefits of economies of scale as well as standardisation. The *second GTE challenge* is "Can internationalisation be stimulated by providing the same level of access to relevant compliance information for, and from, different EU countries?"

The Lynx demonstrator for the GTE use case. To address these two challenges, a web application – *Recommender* (see Fig. 1) – was developed on top of the Lynx service platform. It facilitates searching for relevant documents in multilingual corpora. The tool accepts plain text and PDF documents. Documents are preprocessed and plain text is extracted. The plain text is then annotated by the Entity Linking (EL) service (see Section 5).

The annotated documents are processed by the Semantic Similarity (SeSim) service (Section 5), see Fig. 2. On the left the original document is displayed with highlighted entities from the Legal Knowledge Graph (Section 3.1), identified through the entity linking service. The SeSim service returns not only similarity scores, but also the reasoning behind these scores, visualised as a table behind each document's title. The documents are translated using the Neural Machine Translation (NMT) service and presented in the user's language.

2.2. Contract analysis

Contracting is a common activity in companies, but managing contracts systematically, which includes keeping track of changes or updates, is a cumbersome activity only few companies are effective at. Most companies do not have a database with all the information contained in their contracts, which prevents them from easily finding or monitoring information or applying changes. Let us assume the following situations in the context of a company:

1. A specific contract is needed urgently but no one knows where to find the most recent version, because the responsible employee left the company. Moreover, the other party confronts you with a signed amendment you have never seen before.
2. There is a change in law, and you need to know which of the existing contracts are effected.

[2] https://kennisbank.ebn.nl/en/master-plan-geothermal-energy-in-the-netherlands-2018/. All URLs mentioned in this paper were last visited on 29 October 2021.

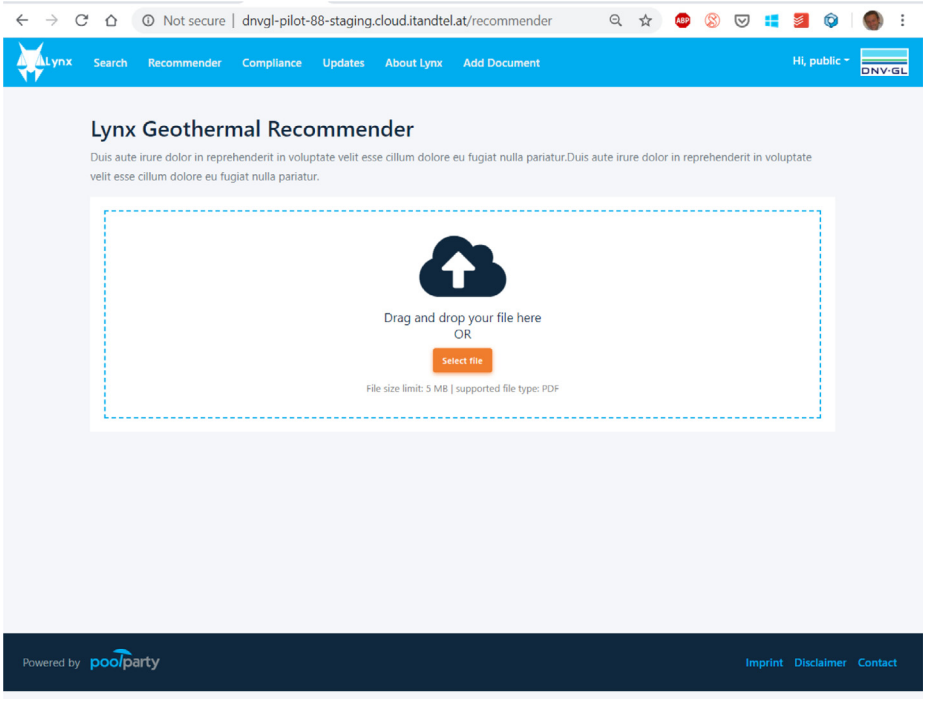

Fig. 1. Geothermal Use Case: Recommender landing page.

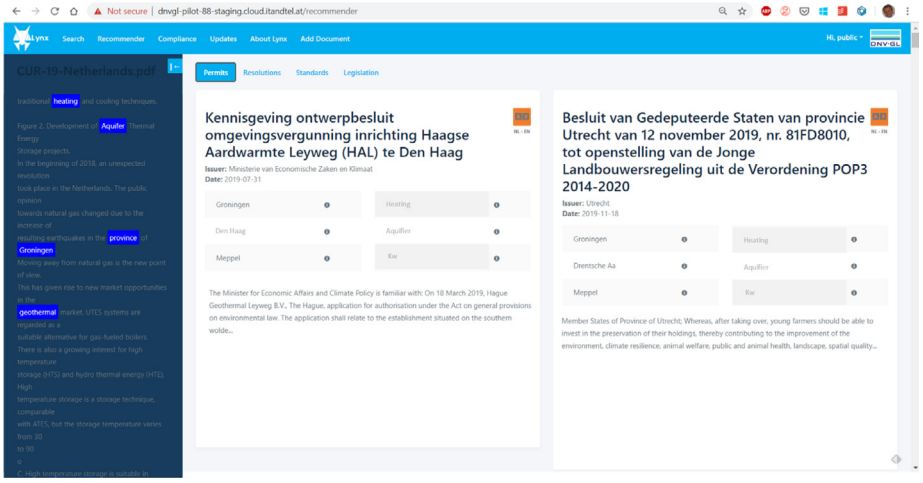

Fig. 2. Geothermal Use Case: Recommender screen.

3. An overview of all obligations with regard to a certain company is needed.

Countless organisations are confronted with such scenarios. The problem can be generalised as follows: due to a lack of central administration, contracts are physically and electronically distributed across the entire organisation. As a result, often no one has an overview, which leads to inconsistent decisions, breaches of contracts and (financial) disadvantages.

One solution appears to be the implementation of a comprehensive cross-organisational contract management process. Flitsch [2] defines contract management as the creation of ideal structures for contract planning, contract design, contract negotiations, implementation of contracts, contract administration and contract archiving.

In many cases, organisations are lacking these structures. Against this backdrop, we are focusing on automated contract analysis. Building on this, we provide smart contract archiving solutions and compliance services. We expect our application to result in enhanced contract compliance, which will ultimately lead to reduced risks and costs for organisations.

These activities are based on the assumption that developing a legal knowledge graph – duly interlinked and integrated – would result in much more direct access of the applicable law and, thus, in facilitating compliant and diligent actions. To this end, we are channelling our efforts to fulfil what Hamming [3] formulated so aptly decades ago: “The purpose of (scientific) computing is insight, not numbers”.

The most simple use case is the analysis of a single contract. However, reality is much more complex. Typically, a large number of highly diverse contracts needs to be analysed and kept track of, taking into account various regulatory frameworks. In order to achieve this, we are pursuing two approaches. On the one hand, we work on pure back-end solutions, and, on the other hand, we provide a visualisation of the created data space (see Fig. 3).

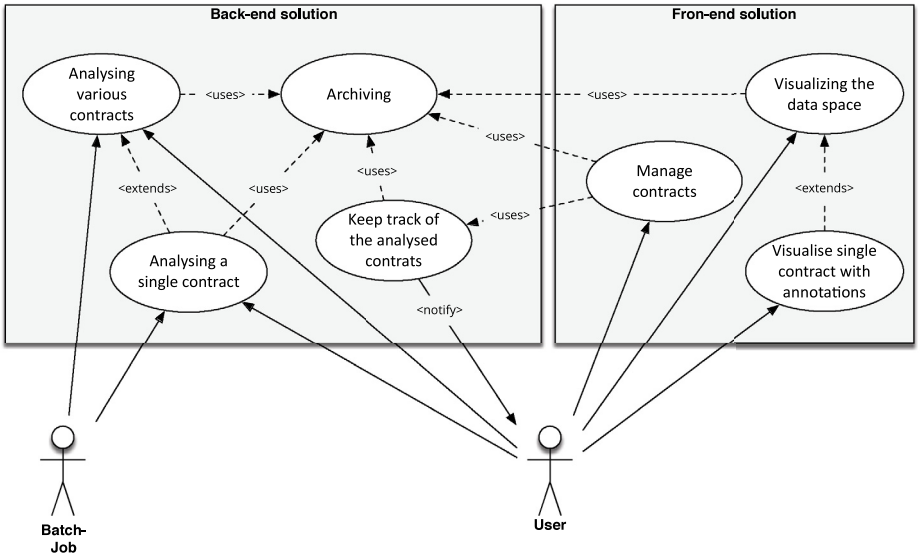

Fig. 3. Contract Analysis Use Case: Solution approaches.

The back-end solution provides the analysis of single or multiple contracts. Both functionalities are based on the archive where contracts and the extracted information is stored. These services can be used by other services and applications. Through the front-end a user has the possibility to manage contracts (add, delete, update, group, search, etc.). The user can view a single contract and annotations or get a broader view of the corresponding data space, e. g., legislation, similar contracts or other contracts with the same partner. In addition, the user is notified when legislation changes with effects on given contracts. These technologies support companies in achieving compliance.

2.3. Labour law

Companies are affected by different regional regulations, almost all of which are published in regional outlets, often only in the respective regional language. This problem is further accentuated at the European level. While there is a common regulation and regulatory framework, the extent to which European directives have been transposed can differ greatly. The Lynx consortium partner Cuatrecasas is a full service law firm, which, although leading in Spain and the Iberian market, also provides legal advice to international companies, which requires the company to deal with many additional languages as well as national laws and regulations. This use case focuses on labour law, which typically involves several international activities because of the clients' geographical expansion (e. g., mergers & acquisitions and due diligence). In large corporations, geographical expansion and differing workers' rights are a common problem, as the regulations of all countries involved differ.

The use case can be extended to other legal practices like tax, intellectual property rights or data privacy and personal data (regulated in the General Data Protection Regulation (GDPR) directive at European level but with global impact). The problem regarding cross-border regulations is more frequent in a more globalised economy, where the level of regulation, the number of laws and the frequent changes they undergo are also increasing yearly.

We aim to cover two use cases that have no significant functional difference. The first, *internal* use case is targeted at lawyers to enable more efficient access to legislation across jurisdictions, while the second, *external* use case, is intended for clients to provide their internal legal department teams or even their human resources department with direct and secure access to legislation. For the two use cases, we envision the same technical solution (Fig. 4). Our solution resembles a legal chatbot with a user interface especially designed for non-legal experts (non-lawyers, junior lawyers or paralegals).

The solution can be thought of as a smart search tool for lawyers, where results are texts or excerpts directly extracted from the law (technical legal language). A chatbot interface would offer the ideal interaction scenario for non-legal experts, relying on a question answering system that would simplify access to regulatory sources and help them interpret the legal content. Combining the chatbot interface with semantic search will be one of our main challenges.

3. Lynx Service Platform (LynxSP)

The targeted functionalities and challenges posed by the three pilots require a portfolio of NLP and IR services that are able to work both independently and in pipelines. We developed the Lynx Service Platform (LynxSP) as a service oriented platform to address this requirement. Early in the development process we identified the following three high-level requirements for the services:

1. All services in the platform share common rules for the development of their APIs using OpenAPI specifications.[3] The rules include:
   - common codes for (error) messages,
   - conventions for the naming of parameters,
   - conventions for the routes of endpoints.

   Compliant services can be called from the workflow manager with relatively little development effort. The responses can be processed and the user can get information about the execution of the service.
2. Ideally, services are to be containerised and deployed through an orchestrated application platform — OpenShift.[4] This deployment strategy allows for scalability as additional instances can be deployed on demand. Moreover, services can be quickly deployed through a new

[3] https://swagger.io/specification/#version-3.0.3

[4] https://www.openshift.com

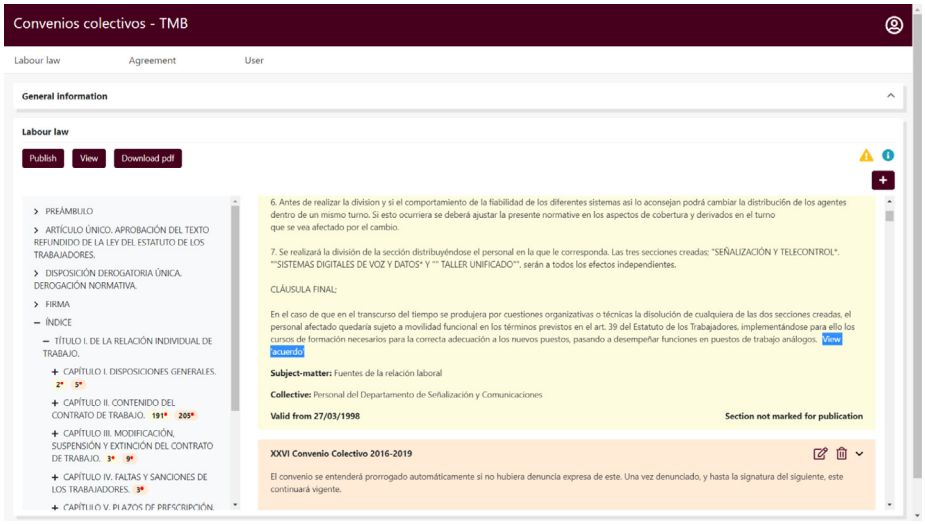

Fig. 4. Labour Law Use Case: Envisioned approach.

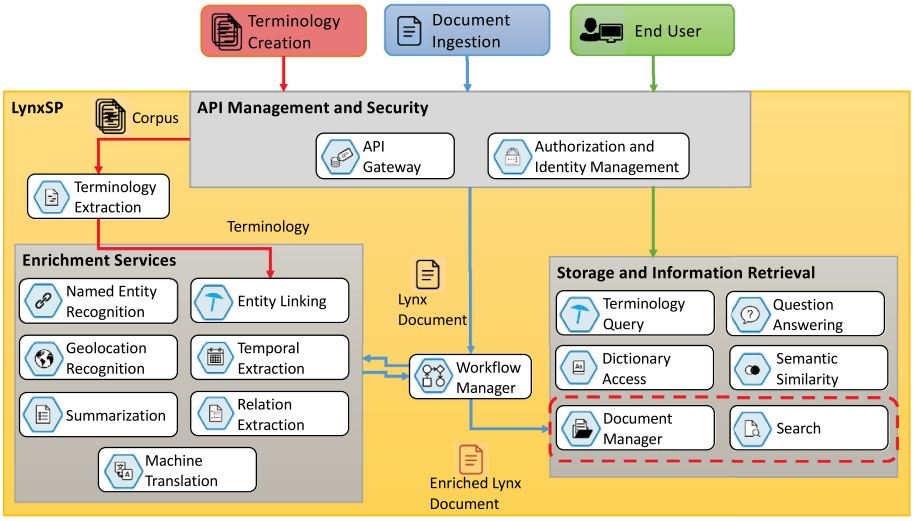

Fig. 5. Lynx architecture with population, terminology extraction and interaction workflows.

infrastructure – together or individually – for example, to enable local processing of sensitive data.

3. All services process the Lynx Document format (Section 3.1), which enables direct exchange of data between different services and easier integration into workflows.

The architecture and interactions between individual services and three different workflows are presented in Fig. 5. The individual components and principles of LynxSP are described in Sections 3–5 .

3.1. Data model for a Legal Knowledge Graph (LKG)

The Lynx services need to operate on documents using a uniform format. Modern legal information systems represent documents or at least their metadata in well structured form, such as in RDF. This trend has been further supported by the public policies of the European institutions and by legal gazetteer publishers who have enthusiastically adopted the corresponding W3C recommendations for publishing open law as open data. The Europe-wide effort European Legislation Identifier (ELI) has harmonised the way legislation is published. ELI defines three pillars. First, every piece of legislation is identified by an HTTP URI; second, the same metadata elements are used across the different jurisdictions; and third, metadata is shared in a machine-readable form, using elements from the ELI Ontology. Yet, not even legislation obtained from ELI-compliant sources is sufficiently coherent for the purposes of Lynx. The implementation level of ELI varies (only a core part of ELI is shared among various jurisdictions and indeed each EU country coined its own specialisation of the ELI ontology for metadata representation of documents), the details are heterogeneous and in any case, many sources of interest for Lynx are not in scope (e. g., contracts, international standards). Moreover, Lynx documents must accommodate subjective annotations that an official law publisher would never dare to do (e. g., recommendations). Therefore, a new data model, inspired by ELI but with additional features, in particular document annotations, is necessary.

Another relevant initiative is Akoma Ntoso[5] (Architecture for Knowledge-Oriented Management of African Normative Texts using Open Standards and Ontologies), one of the most advanced international standards for the representation of judicial, legislative and parliamentary documents of all kinds. This standard was

5 http://www.akomantoso.org

created by the initiative of the "Africa i-Parliament Action Plan",[6] which is a program of the UNDESA (United Nations – Department of Economic and Social Affairs). The project aims to achieve transparency, open access, exchange, and ultimately the maximum democratisation of legal information produced by the relevant experts in parliaments, courts and government institutions. In order to further develop Akoma Ntoso, OASIS[7] (Organization for the Advancement of Structured Information Standards) formed, in 2012, a new technical committee, the OASIS LegalDocumentML Technical Committee. This committee was established to create specifications for a standard of legal documents of parliamentary, legislative and judicial origin. Akoma Ntoso is the foundation of the specifications of the new OASIS standard.

Akoma Ntoso is a highly complex and also extensible standard that can be adapted to many different use cases in the above mentioned parliamentary, legislative and judicial domains. The standard is fully focused on the assumption that human experts create and maintain Akoma Ntoso documents or that existing legacy documents are transformed into Akoma Ntoso using corresponding scripts or stylesheets. Our focus in Lynx, on the other hand, is the automated processing of documents from the legal domain with the goal of extracting various types of semantic information and knowledge in order to enrich the processed documents with this newly discovered semantic information. As mentioned, this use case is not immediately enabled or supported by Akoma Ntoso but there are a number of more 'lightweight' best practice approaches in use in Natural Language Processing that have been easier and more efficient to implement under the umbrella of Lynx. The same is true for the guidelines of the Text Encoding Initiative (TEI) [4], which we also thoroughly examined for potential application in Lynx, arriving at the same conclusion. In the future, interoperability between Lynx Documents and Akoma Ntoso or TEI documents can be easily achieved using transformation tools such as XSLT stylesheets or Python scripts, among others.

The essence of a Lynx Document is the text element. Therefore, a Lynx Document is an identified piece of structured text plus annotations. Much like ELI documents, Lynx documents are identified by a URI and follow the data model defined by an OWL ontology, the Legal Knowledge Graph Ontology.[8]

The notion of a Legal Knowledge Graph (LKG) may suggest including courts, judges, jurisdictions, abstract legal ideas and other general concepts. The Lynx Legal Knowledge Graph, however, does not contain such an assortment of entities, the focus is placed instead on documents and terminological information, serving the purpose to represent multilingual legal information. The main entity, the Lynx Document, comprises both data and metadata and is the most important entity in the Legal Knowledge Graph. Lynx Documents can be grouped in Collections and eventually enriched with Annotations.

Lynx Documents are the basic information units in Lynx: identified pieces of text, possibly with structure, metadata and annotations. A *Lynx Document Part* is one part of a Lynx Document, possibly arranged hierarchically (chapter, section, article, etc.). *Collections* are groups of logically related Lynx Documents, e. g., one collection per use case, jurisdiction, etc. *Annotations* are enrichments of Lynx Documents, such as summaries, translations, recognised entities, etc., these annotations are NIF-compatible (NLP Interchange Format). Original documents are harvested in their original form first, transformed into Lynx Documents and then enriched with annotations.

Lynx documents can reference each other in different manners (e. g., a law implementing a directive, a standard referencing legislation), and they can reference terms and concepts present in terminologies — documents and terminologies are the main elements in the LKG. In addition, a few other elements are natively present in the LKG, i. e., companies and relevant persons. Finally, every element in the LKG is connected to external entities: companies are connected to their external reference (the one provided by Refinitiv's PermID) and to NACE codes to classify their activity, terminologies are connected to other term banks (IATE, Wikidata, etc.), documents are connected to the original sources etc. (see Fig. 6). The LKG is not a closed graph but it is connected to other entities, many of them in the Linked Data Cloud.[9]

The LKG has multiple advantages. Citizens and companies are provided with better access to legislation by finding homogeneous practices throughout Europe; companies are able to run legal information systems more smoothly, and documents are uniquely identified and explicitly described in this context.

3.1.1. Data validation

Data integrity is critical for the design, implementation and usage of any system which stores, processes or retrieves data. The concept becomes even more crucial in our case, since we use an RDF-based data model, which offers a certain amount of flexibility, i. e., any node can in principle have any number of values, possibly of different type, for any given property. However, in some cases it makes sense to specify conditions defining which properties can be applied to nodes (or restricting its value type). We defined a set of conditions[10] using SHACL[11] (Shapes Constraint Language), a validation technology for RDF data.

3.2. Document Manager (DCM)

The Document Manager (DCM) is an integral component of the LynxSP. This is where documents are stored, maintained and accessed. Its basic functionality includes the storage of documents, their metadata and annotations produced by enrichment services, with an emphasis on synchronisation throughout updates, providing read and write access according to the permissions of users and client applications and complex querying. The DCM's REST interface includes a set of Create, Read, Update, Delete (CRUD) APIs to manage collections, documents and the annotations within the LynxSP.

Linked data is typically made available online in an open way, but this is not feasible in all cases. Often, data access has to be restricted to specific users or user groups, which is why external communication is regulated through the API manager. It acts as a central gateway to client applications and users external to the LynxSP. Access to specific collections is authorised using OAUTH2.0.

Supporting the search service, the DCM provides a REST endpoint through which complex queries can be executed such as "which documents contain mentions of *Entity*", or "what are the annotations of type *Place* in document X?".

The use of semantics to formalise the meaning of its classes and properties qualifies the LKG to be called an actual Knowledge Graph. The Lynx data model is an RDF data model. Through their representation as JSON-LD, Lynx documents are not only isolated elements but nodes in the graph as well. This flexible design choice enables the use of different types of databases for storing

6 https://participedia.net/case/5182
7 http://www.oasis-open.org
8 http://lkg.lynx-project.eu/def/
9 https://github.com/lod-cloud
10 http://lynx-project.eu/doc/nif-shapes.ttl, http://lynx-project.eu/doc/lkg-shapes.ttl
11 https://www.w3.org/TR/shacl

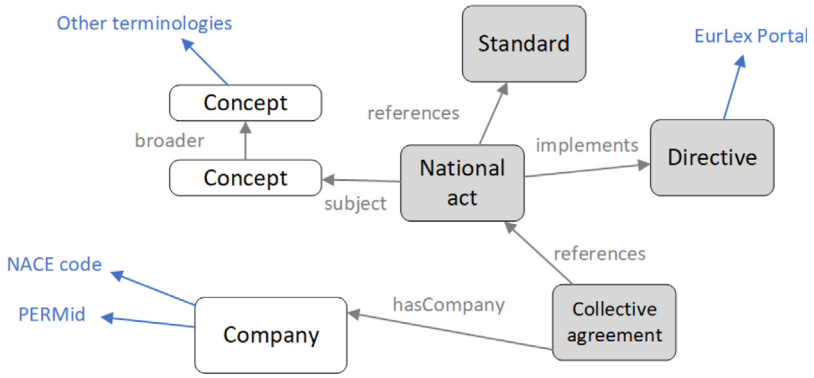

Fig. 6. The Lynx Legal Knowledge Graph (LKG).

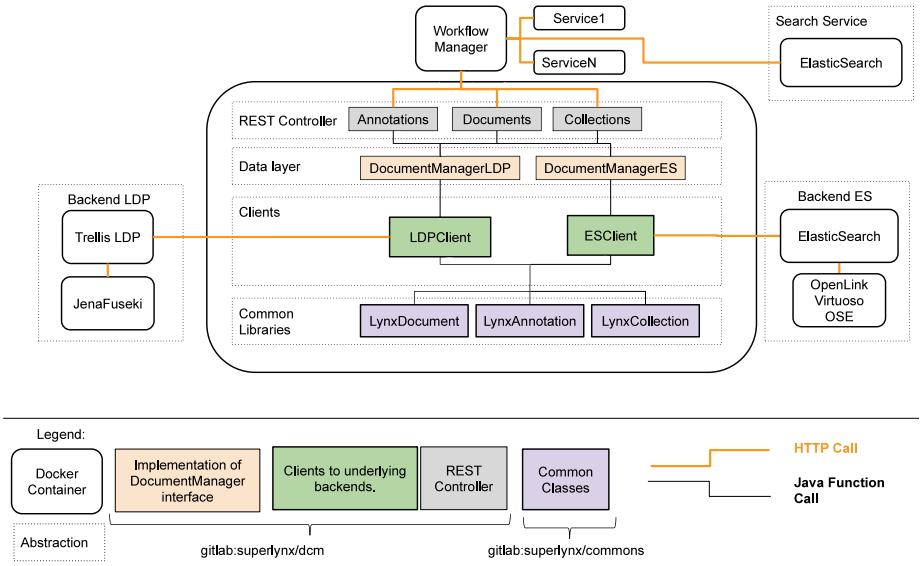

Fig. 7. Document Manager Architecture.

the documents (see Fig. 7). The DCM's design allows different implementations of the data storage layer. We developed and used two different implementations. The first one is implemented as a Linked Data Platform (LDP) server based on Trellis[12] on top of a Jena Fuseki triplestore,[13] supporting RDF natively. The second implementation is based on the ElasticSearch engine, storing the documents as JSON(-LD) documents. Data exports are stored periodically in a public Virtuoso triple store,[14] making data queryable through a SPARQL endpoint.[15] Document structure information and various types of metadata such as subject, jurisdiction, language etc. are triple'fied by the DCM at ingestion time. The NLP Interchange Format (NIF) [5] ontology version 2.1 is used for describing the structure metadata and a mashup of metadata-specific ontologies are used for other descriptive, structural or administrative metadata. The annotations proper of each document are also described using NIF V2.1.[16] Triples from all documents including data and metadata can be queried using the SPARQL endpoint. Extensive usage of vocabularies as values for metadata or annotations increases the value of the LKG and the interoperability of the system. The DCM is the main building block of the Lynx Legal Knowledge Graph, it is where the LKG resides.

3.3. Workflow Manager (WM)

The Workflow Manager (WM) is responsible for the execution and management of workflows, which are models for sequences of tasks. Tasks can be seen as atomic processing steps, they usually consist of processing data and applying operations to databases. Fig. 8 shows the architecture of the WM and how users can interact with it. Each task executor instance is a programme, which executes specific tasks published by the WM service. In order to share large data objects that have to be processed, the WM and task executors use a shared memory service. The WM service provides two REST APIs: the Camunda REST API and a custom REST API designed to accommodate the Lynx requirements.

To manage the ingestion of documents into the LKG, we defined a population workflow. An instance of the population workflow takes as input an RDF Lynx document, enriches it according to the provided enrichment configuration, stores the resulting enriched document in a Document Manager collection and, optionally, indexes it using the Search Service (SEAR). Using enrichment configurations users can specify, which enrichment services to activate, and, for each of them, what model should be used. This workflow is shown in Fig. 9.

12 https://github.com/trellis-ldp/trellis
13 https://jena.apache.org/documentation/fuseki2/
14 http://vos.openlinksw.com/owiki/wiki/VOS
15 http://sparql.lynx-project.eu
16 http://lynx-project.eu/data2/data-models

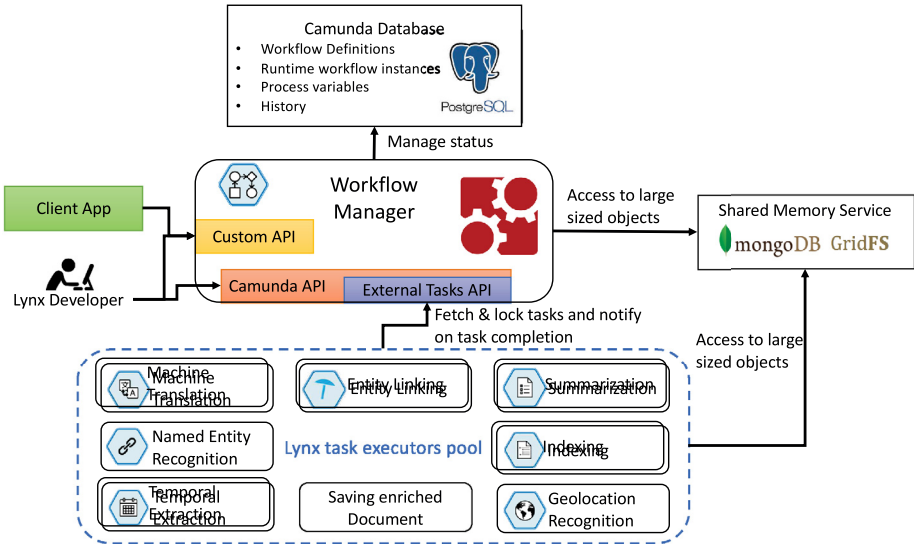

Fig. 8. Workflow Manager (WM) architecture.

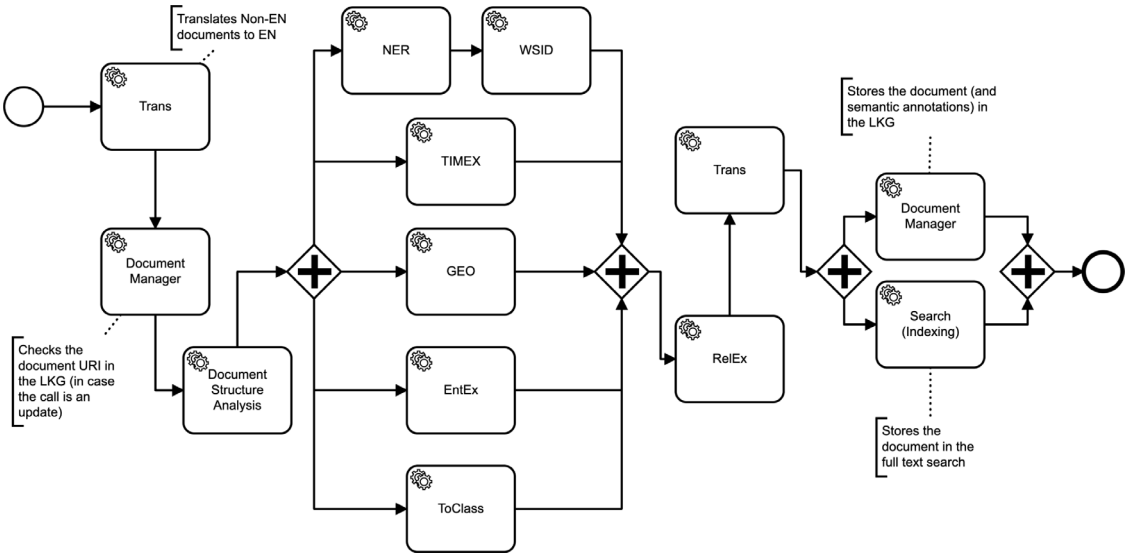

Fig. 9. Legal Knowledge Graph population workflow.

4. Linguistic resources

The linguistic resources that have been created and integrated in the LynxSP and in the LKG comprise domain-independent vocabularies (dictionaries) as well as domain-dependent ones (terminologies).

4.1. Domain-independent vocabularies

The layer offering reliable linguistic services is supported by the integration of domain-independent and domain-dependent vocabularies. While the latter pertain to terminologies, the former provide a common ground across domains that facilitates traversing semantically annotated documents from different specialised domains, and support certain NLP functionalities, such as Word Sense Disambiguation (WSD), by providing a common catalogue of word senses. Such domain-independent data is based on general-language multilingual lexicographic resources, provided by Lynx consortium partner K Dictionaries (KD), for Dutch, English, German and Spanish. They are provided through a JSON API and, primarily, through their Linked Data (LD) version, based on RDF, which guarantees straightforward integration into the LKG. The linguistic information provided by this resource is used by several services such as WSD, Search, QADoc (Section 5), and to retrieve synonyms, term variants and translations that help in the cross-lingual search and question answering.

The semantic representation of the data as LD [6,7] is based on the OntoLex lemon [8] model and its lexicog module.[17] OntoLex lemon is the result of the W3C Ontology Lexica Community Group, which has evolved since 2011 for building a model to initially serve as an interface between an ontology and the natural language descriptions associated with the different elements of an ontology. In recent years, however, this model has been increasingly used to represent lexical resources as LD. It consists of an RDF model that, with a set of core classes (such as, e.g., `ontolex:LexicalEntry` or `ontolex:LexicalSense`) and various modules, allow for the representation of a wide range of lexical descriptions including morphosyntactic properties, translations, and pragmatic information. Along with OntoLex,

[17] https://www.w3.org/2019/09/lexicog/

KD's RDF format also makes use of LexInfo, an ontology serving as a linguistic category registry, which is widely used with OntoLex. Although the latest revision of KD's conversion attempted to align the KD DTD values with LexInfo's most recent version,[18] aiming at a conversion as universal as possible, a custom ontology for the KD data was required to encode the linguistic description provided by those features which could not be mapped to LexInfo elements due to mismatches or granularity differences.

We developed an RDF version of the KD data to improve interoperability, both internally, within their own family of dictionaries, and with external resources. A key element of our approach is the URI naming strategy [9]: the unique identifiers of the dictionary elements were specified with reusability and linking in mind, aiming at preventing the collision of identifiers, and ensuring the reuse of URIs of already defined lexical entries across dictionaries of the series [10]. As an example, the URI of the entry *cintura* 'waist' in Spanish would read: `:lexiconES/cintura-n`. The use of the pattern *lemma – part-of-speech* in the URI facilitates the creation of lexical entries on the fly during the transformation of a dictionary to RDF, as well as their linking across datasets without turning to the original internal dictionary identifiers. At the same time, appending the part-of-speech to the lemma prevents the collision of different lexical entries with the same lemma in a given dictionary. By relying on this URI naming strategy and extracting embedded dictionary elements (e. g., synonyms, antonyms, compound terms and translations of a headword), and treating them as entries of an OntoLex lemon lexicon, we foster reuse of lexical content present at different levels of the dictionary (as regards both its macrostructure and microstructure) and allow for complex queries over these (initially embedded) data.

The process of converting KD data into LD was carried out following an incremental approach, starting with the very basics of a single entry (headword, part of speech, senses, definitions) and proceeding with more complicated elements (synonyms, compounds, examples of usage, translations, etc.), validating the results of the conversion after each iteration.

The incremental process has not only assured constant validation and error handling, but also allowed for an adaptation period, during which the process of writing queries for validation has shed more light on the model and methods of improvement. Taking into account the data requirements of the Lynx services and our initial experiments with the RDF data, we have been able to improve the queries and iteratively change the model so that the results optimally represent the actual users' needs.

The ensuing dataset encompasses a wide array of lexicographic components, including the headword, part of speech, inflections, grammatical information, examples of usage, multiword expressions, synonyms and antonyms, and translation equivalents. The most recent conversion introduced a distinction between lemmas (or *canonical* forms in OntoLex terms) and other `ontolex:Form` elements (usually inflected), thus distinguishing lemmas from their corresponding inflections or variants, as a practical response to the request of partners.

An example of these requirements is the need to obtain synonyms for a given term, for instance, Sp. *norma* 'norm, rule'. This data can be now retrieved with a SPARQL query over KD's Global data. The results comprise the terms "canon"@es, "fórmula"@es, "ley"@es, "máxima"@es, "pauta"@es, "regla"@es, "rite"@es. While these terms were originally embedded as synonyms in sense containers of the entry *norma* in Spanish, they are now independent entries in the Spanish dataset, offering easier retrieval, further exploration of the lexical network, or the exploration of their original representation in a (hierarchical) lexicographic structure with queries that leverage the *lexicog* module.

18 https://github.com/ontolex/lexinfo

4.2. Existing terminology collections

The creation and management of knowledge are essential parts of every business, process and use case. To express knowledge in natural language and to understand written information, terminology is needed, it is like a glue between natural language and knowledge systems. The Lynx use cases demonstrate the importance of terminology throughout the steps of information analysis, processing and maintenance. Each new use case will face a specific challenge, i. e., to acquire a terminology collection that covers its business domain and that is also compatible with the LynxSP. To support the existing Lynx use cases and speed up new use case integration, we have identified, gathered and converted certain terminology collections to the linked data format.

The terminology acquisition process consists of the following steps:

1. Identification — web search, literature analyses, search in data catalogues (such as ELRC-SHARE[19] [11], European Language Grid[20] [12], ReTeLe catalogue,[21] ELRA Language Resource Catalogue[22]),
2. Checking licensing information,
3. Making a decision on usefulness (priority to Lynx languages, Lynx domains, licensing status and resources in machine-readable formats),
4. Processing terminology resource and importing into portal,
5. Description of resource with appropriate metadata and registration to the Lynx data portal.

One of the costliest parts of terminology acquisition is terminology conversion. We categorise resources into three groups depending on their input format:

1. Structured data — the best option is to have terminology already in Linked Data or TBX. Structured formats like XML were processed using dedicated scripts and existing mapping tools;
2. Semi-structured data — e. g., Word documents require more complex approaches. Sometimes human experts had to intervene and revise;
3. Unstructured data — (e. g., PDF, JPG) present the most complicated situation, information extraction methods need to be applied in addition to manual operations. Optical Character Recognition was also applied.

Additionally, terminology platforms were used including tools from the Tilde Terminology[23] platform. For management and storage we used the EuroTermBank[24] portal, a network of stakeholders for publishing and hosting EU-related open terminology data. The Lynx data portal contains certain terminology collections. Use cases can benefit from that terminology, they are available in TBX, RDF of TSV/CSV formats or they can be download as spreadsheets.

4.3. Term Extraction (TermEx)

Complementing the three pilots, three types of domain-specific vocabularies, i. e., terminologies, had to be prepared. We followed a series of linguistic and semantic processes that minimise the manual workload of this task.

19 http://www.lr-coordination.eu
20 https://www.european-language-grid.eu
21 http://catalogo.retele.linkeddata.es
22 http://catalogue.elra.info
23 https://term.tilde.com
24 https://www.eurotermbank.com

First, we used Tilde's Terminology Extraction services to get the most representative terms from the different corpora. This service also provided a piece of context, a window of text surrounding the term, that could afterwards be used to disambiguate (when required). The automatically extracted terms were later semi-automatically cleaned. We analysed the results from a linguistic perspective and elaborated a series of linguistic rules to delete everything that could not be considered a *term* (for instance symbols, adverbs, temporal expressions and certain named entities).

At this stage, we had filtered lists of domain-specific monolingual terms. However, the LynxSP services require additional terminological information: synonyms, translations, definitions, etc. To retrieve this information, we relied on the *Linguistic Linked Open Data cloud*[25] (LLOD), from which we retrieve data from diverse resources: EuroVoc,[26] Unesco Thesaurus[27] and Wikidata[28] (translations, synonyms and hierarchical relations). We have also retrieve data from Lexicala API,[29] from which we retrieved synonyms, translations and definitions, and from IATE,[30] that, although it was transformed into RDF in [13], we have used the JSON API[31] which returns a more updated and complete content, which includes translations, alternative labels, definitions, usage notes, references and related terms. Since Wikidata, Lexicala and IATE belong to the general domain, meaning that the terms contained are not univocal and may have more than one sense, a sense disambiguation step was required. Therefore, in order to link the extracted terms which their adequate matches in the general knowledge bases, we implemented an Word Sense Disambiguation (WSD) algorithm,[32] based on BERT,[33] provided by Semantic Web Company. The algorithm receives as input a series of *sense indicators* used to represent the sense of a term. This is, one source sense indicator, built by the source term and its context, is compared with several candidate sense indicators from the knowledge bases, built with any piece of information available, such as definitions, synonyms, broader, narrower or related terms, etc. From the most similar sense, we retrieved additional data to enrich the term. We also added new entries to the initial terminology, i. e., the terms that have a broader, narrower or related relation with our source terms.

Following this step, we grouped the final list of terms to create hierarchies, based on those that share similar tokens. Finally, we converted the resulting terminologies into RDF, following the SKOS[34] model and helped by Dublin Core[35] ontology for metadata (see Table 1), obtaining a series of domain-specific terminologies linked to the LLOD. The resulting terminologies were duly post-processed by professional linguists and legal experts from different partner entities in the project.

5. Semantic services

This section describes, in detail, the semantic services of the Lynx Service Platform and their most recent improvements.[36]

5.1. Named Entity Recognition (NER)

Named entity recognition is a well-known NLP task. NER systems use models to annotate named entities, where the models are trained on language data in which different types of entities are annotated. The most common types are *person*, *organisation* and *location*. Using the trained models, the system can identify and annotate entities that were not present in the training documents. Many different methods have been applied for the recognition of named entities depending on the domain and application. For the Lynx NER service we experimented with four different approaches: (i) statistical language model; (ii) Bidirectional Encoder Representations from Transformers (BERT); (iii) Conditional Random Fields (CRF); and (iv) Bilateral Long Short Term Memory Neural Networks (BiLSTM).

Statistical language model. It is implemented using OpenNLP's Name Finder module,[37] an open-source NLP framework. The Name Finder can detect named entities and numbers in text for which it needs a model, which is dependent on the language and entity types it was trained for. This approach aims to identify general rather than domain-specific entities. We trained four different models using the training data provided by Nothman [16]. The four models cover two languages, English and German, and two types of entities, *person* and *organisation*: English-PER, English-ORG, German-PER, German-ORG.

BERT. The second approach is based on the work of Kamal Raj.[38] We have adapted it to be able to train new models in the four languages of the project: English, German, Spanish and Dutch. The recogniser is based on the language model BERT [17] and, similarly to the statistical language model, it was trained using WikiNer (provided by Nothman [16]). In this case we do not have to train a different model for each entity type but they are all recognised using the same model. Therefore, we have trained four models, one for each required language in the project: English ('BERTNER_EN'), German ('BERTNER_DE'), Spanish ('BERTNER_ES') and Dutch ('BERTNER_NL').

CRF and BiLSTM. For CRF and BiLSTM we use two sequence labelling tools, sklearn-crfsuite,[39] and UKPLab-BiLSTM [18]. To adapt the CRF and BiLSTM approaches to the needs of the project, i. e., to the legal domain, we created a dataset containing annotated legal entities [19]. Detailed descriptions of the dataset, the adaptation process of the CRF and BiLSTM and the evaluation can be found in [19,20].

5.2. Entity Linking (EL)

The Entity Linking (EL) service combines the functionality of entity extraction and disambiguation. Entity extraction enables the insertion of links between documents and elements of controlled vocabularies in the LKG. These relations are the first step for enriching text fragments with knowledge from the LKG. Importantly, the inclusion of labels in many languages allows linking of documents in different languages, combining the knowledge derived from them, as well as multilingual search and recommendation. Entity extraction can be performed in as many languages as the terminologies have labels in, and thus we can leverage multinational efforts for creating multilingual terminologies such as EUROVOC[40] or UNBIS[41]. The Entity Extraction is performed

25 http://linguistic-lod.org
26 http://eurovoc.europa.eu
27 http://vocabularies.unesco.org
28 https://www.wikidata.org
29 https://api.lexicala.com
30 https://iate.europa.eu
31 https://iate.europa.eu/developers
32 https://github.com/semantic-web-company/ptlm_wsid
33 https://github.com/google-research/bert
34 https://www.w3.org/2004/02/skos/
35 http://dublincore.org
36 Initial descriptions of the Lynx services are provided in [14,15].

37 https://opennlp.apache.org/docs/1.8.3/apidocs/opennlp-uima/opennlp/uima/namefind/NameFinder.html
38 https://github.com/kamalkraj/BERT-NER
39 https://pypi.org/project/sklearn-crfsuite/
40 https://publications.europa.eu/en/web/eu-vocabularies/
41 http://metadata.un.org/?lang=en

Table 1
Example of a term entry modelled in RDF. Term relations hold amongst terms in the terminology (such as `broader` and `related`), or amongst concepts in external resources (such as `narrower`). The `skos:note` property is used to represent the context from which the term was extracted, and the `dc:jurisdiction` to expose the jurisdiction to which the source corpus applies. The `skos:closeMatch` property is used to represent links with external resources in RDF (such as Wikidata, Unesco Thesaurus and EuroVoc), and `dc:source` represents other resources from which the information was extracted (such as IATE and the Lexicala API, that are not available in Semantic Web formats).

Properties Applied	Term Entry Example
`skos:Concept`	http://lynx-project.eu/kos/LT7588489
`skos:prefLabel`	"lawyer"@en, "abogado"@es, "Advokat"@de, "advocaten"@nl
`skos:altLabel`	"attorney"@en
`skos:definition`	"professional who provides legal counsel and who represents clients in proceedings of various kinds"@en
`skos:note`	"lawyers give advice to their customers about the disagreements inside the court"@en
`skos:broader`	http://lynx-project.eu/kos/LT9644423 ("legal bar"@en)
`skos:narrower`	https://www.wikidata.org/wiki/Q512345 ("criminal defense lawyer"@en)
`skos:related`	http://lynx-project.eu/kos/LT5019609 ("law"@en)
`skos:closeMatch`	https://www.wikidata.org/wiki/Q40348
`dc:source`	https://iate.europa.eu/entry/result/3589074
`dc:jurisdiction`	UK

using the PoolParty Semantic Suite, provided by Lynx partner Semantic Web Company.[42]

If entities in the terminologies share the same label, i.e., if a label is ambiguous, each mention is disambiguated. The disambiguation mechanism decides which entity from the LKG should be linked to the considered context. For this purpose we use either DistilBERT [21] or BERT [17]. For a given contextual mention of an ambiguous label we retrieve all entities from the LKG that share this label. Next, for each candidate entity we query the superclass of the entity in LKG – we exploit these as sense indicators. Following the methodology of the unsupervised baselines Task 2 in [22], the context and superclasses are used as input for the language model to produce word representations and estimate the similarity between the target label in the specific context and the superclasses.

The EL service is a prerequisite for several other Lynx services. The RelEx service first finds known entities in a document, and then recognises whether a given relationship is expressed between them. The SeSim service makes use of linked entities, as they serve to compute similarity between snippets of text using the information explicitly contained in them and the information from the LKG. Likewise, the QA service uses extracted entities to enhance the information about the query and the documents from which answers are to be retrieved.

5.3. Temporal Expression Analysis (TimEx)

The Temporal Expression Analysis (TimEx) service handles temporal expressions, including any word or sequence of words referring to a time instant (e.g., "five o'clock") or interval (e.g., "from nine to ten"). This task includes two subtasks; first, temporal expressions need to be *identified* in a text; second, the temporal expression has to be *normalised* to arrive at a specific date from a relative expression such as "tomorrow", based on a reference date (e.g., a date mentioned in the text or the creation date of the document). Our annotations follow the ISO-TimeML standard [23] and distinguish four types of temporal expressions: dates (expressions such as "October 7, 1991", "22/01/2018", or relative expressions like "two days ago"), times (points in time like "at seven o'clock" or "3:30pm"), durations (e.g., "three years and six months", "two centuries" or "half an hour") and sets (repetitions in time, such as "monthly", "twice a week", "every first of the month"); in addition, the annotation of intervals (i.e., periods between two temporal expressions, such as "from 14 h to 20h" or "from October to December") is also available in the Spanish language service, and will be eventually added to the English service. This service is rule-based and able to handle temporal expressions in English, Spanish, German, Dutch and Italian. While for the first three languages specific approaches have been developed to target temporal expressions in the legal domain, Dutch and Italian use default functionality of a third-party component, HeidelTime [24].

Spanish and english. For these languages we use the software Añotador [25], that is built on top of CoreNLP [26] and applies a series of rules to detect temporal expressions using TokensRegex [27]. The rules take into consideration problems that generic temporal taggers tend to have when processing legal texts, such as the appearance of dates as part of legal references (e.g., in "the Council Directive 93/13/EEC of *5 April 1993*", the emphasised date is part of a reference to a legal document, i.e., it is *not* a date referring to the narrative of the text) and the wrong normalisation it implies for the surrounding temporal expressions (e.g., if we had considered "5 April 1993" as a temporal expression, a surrounding expression such as "the following month" would be considered by most taggers as anchored to "5 April 1993" and, therefore, referring to May 1993). To learn how the service should deal with these kinds of particularities, the corresponding needs and requirements were collected with the Lynx pilot partners. Together with the expertise provided by legal experts, this led to improvements, such as the addition of intervals as a type of temporal expressions and the coverage of new temporal expressions with new normalisation patterns (e.g., "one working day", that following the standard annotation would be normalised as "P1D", the same as "one day", while our service normalises it as "P1BD", where "BD" stands for "business day"). The service has been evaluated against several corpora, both from the legal domain (the TempCourt corpus [28]) and general texts (the HourGlass corpus [25]).

German. Due to the lack of a suitable corpus in the domain, a small collection of German texts was annotated with temporal expressions following the TimeML standard. As with the Spanish and English texts, one of the specifics of the domain identified in German texts are references to other legal texts which contain (alleged) dates (e.g., "Richtlinie 2008/96/EG", "Directive 2008/96/EG"). Other peculiarities of the domain and the German language are frequent use of compounds such as "Kalenderjahr", "Fälligkeitsmonat" or "Bankarbeitstag" (*calendar year*, *due month*, *banking day*), generic use of temporal expressions such as "jeweils zum 1. Januar" (*1st January of each year*) and event-anchored temporal expressions "Tag der Verkündigung" (*Proclamation Day*). Based on the newly annotated corpus, the temporal tagger HeidelTime [24] was adapted to the domain covering these phenomena. Our evaluation shows that this approach outperforms Heideltime.

42 https://www.poolparty.biz

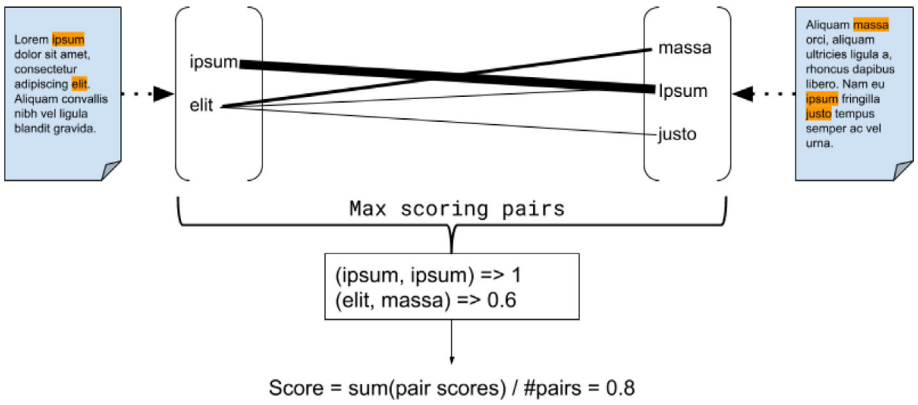

Fig. 10. Semantic similarity computation.

Regarding how the temporal information is represented, we do not use time specific ontologies such as the Time Ontology[43] because it would not offer any advantage with regard to maintaining the TimeML format. Using a time related ontology would imply to convert the TimeML normalised value format (that is already standard, since it is a string following the 8601 ISO) to the different classes in the ontology (i. e., the string "2012-03-20" would be a date with a day (20), a month (03) and a year (2012)), that represents no improvement with regard to maintain the string, since no latter services would use this new format for reasoning, search, or any other additional use. Additionally, using a String as a value allows us to add values specific to the legal domain (such as "Workday") that are not represented in existing ontologies, to the best of our knowledge. Finally, for the possible type of temporal expression (*DATE, SET, DURATION, TIME*), we found no ontology that takes into account *SETs*, so the used solution (using `itsrdf:taClassRef` with constrained values lkg:{DATE,TIME,SET,TIME} seemed the optimal solution.

5.4. Semantic Similarity (SeSim)

We use a hybrid similarity measure. First, the text of the document is annotated and linked to the LKG, including, among others, the resolution of temporal and geographical references. Second, similarity is computed using a linear combination of text-based and knowledge-based similarities. The former are encoded by cosine-similarity of TF–IDF vectors and the latter by the overlap as measured by the Jaccard coefficient of entities that two documents either mention directly or indirectly, through links to the LKG. The overlaps are weighted depending on the path distances between the entities, mentioned in the document, in the LKG (Fig. 10). This approach allows us to measure the similarity between two documents even if they have only few entities in common. The service is used in the GTE use case (Section 2.1).

Evaluation. No existing benchmark was identified as suitable for the evaluation of this service. Therefore, we collected a dataset manually. The Stackexchange[44] data of chemistry pairs of questions, some of which are marked as duplicates, was used for these experiments. Stackexchange includes manually curated information about duplicates. Unfortunately, we could not find a similar resource for the legal domain. We used MeSH[45] as the controlled vocabulary for the annotation of the dataset. The Medical Subject Headings (MeSH) thesaurus is a controlled and hierarchically organised vocabulary by the National Library of Medicine (NLM).

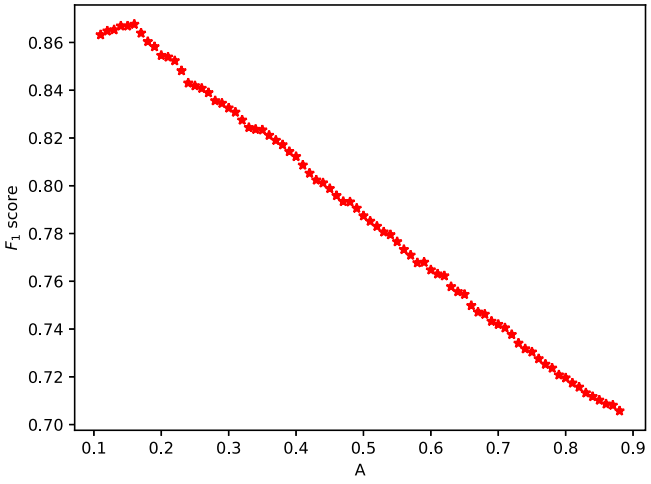

Fig. 11. Dependency of F_1 scores of the combined similarity on parameter A.

It is used for indexing and cataloguing biomedical and health-related information. MeSH includes the subject headings from MEDLINE/PubMed and other NLM databases. The dataset contains 660 questions marked as duplicates and an additional 861 pairs of non-duplicates collected manually from questions not marked as duplicates[46]. Each question was transformed into two different representations: *Text*, the text itself, no words were ignored, only mathematical equations and HTML code were removed; and *Concepts*, URIs of MeSH concepts whose `prefLabel` or `altLabel` is present in the text.

We used the SeSim service to compute the different similarity scores for the documents. The *text similarity* scores are based purely on the frequency of overlapping tokens. *Semantic similarity* refers to the method shown in Fig. 10. The *concept-based similarity* score is based on comparing the number of overlapping annotations. The *combined similarity* score is obtained using the formula: $A * (\text{semantic} - hv_{th}) + (1 - A) * \text{text}$. As we have the freedom to choose the classification threshold, it suffices to only variate one parameter. Therefore, for historical reason we set the value $hv_{th} = 0.65$. In order to identify the best value for A we compute the F_1 score and the best classification threshold for the different values of A, see Figs. 11 and 12. The best value of parameter A is determined to be 0.16, the best threshold is ≈ 0.04.

Though the combined score does not uniformly beat all scores, it is the most robust one. Indeed, for all documents (Table 2) it

43 https://www.w3.org/TR/owl-time/
44 https://chemistry.stackexchange.com
45 https://www.nlm.nih.gov/mesh/meshhome.html
46 The dataset is openly available at https://zenodo.org/record/4590265.

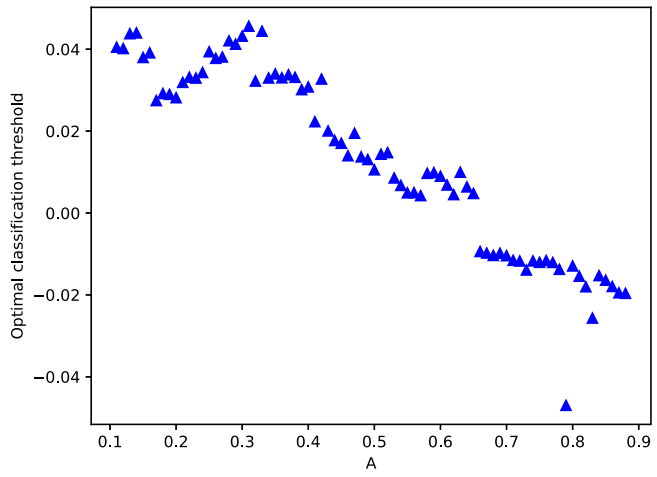

Fig. 12. Dependency of optimal classification thresholds of the combined similarity on parameter *A*.

Table 2
Results of identifying duplicate questions (all questions). **Bold** marks the highest score in the respective column.

Similarity	Accuracy	Precision	Recall	F_1
Text	0.8416	**0.9320**	0.6848	0.7895
Semantic	0.7304	0.7441	0.5773	0.6502
Concept-based	0.8468	0.8875	0.7409	0.8076
Combined	**0.8882**	0.8926	**0.8439**	**0.8676**

Table 3
Results of identifying duplicate questions (only questions containing more than five annotations taken into account). **Bold** marks the highest score in the respective column.

Similarity	Accuracy	Precision	Recall	F_1
Text	0.8448	**0.9126**	0.6746	0.7758
Semantic	0.7657	0.7098	0.6962	0.7029
Concept-based	**0.8943**	0.8717	0.8612	**0.8664**
Combined	0.8886	0.8575	**0.8636**	0.8605

yields the best scores, and for well annotated documents (Table 3) it does not lose much in performance.

5.5. Question Answering (QA)

The Question Answering (QA) service accepts a natural language question and responds with an answer, extracted from a document in a given corpus. The end-to-end system consists of three components: 1) The Query Formulation module transforms a question into a query, which can be expanded using a domain-specific vocabulary from the LKG. The query is processed through an indexer to obtain matching documents from the corpora; 2) The Answer Generation module extracts potential answers from the retrieved documents from the LKG; 3) The Answer Selection module identifies the best answer based on various criteria such as local structure of the text and global interaction between each pair of words based on specific layers of the model.

The service was originally deployed with the QANet model [29] as the Answer Selection modules. QANet processes inputs in English and therefore relies on the translations produced by the MT service. Recently, BETO [30] was introduced. The BETO model achieves state-of-the-art results for Spanish Question Answering. The Spanish QA will work the same as the English QANet model; it will be based on BETO and processes documents originally in Spanish or translated into Spanish.

5.6. Search Service (SEAR)

SEAR is a full text search service based on ElasticSearch,[47] which is a RESTful search engine. SEAR provides full text, boolean search with filter capabilities for multiple languages (English, Spanish, German, Dutch). Indexing is performed with special analysis for each language. It makes use of the annotations done by other Lynx Services, e. g., NER, GEO, etc. Since most industry full text search solutions are only built to support monolingual search, they will not translate the query to support queries in other languages. Our solution allows cross-lingual search, i. e., users can search for documents in other languages than the search query's language. The solution consists of the following components:

1. The Indexer module processes a given document and adds it to the index.
2. The Search module processes a search query and returns a result set. It makes use of the following modules:
   (a) The Query Analysis module identifies various information from the query, e. g., the source language, and the jurisdictions in which the query should be performed. For example, if the query is "maternity leave in Austria and Holland", the source language is English, the query string is "maternity leave", and jurisdictions are "Austria" and "the Netherlands".
   (b) The Query Parser module parses the query.
   (c) The Query Expansion module expands the search string by using lexical and terminological information, translates the search terms to the target language of the corpora, and generates the final query. The performance losses due to translation are considerable. Without translation, a query takes about 50 ms to 100 ms, depending on the complexity. A translation cycle takes about 750 ms to 900 ms even for a simple term. So this time is always in addition to the normal query time. Currently there is no optimisation here, but in a final production system translated terms are cached so that frequently used terms are taken out of the cache. This would reduce the query time significantly and be more similar to the initial query time.
   (d) The Query module performs the search in the multilingual corpora.

5.7. Summarisation (SUMM)

To enable users to get a quick overview of the main ideas of a specific piece of content (paragraph, text, document, multiple documents), methods for document summarisation have been integrated into the LynxSP. They provide additional layers of useful annotations that enable the human experts to comprehend a document faster and more efficiently. We developed two different approaches.

Extractive summarisation. Centroid Summarisation is an unsupervised extractive summarisation method suitable for processing single or multiple documents [31,32]. The representation model assigns a score to each sentence. We first collected a reference corpus that consists of documents from the same field. If news articles were to be summarised, then the reference corpus would include articles from different newspapers. We first compute IDF scores over the reference corpus after removal of stopwords, we

47 https://www.elastic.co/elasticsearch

then calculate the TF–IDF scores for all non-stopwords appearing in texts. This way we can create a weighted list of words, with their weights representing their relevance to the document. We then select all words with a weight greater than a certain threshold and retrieve their embeddings. The properties of word embeddings were used to create a centroid vector for one or multiple documents. This vector represents the condensed meaningful information of one or more documents and is calculated by adding up the embeddings of the most relevant words.

To narrow down the number of sentences to extract from, we calculate the relevance of each sentence. By adding up the word embeddings, the sentences were then embedded. The closeness of the sentence embeddings to the centroid embedding represents their relevance to summarising the document. To create the summary first the sentences closest to the centroid are picked. Until the summary length is reached the sentences are added iteratively in order of their closeness to the centroid. But before adding a new sentence to the summary it is compared to every sentence already in the summary. This is done to avoid redundancy and to add different information to the summary. The cosine similarity between the two sentence embeddings is computed. If the sentences are more similar than a set threshold, it is assumed that it would not add much new information to the summary and is, therefore, skipped.

Abstractive summarisation. This summarisation model is based on the Transformer architecture, it adopts the model of Vaswani et al. [33]. On top of the decoder, we use a Pointer-Generator to increase the extractive capabilities of the network. Aksenov et al. [34] describe the approach in more detail.

5.8. Geolocation (GEO)

This service is responsible for the annotation and linking of geographical information in documents from the legal domain. It accepts text as input, both in plain text or LynxDocument format (NIF-based format). The text is analysed using one or several of the methods described below, and returns a LynxDocument containing annotations for each of the geographic entities (*itsrdf:taClassRef dbo:Location*). Apart from the Entity Type annotation, it also links the entity with an external linked data source (*itsrdf:taIdentRef*) for every entity. It is based on three different methods for annotating geographical entities: 1. Language Models; 2. Dictionaries; 3. Rules.

Language models — statistical models. The language model method uses the same approach as the one described for NER (see Section 5.1). For the linking, it uses Geonames URIs.[48] We trained two different models using OpenNLP using the training data provided by Nothman [16]. The models cover two languages, English (English-LOC) and German (German-LOC). Below, we focus on the fine-grained classification of geographical entities with 14 additional fine-grained location subcategories (e. g., city, country, state, park etc.) plus three main categories (organisation, person, other). We use this scheme to annotate manually a small English dataset consisting of 92 texts on the Berlin district of Moabit, crawled from the internet. The dataset has a total number of 3432 sentences and 57,067 tokens with an average sentence length of 16.6 tokens. We use this dataset to evaluate two approaches: HMM (using the module in NLTK[49]) and CRF (using the CRFTagger in NLTK). To compute the F-score, the dataset was divided into a train set (2745 sentences) and a test set (687 sentences). The results are shown in Table 4.

Table 4
Geolocation annotation: evaluation results based on the Moabit Dataset.

	Precision	Recall	F_1
HMM	81.42	90.23	85.59
CRF	93.55	94.62	93.85

Dictionary-based method. The second model uses a dictionary for lexicon-based proper noun identification and only has limited mechanisms for disambiguation. The method is based on the DictionaryNameFinder[50] module of OpenNLP. It allows the spotting of entities defined in dictionaries.

Rules-based method. This approach uses a set of manually specified rules to identify geographical entities, defined as regular expressions that are checked against the text using the RegExNameFinder[51] module of OpenNLP. The two previous methods are not suitable for very fine-grained geographic entities. We use rules because it proved difficult to identify addresses using language models or dictionaries, since the streets can have various names. Addresses have a rather fixed structure, which enables their recognition using regular expressions. We defined rules for the identification of addresses in the four languages relevant in the project: English, German, Spanish, Dutch (see Table 5).

While this approach is useful in practice, we know that many variations cannot be covered by these rules. Therefore, we are still improving the rule-based method in order to increase recall, recognising a wider variety of addresses.

5.9. Neural Machine Translation (NMT)

Europe's multilingualism [35] and the ability to deal with documents in this multilingual landscape is one of the main aspects of LynxSP, which is why various Machine Translation services have been integrated in the platform. As all use cases require a certain translation quality level, we created custom machine translation systems for each pilot based on their business needs and multilingual strategy. Lynx consortium partner Tilde provided its cloud-based MT platform[52] and trained the Neural Machine Translation (NMT) services. The platform provides all needed facilities for customising NMT systems for specific languages and domains. In total, we customised the following ten NMT systems:

1. English–German and German–English
2. English–Spanish and Spanish–English
3. German–Dutch and Dutch–German
4. English–Dutch and Dutch–English
5. Spanish–German and German–Spanish

NMT customisation started with data collection activities, i. e., the use case partners selected multilingual or monolingual documents from their own document libraries or they provided lists of online resources with related content. We prepared the translation corpora using a parallel corpus creation workflow and by preparing corpora based on crawled web data. Large amounts of data are needed to train an NMT system so that it produces translations with sufficient quality. Since our use case partners did not have the required amounts of data available, we added existing corpora within the specified domains.

48 http://www.geonames.org
49 https://www.nltk.org
50 https://opennlp.apache.org/docs/1.7.0/apidocs/opennlp-tools/opennlp/tools/namefind/DictionaryNameFinder.html
51 https://opennlp.apache.org/docs/1.8.4/apidocs/opennlp-tools/opennlp/tools/namefind/RegexNameFinder.html
52 https://tilde.com/mt

Table 5
Results of identifying duplicate documents. All documents.

Language	Rule
English (UK) Addresses	(^|,\s*)((?<number>(\d+[\s-{]*[\d]*)))\s+(?<street>[\wß äöüÄÖÜñ]+)(,?)\s+(?<city>([’\- a-zA-ZßäöüÄÖÜñ]+))\s+(([\d\w][\d\w]+))(,?)\s+(?<country>([\wßäöüÄÖÜñ]+))($)
German / Austrian Addresses	(^|,)(?<street>[\wßäöüÄÖÜ]+)\s+(?<number>\d+[\s-{]*[\d]*) , \s+(?<zip>(\d+))\s*(?<city>([\wßäöüÄÖÜ\s\.]+))
Spanish Addresses	(^|,)(?<street>[\wßäöüÄÖÜÁÉÍÓÚáéíóúñ\(\)\s]+),(\s+(? <number>((\d+[\s-{]*[\d]*))|([Ss]\/[Nn])),)?\s+(?<flat> (\d+\.?\s*(\w|\d)))\s+(?<zip>(\d+))(,?)\s*(?<city>([\wßäöüÄÖÜÁÉÍÓÚáéíóúñ\s\.]+))(,?)\s+(?<region> ([\wßÁÉÍÓÚáéíóúüäöüÄÖÜñ\s]+))($)
Dutch Addresses	(^|,)(?<street>[\wßäöüÄÖÜñ\s]+)\s+((?<number>(\d+[\s-{] *[\d]*)))?\s+(?<zip>(\d+))\s+(?<regioncode>([\wßäöüÄÖ Üñ\s]+))\s+(?<city>([’\- \wßäöüÄÖÜñ\s]+))\s+(?<country> ([\wßäöüÄÖÜñ\s\.]+))($)

The Tilde MT platform supports two types of translation scenarios across LynxSP: *synchronous* requests for short text fragments and an *asynchronous* mode for the scenario where a complete source document is submitted for translation. Once the translation is finished, the translated document can be downloaded. Supported formats are plain text and NIF documents; for document translation multiple document formats are supported, including DOCX, TMX, XLIFF. For Lynx, a NIF document translation was developed, i. e., NIF documents can contain context (full document text), parts (some parts of the original document), annotations added by other Lynx services, and other information encoded in RDF not used by NMT. Finally, NMT appends the translation to the whole context of the original NIF document. For each translated part of the document, the existing annotation is restored using the context and word alignment indexes so that the annotations are preserved.

6. Related work

The platform we developed serves two primary purposes, i. e., generate the Legal Knowledge Graph and semantically process documents from the legal domain. Focusing on the area of legal document processing, technologies from several fields are relevant including, among others, knowledge technologies, citation analysis and information retrieval. Recent literature overviews can be found in [36,37]. In this section we concentrate on systems and platforms similar to the Lynx Service Platform.

Research prototypes. Most research prototypes were developed in the 1990s under the umbrella of Computer Assisted Legal Research (CALR) [38]. In the following we briefly review several of these systems, which usually focus on one very specific feature or functionality. One example is the open source software for the analysis and visualisation of networks of Dutch case law [39]. This technology determines relevant precedents (analysing the citation network of case law), compares them with those identified in the literature, and determines clusters of related cases. A similar prototype is described by [40]. [41] propose a search engine for legal documents where arguments are extracted from appellate cases and are accessible through selecting nodes in a litigation issue ontology or relational keyword search. Lucem [42] mirrors the way lawyers approach legal research, developing visualisations that provide lawyers with an additional tool to approach their research results. The Eunomos prototype semi-automates the construction and analysis of knowledge [43]. The main difference between these tools and LynxSP is the type of documents they work with. Most systems are limited to a single type of document, while we work with a wide variety, from contracts or laws to industrial standards. In addition, each of these tools has a specific functionality, while LynxSP combines them all in a single ecosystem.

Related initiatives. Apart from Lynx, there are other initiatives currently concentrating on the legal domain. Special[53] (Scalable Policy-aware Linked Data Architecture For Privacy, Transparency and Compliance) addresses the conflict between Big Data innovation and privacy-aware data protection, proposing a technical solution that makes both of these goals realistic, allowing citizens and organisations to share more data, while guaranteeing data protection and transparency. The ManyLaws Project[54] (EU-wide Legal Text Mining using Big Data Processing Infrastructures) is a platform set up to deliver a set of services for citizens, businesses and administrations in the European Union. MARCELL (Multilingual Resources for CEF.AT in the legal domain)[55] aims at providing automatic translation on the body of national legislation (laws, decrees, regulations) in seven countries: Bulgaria, Croatia, Hungary, Poland, Romania, Slovakia and Slovenia. e-SIDES[56] explores the societal and ethical implications of Big Data technologies and provides a broad basis and wider context to validate privacy-preserving technologies in EU-funded Research and Innovation Actions (RIAs). The D2D CRC[57] regulatory and legal project was aimed at the development of (semi-)automated legal compliance solutions for information sharing related to the National Criminal Intelligence System.

Commercial systems and services. Some commercial solutions partially bring together most of the functionalities that we have in the Lynx Service Platform. LexisNexis is the market leader in the legal domain; it offers services, such as legal research, practical guidance, company research and media monitoring as well as compliance and due diligence. WestLaw is an online service that allows legal professionals to find and consult relevant legal information.[58] One of its goals is to enable professionals to put together a strong argument. There are also smaller companies that offer legal research solutions and analytic environments, such as RavelLax,[59] or Lereto.[60] A commercial search engine for legal documents, iSearch, is a service offered by LegitQuest.[61] The Casetext CARA Research Suite allows uploading a brief and then retrieving, based on its contents, useful case law.[62] There is also a growing number of startup companies active in the legal domain.

53 https://www.specialprivacy.eu
54 https://www.manylaws.eu
55 https://marcell-project.eu
56 https://e-sides.eu
57 https://www.d2dcrc.com.au
58 http://legalsolutions.thomsonreuters.com/law-products/westlaw-legal-research/
59 http://ravellaw.com
60 https://www.lereto.at
61 https://www.legitquest.com
62 https://casetext.com

NLP/knowledge extraction platforms. Apart from systems focused on the legal domain, there are other platforms and frameworks that can perform document processing and service orchestration. Prominent examples are GATE,[63] UIMA[64] and SparkNLP[65]. UIMA mainly supports automated annotation tools developed in JAVA or C++, although it can support other languages such as Python or Perl. GATE supports JAVA. SparkNLP has implementations in various languages (JAVA, Python, Scala, etc.). UIMA, GATE and SparkNLP force users who want to develop tools or services to implement the respective interfaces of UIMA, GATE or SparkNLP, which was much too restrictive for the Lynx consortium, where the different commercial or academic project partners insisted on complete independence, i. e., for the development or improvement of the services they contribute to LynxSP they wanted to continue developing in their own respective frameworks, programming languages and development environments. Eventually, the orchestration and integration of the services was achieved using the Lynx Service Platform: the independent services only need to be dockerized and comply to our HTTP REST API interfaces.

Ellogon[66] is an NLP platform similar to LynxSP, which also allows for the processing of textual documents and orchestration of services. However it uses the TIPSTER data model. While TIPSTER can in fact be used to represent Linked Data, its implementation would have represented a significant extra effort that would have drastically increased the effort for implementing services in Lynx. The framework FREME[67], allows for the handling of Linked Data (especially NIF), document processing and pipelines generation. Two disadvantages of FREME are that all services must be implemented in JAVA and that it does not allow for an easy integration and orchestration of external HTTP REST API services.

7. Summary and future work

The design, implementation and use of a knowledge graph for the legal domain is an area that has not been sufficiently developed. The Lynx project has studied this area in depth in order to apply it to three different use cases: geothermal energy, labour law and contract analysis. The Legal Knowledge Graph includes information regarding the structure of the documents and the relationships between them, as well as a large amount of linguistic information elaborated for the legal domain, divided into two types: domain independent vocabularies and domain dependent vocabularies (terminologies).

With the LKG, Lynx has filled a crucial resource and technology gap in the legal domain. It has also been able to develop the Lynx Service Platform, specialised on the semantic processing of documents from the legal domain. LynxSP is a complete platform, which, in addition to the semantic processing services, also includes specific characteristics that allow the development of workflows that combine some or all of the semantic processing services (through a Workflow Manager), a storage and management service for documents and linked information (through the Document Manager) and various other components (API Manager, Authentication Module, etc.).

Among the different semantic processing services available in the platform are named entity recognition, entity linking, geographical entity recognition, temporal expression analysis, semantic similarity, question answering, machine translation, summarisation and search. They have all been developed especially with the legal domain in mind, and the combination of all (or some) of these services allows for complex analysis pipelines of legal domain documents.

63 https://gate.ac.uk
64 https://uima.apache.org
65 https://nlp.johnsnowlabs.com
66 https://www.ellogon.org
67 https://freme-project.github.io

List of acronyms

AI Artificial Intelligence

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

BiLSTM Bilateral Long Short Term Memory Neural Networks

CRF Conditional Random Fields

CRUD Create, Read, Update, Delete

CSV Comma-Separated Values

DCM Document Manager

DOCX Office Open XML Document

DS Data Science

EL Entity Linking

ELI European Legislation Identifier

EU European Union

GDPR General Data Protection Regulation

GEO Geolocation

GTE Geothermal Energy

HMM Hidden Markov Model

HTML Hypertext Markup Language

HTTP Hypertext Transfer Protocol

IATE Interactive Terminology for Europe

IDF Inverse Document Frequency

IR Information Retrieval

JPG Joint Photographic Experts Group (graphics file type/extension)

JSON JavaScript Object Notation

KD K Dictionaries

LD Linked Data

LDP Linked Data Platform

LKG Legal Knowledge Graph

LLOD Linguistic Linked Open Data

LynxSP Lynx Service Platform

MeSH Medical Subject Headings

NACE Nomenclature of Economic Activities

NER Named Entity Recognition

NIF NLP Interchange Format

NLM National Library of Medicine

NLP Natural Language Processing

NMT Neural Machine Translation

OWL Web Ontology Language

PDF Portable Document Format

QA Question Answering

QADoc Question Answering System

RDF Resource Description Framework

RelEx Relation Extraction

REST Representational State Transfer

SEAR Search Service

SeSim Semantic Similarity

SHACL Shapes Constraint Language

SKOS Simple Knowledge Organisation System

SPARQL SPARQL Protocol and RDF Query Language

SUMM Summarisation

TBX Termbase Exchange

TF–IDF Term Frequency–Inverse Document Frequency

TimEx Temporal Expression Analysis

TMX Translation Memory Exchange

TSV Tab-Separated Values

URI Uniform Resource Identifier

W3C World Wide Web Consortium

WM Workflow Manager

WSD Word Sense Disambiguation

XLIFF XML Localisation Interchange File Format

XML Extensible Markup Language

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work has been partially funded by the project Lynx, which has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement no. 780602, see http://www.lynx-project.eu.

References

[1] C. Willems, H. M. Nick, Towards optimisation of geothermal heat recovery: An example from the west netherlands basin, Appl. Energy 247 (2019) 582–593, http://dx.doi.org/10.1016/j.apenergy.2019.04.083.

[2] M. Flitsch, Verträge Und Vertragsmanagement in Unternehmen, in: Fachbuch Recht, Linde, 2010.

[3] R. Hamming, Numerical Methods for Scientists and Engineers, in: International series in pure and applied mathematics, McGraw-Hill, 1962.

[4] L. Burnard, S. Bauman (Eds.), TEI P5: Guidelines for electronic text encoding and interchange, Text Encoding Initiative Consortium, 2007.

[5] S. Hellmann, J. Lehmann, S. Auer, NIF: An ontology-based and linked-data-aware NLP Interchange Format, Citeseer, 2012.

[6] J. Bosque-Gil, D. Lonke, J. Gracia, I. Kernerman, Validating the OntoLex-lemon lexicography module with K Dictionaries' multilingual data, in: Electronic Lexicography in the 21st Century. Proceedings of the ELex 2019 Conference, Sintra, Portugal, 2019, pp. 726–746.

[7] D. Lonke, J. Bosque-Gil, Applying the OntoLex-lemon lexicography module to k dictionaries' multilingual data, K Lexical News (28) (2020) 30–36.

[8] J.P. McCrae, J. Bosque-Gil, J. Gracia, P. Buitelaar, P. Cimiano, The OntoLex-lemon model: Development and applications, in: Electronic Lexicography in the 21st Century. Proc. of ELex 2017, Leiden, Netherlands, 2017, pp. 587–597.

[9] J. Gracia, E. Montiel-Ponsoda, D. Vila-Suero, G. Aguado-De-Cea, Enabling language resources to expose translations as linked data on the web, in: Proceedings of the 9th Language Resources and Evaluation Conference, LREC 2014, European Language Resources Association (ELRA), Reykjavik, Iceland, 2014, pp. 409–4013.

[10] J. Bosque-Gil, J. Gracia, E. Montiel-Ponsoda, G. Aguado-de Cea, Modelling multilingual lexicographic resources for the Web of Data: The K Dictionaries case, in: GLOBALEX 2016 Lexicographic Resources for Human Language Technology Workshop Programme, Portorož, Slovenia, 2016, pp. 65–72.

[11] A. Lösch, V. Mapelli, S. Piperidis, A. Vasiljevs, L. Smal, T. Declerck, E. Schnur, K. Choukri, J.V. Genabith, European Language resource coordination: Collecting language resources for public sector multilingual information management, in: N.C.C. chair), K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018.

[12] G. Rehm, M. Berger, E. Elsholz, S. Hegele, F. Kintzel, K. Marheinecke, S. Piperidis, M. Deligiannis, D. Galanis, K. Gkirtzou, P. Labropoulou, K. Bontcheva, D. Jones, I. Roberts, J. Hajic, J. Hamrlová, L. Kačena, K. Choukri, V. Arranz, A. Vasiljevs, O. Anvari, A. Lagzdiņš, J. Meļnika, G. Backfried, E. c Dikici, M. Janosik, K. Prinz, C. Prinz, S. Stampler, D. Thomas-Aniola, J.M.G. Pérez, A.G. Silva, C. Berrío, U. Germann, S. Renals, O. Klejch, European Language grid: An overview, in: Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 3359–3373.

[13] P. Cimiano, J.P. McCrae, V. Rodríguez-Doncel, T. Gornostay, A. Gómez-Pérez, B. Siemoneit, A. Lagzdins, Linked terminologies: applying linked data principles to terminological resources, in: Proceedings of the ELex 2015 Conference, Herstmonceux Castle, United Kingdom, 2015, pp. 504–517.

[14] G. Rehm, J. Moreno-Schneider, J. Gracia, A. Revenko, V. Mireles, M. Khvalchik, I. Kernerman, A. Lagzdins, M. Pinnis, A. Vasilevskis, E. Leitner, J. Milde, P. Weißenhorn, Developing and orchestrating a portfolio of natural legal language processing and document curation services, in: Proceedings of Workshop on Natural Legal Language Processing, NLLP 2019, Minneapolis, USA, Co-located with NAACL 2019. 7 June 2019, 2019.

[15] J. Moreno-Schneider, G. Rehm, E. Montiel-Ponsoda, V. Rodriguez-Doncel, A. Revenko, S. Karampatakis, M. Khvalchik, C. Sageder, J. Gracia, F. Maganza, Orchestrating NLP services for the legal domain, in: Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 2325–2333.

[16] J. Nothman, N. Ringland, W. Radford, T. Murphy, J.R. Curran, Learning multilingual named entity recognition from wikipedia, Artificial Intelligence 194 (2013) 151–175.

[17] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186, http://dx.doi.org/10.18653/v1/N19-1423.

[18] N. Reimers, I. Gurevych, Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 338–348.

[19] E. Leitner, G. Rehm, J. Moreno-Schneider, A dataset of german legal documents for named entity recognition, in: Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 4480–4487.

[20] E. Leitner, G. Rehm, J. Moreno-Schneider, Fine-grained named entity recognition in legal documents, in: Semantic Systems. the Power of AI and Knowledge Graphs. Proceedings of the 15th International Conference, SEMANTiCS 2019, in: Lecture Notes in Computer Science, (11702) Springer, Karlsruhe, Germany, 2019, pp. 272–287, 10/11 September 2019.

[21] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019, CoRR abs/1910.01108.

[22] A. Breit, A. Revenko, K. Rezaee, M.T. Pilehvar, J. Camacho-Collados, WiC-TSV: An evaluation benchmark for target sense verification of words in context, 2020, arXiv:2004.15016.

[23] J. Pustejovsky, K. Lee, H. Bunt, L. Romary, ISO-TimeML: An international standard for semantic annotation., in: Proceedings of the Seventh International Conference on Language Resources and Evaluation, Vol. 10, LREC'10, European Language Resources Association (ELRA), Valletta, Malta, 2010, pp. 394–397.

[24] J. Strötgen, M. Gertz, Multilingual and cross-domain temporal tagging, Lang. Resour. Eval. 47 (2) (2013) 269–298, http://dx.doi.org/10.1007/s10579-012-9179-y.

[25] M. Navas-Loro, V. Rodríguez-Doncel, Annotador: a temporal tagger for Spanish, J. Intell. Fuzzy Syst. 39 (2020) 1979–1991, http://dx.doi.org/10.3233/JIFS-179865, 2.

[26] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Baltimore, Maryland, 2014, pp. 55–60.

[27] A.X. Chang, C.D. Manning, TOKENSREGEX: Defining cascaded regular expressions over tokens, Stanford University Computer Science Technical Reports. CSTR, Department of Computer Science, Stanford University, 2014.

[28] M. Navas-Loro, E. Filtz, V.c. Rodríguez-Doncel, A. Polleres, S. Kirrane, TempCourt: Evaluation of temporal taggers on a new corpus of court decisions, Knowl. Eng. Rev. 34 (2019) e24, http://dx.doi.org/10.1017/S0269888919000195.

[29] A.W. Yu, D. Dohan, M. Luong, R. Zhao, K. Chen, M. Norouzi, Q.V. Le, QAnet: Combining local convolution with global self-attention for reading comprehension, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.

[30] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Spanish pre-trained BERT model and evaluation data, in: Practical ML for Developing Countries Workshop ICLR 2020, Virtual Conference, Formerly Addis Ababa, ETHIOPIA, 2020.

[31] D. Gholipour Ghalandari, Revisiting the centroid-based method: A strong baseline for multi-document summarization, in: Proceedings of the Workshop on New Frontiers in Summarization, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 85–90, http://dx.doi.org/10.18653/v1/W17-4511.

[32] G. Rossiello, P. Basile, G. Semeraro, Centroid-based text summarization through compositionality of word embeddings, in: Proceedings of the MultiLIng 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 12–21, http://dx.doi.org/10.18653/v1/W17-1003.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017.

[34] D. Aksenov, J. Moreno-Schneider, P. Bourgonje, R. Schwarzenberg, L. Hennig, G. Rehm, Abstractive text summarization based on language model conditioning and locality modeling, in: Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020, European Language Resources Association (ELRA), Marseille, France, 2020, pp. 6682–6691.

[35] G. Rehm, K. Marheinecke, S. Hegele, S. Piperidis, K. Bontcheva, J. Hajic, K. Choukri, A. Vasiļjevs, G. Backfried, C. Prinz, J.M.G. Pérez, L. Meertens, P. Lukowicz, J. van Genabith, A. Lösch, P. Slusallek, M. Irgens, P. Gatellier, J. Köhler, L.L. Bars, D. Anastasiou, A. Auksoriutė, N. Bel, A. Branco, G. Budin, W. Daelemans, K.D. Smedt, R. Garabík, M. Gavriilidou, D. Gromann, S. Koeva, S. Krek, C. Krstev, K. Lindén, B. Magnini, J. Odijk, M. Ogrodniczuk, E. Rögnvaldsson, M. Rosner, B. Pedersen, I. Skadina, M. Tadić, D. Tufis, T. Váradi, K. Vider, A. Way, F. Yvon, The European language technology landscape in 2020: Language-centric and human-centric AI for cross-cultural communication in multilingual europe, in: Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), European Language Resources Association (ELRA), Marseille, France, 2020, pp. 3315–3325.

[36] J. Moreno-Schneider, G. Rehm, Curation technologies for the construction and utilisation of legal knowledge graphs, in: Proceedings of the LREC 2018 Workshop on Language Resources and Technologies for the Legal Knowledge Graph, Miyazaki, Japan, 2018, pp. 23–29, 12 May 2018.

[37] T. Agnoloni, G. Venturi, Semantic processing of legal texts, in: J. Visconti (Ed.), Handbook of Communication in the Legal Sphere, De Gruyter, Berlin, Boston, 2018, pp. 109–138.

[38] G. Span, LITES: AN intelligent tutoring system shell for legal education, International Review of Law, Computers & Technology 8 (1) (1994) 103–113.

[39] D. van Kuppevelt, G. van Dijck, Answering legal research questions about dutch case law with network analysis and visualization, in: Legal Knowledge and Information Systems – JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017., in: Frontiers in Artificial Intelligence and Applications, vol. 302, IOS Press, 2017, pp. 95–100, See [44].

[40] T. Agnoloni, L. Bacci, G. Peruginelli, M. van Opijnen, J. van den Oever, M. Palmirani, L. Cervone, O. Bujor, A.A. Lecuona, A.B. García, L.D. Caro, G. Siragusa, Linking European case law: BO-ECLI parser, an open framework for the automatic extraction of legal links, in: Legal Knowledge and Information Systems – JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017, in: Frontiers in Artificial Intelligence and Applications, vol. 302, IOS Press, 2017, pp. 113–118, See [44].

[41] M. Gifford, LexrideLaw: AN argument based legal search engine, in: Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law, ICAIL '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 271–272, http://dx.doi.org/10.1145/3086512.3086548.

[42] J. Bhullar, N. Lam, K. Pham, A. Prabhakaran, A.J. Santillano, Lucem: A Legal Research Tool, (63) Computer Engineering Senior Theses, 2016.

[43] G. Boella, L. di Caro, L. Humphreys, L. Robaldo, L. van der Torre, NLP Challenges for eunomos a tool to build and manage legal knowledge, in: Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC'12, European Language Resources Association (ELRA), Istanbul, Turkey, 2012, pp. 3672–3678.

[44] A.Z. Wyner, G. Casini, Legal Knowledge and Information Systems – JURIX 2017: The Thirtieth Annual Conference, Luxembourg, 13-15 December 2017, Frontiers in Artificial Intelligence and Applications, vol. 302, IOS Press, 2017.