Building the Legal Knowledge Graph for Smart Compliance Services in Multilingual Europe

D3.8 Summarisation and annotation services

PROJECT ACRONYM	Lynx
PROJECT TITLE	Building the Legal Knowledge Graph for Smart Compliance Services in Multilingual Europe
GRANT AGREEMENT	H2020-780602
FUNDING SCHEME	ICT-14-2017 - Innovation Action (IA)
STARTING DATE (DURATION)	01/12/2017 (36 months)
PROJECT WEBSITE	http://lynx-project.eu
COORDINATOR	Elena Montiel-Ponsoda (UPM)
RESPONSIBLE AUTHORS	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI), María Navas-Loro (UPM)
CONTRIBUTORS	
REVIEWERS	Andis Lagzdiņš, Ēriks Ajausks (TILDE), Jorge González-Conejero (UAB)
VERSION | STATUS	V1.0 | Final
NATURE	Other
DISSEMINATION LEVEL	Public
DOCUMENT DOI	10.5281/zenodo.3865668
DATE	31/05/2020 (M30)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780602

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	First TOC version	21/04/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.2	Including content from previous deliverable.	22/04/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.3	Including new content for TIMEX service.	28/04/2020	Maria Navas (UPM)
0.4	Including new content for Annotation Services (NER).	04/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.5	Including content from SUMM service.	08/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.6	Including content from GEO service.	12/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.7	Finishing Conclusions, Annexes and References.	13/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.8	Inclusion of UAB reviews	20/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.9	Inclusion of TILDE reviews	20/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
1.0	Final refinement	27/05/2020	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)

LIST OF ACRONYMS

BERT	Bidirectional Encoder Representations from Transformers
BB	Building Block
BiLSTM	Bilateral LSTM
CRF	Conditional Random Fields
GEO	Geographical Location
HMM	Hidden Markov Model
LKG	Legal Knowledge Graph
LSTM	Long-Short Term Memory
NER	Named Entity Recognition
NLP	Natural Language Processing
RNN	Recurrent Neural Network
Sear	Search
SeSim	Semantic Similarity
Summ	Summarisation
TIMEX	Temporal Expression Analysis
URI	Universal Resource Identifier
WP	Work Package

EXECUTIVE SUMMARY

This report provides the final description of the summarisation and annotation services developed under Task 3.2 in the Lynx project. This report describes several services that are classified into Annotation Services, whose goal is to enrich documents with semantic annotations, and Summarisation Service, which aim at generating a new and shorter piece of a text from one or several longer texts (documents or parts of documents).

The description of the services consists of two parts: for each of the services, first, its general approach is presented, and then, its application in the Lynx project is introduced, putting the focus on datasets used for training new models, rules defined for domain adaptability or generation of dictionaries for specific topics or scenarios.

Although being the final report, the described services are still going to be further developed and evaluated, and will still experience changes and improvements in the following months (and until the end of the project).

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

This report aims to describe the status of summarisation and annotation services as of the end of M30 of the project and fulfils the two main objectives. The first is a reporting effort to better assess (i) the progress of the project, (ii) the final stand of the annotation and conversion services, and (iii) the risk factors in the future usage. The second is the documentation of the services, conventions and functionalities, in order to serve as future references within the project.

The Lynx platform has been defined as a microservice architecture where each service can be designed, implemented and developed independently, even in different programming languages, and then containerized in Docker[1] containers in order to deploy them under the same platform using OpenShift[2] [LynxD11]. In order to allow the usage of the different services by the Curation Workflow Manager (WP4), a REST API interface that manages the communication must be implemented.

1.1 PURPOSE OF THIS DOCUMENT

This report gathers the final status of the annotation and summarisation services in the Lynx project. A description of the services as well as their current implementation, development and deployment status, is provided.

1.2 STRUCTURE OF THIS DOCUMENT

Section 2 gives an overview of the semantic annotation services which aim at annotating and enriching documents within the legal domain. Section 3 gives an overview of the conversion services, especially summarization, which aims at generating summaries from documents. Finally, section 4 describes the future steps of the set of services in general, the individual components, as well as general conclusions of the current status of the implementation.

Technical details in the form of API calls are documented in the appendices.

[1] Docker home page available at: https://www.docker.com/

[2] Openshift home page available at: http://openshift.com

2 ANNOTATION SERVICES

This section describes three annotation services developed in the Lynx project. The goal of these services is the enrichment of the documents to be consumed by the three business cases (see deliverable D4.1 and D4.2 [LynxD41, LynxD42]). First, a general description of the service functionality is provided, and then its application in the Lynx project is explained.

2.1 NAMED ENTITY RECOGNITION

2.1.1 General Description of Method

Named entity recognition is one of the best-known natural language processing tasks. It consists of a system which uses models to annotate named entities. These models are trained by examples annotated with named entities of different types. Generally, the most common types of named entities are PERSON, ORGANIZATION and LOCATION. Using the trained models, the system can annotate (identify) entities that were not present in the training documents.

Many different approaches have been applied for the recognition of named entities depending on the domain and application. In this case, we are describing four different approaches: (i) statistical language model; (ii) BERT based Neural Networks; (iii) Conditional Random Fields (CRF); and (iv) Bilateral Long Short Term Memory Neural Networks (BiLSTM).

2.1.1.1 Statistical Language Model

The recognition of named entities based on statistical language models was implemented using the Name Finder module[3] of OpenNLP, a well-known and established open-source NLP framework developed by Apache. The Name Finder can detect named entities and numbers in text. To be able to detect entities, the Name Finder needs a model. The model is dependent on the language and the entity type it was trained for. To find names in raw text, the text must be segmented into tokens and sentences.

We proceeded with retrieving a unique identifier (URI) for the spotted entities. This component uses the DBpedia SPARQL[4] and DBpedia Spotlight.[5] If a URI is retrieved (the most likely reasons for not retrieving one are either because no Wikipedia or DBpedia entry exists for this particular entity, or our implementation faced a time-out of the SPARQL endpoint), it is stored as part of the entity annotation. In the case of persons and organisations in German, our system points to URIs at Deutsche Nationalbibliothek.[6]

2.1.1.2 BERT based Neural Networks

Neural networks have become of great importance in recent times in all areas of machine learning, especially in natural language processing. Furthermore, in the vast majority of disciplines they have been shown to perform better than previously used systems. This is the case for the recognition of entities, where the approaches and tools that use neural networks do not stop increasing.

This has led us to consider developing a new named entity recognition module based on this technology, especially in BERT [Devlin2018].

[3] https://opennlp.apache.org/docs/1.8.3/apidocs/opennlp-uima/opennlp/uima/namefind/NameFinder.html

[4] https://dbpedia.org/sparql

[5] https://www.dbpedia-spotlight.org/

[6] http://www.dnb.de

To develop our NER system we have based on the work of Kamal Raj[7] and we have adapted it to be able to train new models in the four languages of the project: English, German, Spanish and Dutch.

2.1.1.3 Conditional Random Fields (CRF) method

Conditional Random Fields present a statistical modelling method used for structured prediction. CRFs can be considered as a sequence modelling approach. CRF takes context into account, e.g., the linear chain CRF predicts sequences of labels for sequences of input samples. They are used to encode known relationships between observations and construct consistent interpretations and are often used for labelling or parsing of sequential data. For this approach, a sequence labelling tool, sklearn-crfsuite,[8] is used. A total of 6 models were tested, i.e., three CRF models with coarse- and fine-grained classes. For CRFs, the following groups of features and sources have been selected:

1. F – features for the current word in a window between -2 and 2, which are case and shape features, prefixes, and suffixes.
2. G – gazetteers of persons, countries, cities, streets, landscapes, companies, laws, ordinances and administrative regulations for the current word.
3. L – lookup table for the word similarity, time shifted between -2 and 2, as in [Benikova2015], which contains the four most similar words to the current word.

Overall, three models were designed to chain these three groups of features and gazetteers: (i) CRF-F with features; (ii) CRF-FG with features and gazetteers; and (iii) CRF-FGL with features, gazetteers, and the lookup table. Accordingly, the abbreviations of CRF model names reflect the affected groups.

2.1.1.4 BiLSTM

Long Short Term Memory (LSTMs) networks are capable of learning long-term dependencies. They were introduced in [Hochreiter1997]. LSTM architectures are used when the learning problem is sequential, e.g., if you want to process a line of document. LSTMs and their bidirectional variants (BiLSTM) are popular because they have tried to learn how and when to forget and when not to using gates in their architecture. In previous RNN architectures, vanishing gradients was a big problem and caused those nets not to learn so much. A BiLSTM architecture learns bidirectional long-term dependencies between time steps of time series or sequence data. These dependencies can be useful when you want the network to learn from the complete time series at each time step.

We have applied a sequence labelling tool: UKPLab-BiLSTM [Reimers2017b], in which a total of 6 models were tested, i.e., three BiLSTM models with coarse- and fine-grained classes.

1) BiLSTM-CRF.
2) BiLSTM-CRF + with character embeddings from BiLSTM.
3) BiLSTM-CNN-CRF with character embeddings from CNN.

In the process, such hyper-parameters were used that achieved the best performance in NER according to [Reimers2017a]. The BiLSTM models have two BiLSTM layers, each with a size of 100 units and a dropout of 0.25. The maximum number of epochs is 100. At the same time, the tool uses pre-trained word embeddings for the German language [Reimers2014]. The results were measured with the micro-precision, -recall and -F1 measures. In order to reliably estimate the performance of the models, the evaluation method used is the stratified 10-fold cross-validation. The dataset is mixed sentence-wise and

[7] https://github.com/kamalkraj/BERT-NER

[8] https://sklearn-crfsuite.readthedocs.io

divided into ten mutually exclusive partial sets of similar size. One iteration uses one set for validation and the rest for training. It iterates ten times, so that each part of the dataset is used nine times for training and once for validation. The distribution of named entities in the training and validation set remain the same over the iterations. The cross-validation prevented overfitting during training and the stratification prevented measurement errors in unbalanced data.

2.1.2 Description of Service within Lynx

In the Lynx project, named entity recognition is used for Pilot 1 “Contract Analysis” and Pilot 2 “Oil&Gas – Geothermal Energy” (as named in deliverable 4.3 [LynxD43]) as well as in the Legal Knowledge Graph Population (see deliverable 4.3 [LynxD43]). Currently, the named entity recognition service is composed of both the BERT based Neural Networks and the BiLSTM methods. The statistical language model approach is no longer used because the approach based on BERT provides better results [Raj2020].

2.1.2.1 Language Model

This approach aims to identify more general rather than domain specific entities. Therefore, we trained four different models for the Lynx project using the training data provided by Nothman [Nothman2013]. The four models cover two languages, English and German, and two types of entities, PERSON and ORGANIZATION: (i) English-PER; (ii) English-ORG; (iii) German-PER and German-ORG.

Although we have only generated models for German and English, the Wikiner collection includes also data in other languages such as Spanish, allowing to train models for other languages.

An example of annotated named entities is shown in Table 1.

```
@prefix dbo:   <http://dbpedia.org/ontology/> .
@prefix eli:   <http://data.europa.eu/eli/ontology#> .
@prefix dct:   <http://purl.org/dc/terms/> .
@prefix dbr:   <http://dbpedia.org/resource/> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix lkg:   <http://lkg.lynx-project.eu/def/> .
@prefix skos:  <http://www.w3.org/2004/02/skos/core#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix nif:   <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .

<http://lkg.lynx-project.eu/res/3f515f41#offset_74_83>
    a                 nif:OffsetBasedString , lkg:LynxAnnotation ;
    nif:anchorOf        "Microsoft" ;
    nif:annotationUnit    [ a                 nif:AnnotationUnit ;
                        itsrdf:taAnnotatorsRef  <http://dfki.lynx-project.eu/annotators/NER> ;
                        itsrdf:taClassRef       dbo:Organisation
                      ] ;
    nif:beginIndex        "74"^^xsd:nonNegativeInteger ;
    nif:endIndex        "83"^^xsd:nonNegativeInteger ;
    nif:referenceContext  <http://lkg.lynx-project.eu/res/3f515f41> .

<http://lkg.lynx-project.eu/res/3f515f41#offset_87_93>
    a                 nif:OffsetBasedString , lkg:LynxAnnotation ;
    nif:anchorOf        "Google" ;
    nif:annotationUnit    [ a                 nif:AnnotationUnit ;
                        itsrdf:taAnnotatorsRef  <http://dfki.lynx-project.eu/annotators/NER> ;
                        itsrdf:taClassRef       dbo:Organisation
```

```
                    ] ;
    nif:beginIndex      "87"^^xsd:nonNegativeInteger ;
    nif:endIndex        "93"^^xsd:nonNegativeInteger ;
    nif:referenceContext  <http://lkg.lynx-project.eu/res/3f515f41> .

<http://lkg.lynx-project.eu/res/3f515f41>
    a            lkg:LynxDocument , nif:Context ;
    lkg:metadata    [ eli:id_local  "3f515f41" ;
                    dct:language  "en"
                  ] ;
    nif:beginIndex  "0"^^xsd:nonNegativeInteger ;
    nif:endIndex    "171"^^xsd:nonNegativeInteger ;
    nif:isString    "The text to be processed is this one, and probably it can be annotated by Microsoft or Google and
distributed to a company located in Berlin, which benefits Salvador Dali." .
```

Table 1. Example of annotated named entities in NIF format (Turtle format)

2.1.2.2 BERT based Neural Networks

This approach also aims to identify general entities. It is based on the BERT neural networks, and similarly to the statistical language model, it has been trained using the training data WikiNer (provided by Nothman [Nothman2013]). In this case we do not have to train different models for each entity type (Person, Location, Organization) but all are recognized 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'*). The models recognize four types of entities: PERSON, ORGANIZATION, LOCATION (although these entities are also recognized by the Geolocation Service) and MISCELLANEA.

In Table 2 an example of annotated entities for each language is shown.

English	PREFIXES[9] **<http://lynx-project.eu/res/c2dc6568-8ed1-46d5-946d-b759af404466#offset_11_17>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;** **ns1:anchorOf "Berlin" ;** **ns1:annotationUnit [** **a ns1:AnnotationUnit ;** **ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns2:taClassRef <http://dbpedia.org/ontology/Miscellaneous>** **] ;** **ns1:beginIndex "11" ^^xsd:nonNegativeInteger ;** **ns1:endIndex "17"^^xsd:nonNegativeInteger ;** **ns1:referenceContext <http://lynx-project.eu/res/c2dc6568-8ed1-46d5-946d-b759af404466> .** **<http://lynx-project.eu/res/c2dc6568-8ed1-46d5-946d-b759af404466#offset_71_77>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;** **ns1:anchorOf "Merkel" ;** **ns1:annotationUnit [** **a ns1:AnnotationUnit ;** **ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns2:taClassRef <http://dbpedia.org/ontology/Person>** **] ;**

[9] The prefix section has not been included in the examples to avoid repeating content. This part in all examples is:

@prefix ns1: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix ns2: <http://www.w3.org/2005/11/its/rdf#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

	ns1:beginIndex "71"^^xsd:nonNegativeInteger ; ns1:endIndex "77" ^^xsd:nonNegativeInteger ; ns1:referenceContext <http://lynx-project.eu/res/c2dc6568-8ed1-46d5-946d-b759af404466> . <http://lynx-project.eu/res/c2dc6568-8ed1-46d5-946d-b759af404466> a <http://lkg.lynx-project.eu/def/LynxDocument>, ns1:Context ; ns1:beginIndex "0" ^^xsd:nonNegativeInteger ; ns1:endIndex "100" ^^xsd:nonNegativeInteger ; ns1:isString """Welcome to Berlin on 2019 and 2020 with the help of Microsoft. Angela Merkel was in Berlin in 2016.""" .
German	PREFIXES[7] **<http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f#offset_100_106>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;** **ns1:anchorOf "Berlin" ;** **ns1:annotationUnit [** **a ns1:AnnotationUnit ;** **ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns2:taClassRef <http://dbpedia.org/ontology/Location>** **] ;** **ns1:beginIndex "100" ^^xsd:nonNegativeInteger ;** **ns1:endIndex "106" ^^xsd:nonNegativeInteger ;** **ns1:referenceContext <http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f> .** **<http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f#offset_14_20>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;** **ns1:anchorOf "Berlin" ;** **ns1:annotationUnit [** **a ns1:AnnotationUnit ;** **ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns2:taClassRef <http://dbpedia.org/ontology/Location>** **] ;** **ns1:beginIndex "14" ^^xsd:nonNegativeInteger ;** **ns1:endIndex "20" ^^xsd:nonNegativeInteger ;** **ns1:referenceContext <http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f> .** **<http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f#offset_63_72>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;** **ns1:anchorOf "Microsoft" ;** **ns1:annotationUnit [** **a ns1:AnnotationUnit ;** **ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns2:taClassRef <http://dbpedia.org/ontology/Organisation>** **] ;** **ns1:beginIndex "63" ^^xsd:nonNegativeInteger ;** **ns1:endIndex "72" ^^xsd:nonNegativeInteger ;** **ns1:referenceContext <http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f> .** <http://lynx-project.eu/res/2fd7757c-0ec3-4ab7-b061-ec95ab48fe1f> a <http://lkg.lynx-project.eu/def/LynxDocument>, ns1:Context ; ns1:beginIndex "0" ^^xsd:nonNegativeInteger ; ns1:endIndex "107" ^^xsd:nonNegativeInteger ; ns1:isString "Willkommen in Berlin in den Jahren 2019 und 2020 mit Hilfe von Microsoft. Angela Merkel war 2016 in Berlin."
Spanish	PREFIXES[7] **<http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3#offset_13_19>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;** **ns2:anchorOf "Berlín" ;** **ns2:annotationUnit [** **a ns2:AnnotationUnit ;** **ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns1:taClassRef <http://dbpedia.org/ontology/Location>** **] ;** **ns2:beginIndex "13"^^xsd:nonNegativeInteger ;** **ns2:endIndex "19" ^^xsd:nonNegativeInteger ;**

	ns2:referenceContext <http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3> . **<http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3#offset_51_60>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;** **ns2:anchorOf "Microsoft" ;** **ns2:annotationUnit [** **a ns2:AnnotationUnit ;** **ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns1:taClassRef <http://dbpedia.org/ontology/Organisation>** **] ;** **ns2:beginIndex "51" ^^xsd:nonNegativeInteger ;** **ns2:endIndex "60" ^^xsd:nonNegativeInteger ;** **ns2:referenceContext <http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3> .** **<http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3#offset_62_75>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;** **ns2:anchorOf "Angela Merkel" ;** **ns2:annotationUnit [** **a ns2:AnnotationUnit ;** **ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns1:taClassRef <http://dbpedia.org/ontology/Person>** **] ;** **ns2:beginIndex "62" ^^xsd:nonNegativeInteger ;** **ns2:endIndex "75" ^^xsd:nonNegativeInteger ;** **ns2:referenceContext <http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3> .** **<http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3#offset_86_92>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;** **ns2:anchorOf "Berlín" ;** **ns2:annotationUnit [** **a ns2:AnnotationUnit ;** **ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns1:taClassRef <http://dbpedia.org/ontology/Location>** **] ;** **ns2:beginIndex "86"^^xsd:nonNegativeInteger ;** **ns2:endIndex "92" ^^xsd:nonNegativeInteger ;** **ns2:referenceContext <http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3> .** <http://lynx-project.eu/res/d5007e94-cd43-4bea-a9d5-ca1ababbaaf3> a <http://lkg.lynx-project.eu/def/LynxDocument>, ns2:Context ; ns2:beginIndex "0" ^^xsd:nonNegativeInteger ; ns2:endIndex "101" ^^xsd:nonNegativeInteger ; ns2:isString "Bienvenido a Berlín en 2019 y 2020 con la ayuda de Microsoft. Angela Merkel estuvo en Berlín en 2016." .
Dutch	PREFIXES[7] **<http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29#offset_0_6>** **a ns2:Annotation, ns2:OffsetBasedString;** **ns2:anchorOf "Welkom" ;** **ns2:annotationUnit [** **a ns2:AnnotationUnit ;** **ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns1:taClassRef <http://dbpedia.org/ontology/Location>** **] ;** **ns2:beginIndex "0"^^xsd:nonNegativeInteger ;** **ns2:endIndex "6" ^^xsd:nonNegativeInteger ;** **ns2:referenceContext <http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29> .** **<http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29#offset_11_18>** **a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;** **ns2:anchorOf "Berlijn" ;** **ns2:annotationUnit [** **a ns2:AnnotationUnit ;** **ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;** **ns1:taClassRef <http://dbpedia.org/ontology/Location>** **] ;**

```
    ns2:beginIndex "11"^^xsd:nonNegativeInteger ;
    ns2:endIndex "18" ^^xsd:nonNegativeInteger ;
    ns2:referenceContext <http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29> .
<http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29#offset_51_60>
    a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;
    ns2:anchorOf "Microsoft" ;
    ns2:annotationUnit [
        a ns2:AnnotationUnit ;
        ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;
        ns1:taClassRef <http://dbpedia.org/ontology/Miscellaneous>
    ] ;
    ns2:beginIndex "51" ^^xsd:nonNegativeInteger ;
    ns2:endIndex "60" ^^xsd:nonNegativeInteger ;
    ns2:referenceContext <http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29> .
<http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29#offset_62_75>
    a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;
    ns2:anchorOf "Angela Merkel" ;
    ns2:annotationUnit [
        a ns2:AnnotationUnit ;
        ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;
        ns1:taClassRef <http://dbpedia.org/ontology/Person>
     ] ;
    ns2:beginIndex "62" ^^xsd:nonNegativeInteger ;
    ns2:endIndex "75" ^^xsd:nonNegativeInteger ;
    ns2:referenceContext <http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29> .
<http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29#offset_91_98>
    a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns2:Annotation, ns2:OffsetBasedString ;
    ns2:anchorOf "Berlijn" ;
    ns2:annotationUnit [
        a ns2:AnnotationUnit ;
        ns1:taAnnotatorsRef <http://lkg.lynx-project.eu/def/NER> ;
        ns1:taClassRef <http://dbpedia.org/ontology/Location>
    ] ;
    ns2:beginIndex "91"^^xsd:nonNegativeInteger ;
    ns2:endIndex "98" ^^xsd:nonNegativeInteger ;
    ns2:referenceContext <http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29> .
<http://lynx-project.eu/res/69b4f92e-4f73-4590-be96-91696ff15d29>
    a <http://lkg.lynx-project.eu/def/LynxDocument>, ns2:Context ;
    ns2:beginIndex "0" ^^xsd:nonNegativeInteger ;
    ns2:endIndex "99" ^^xsd:nonNegativeInteger ;
    ns2:isString "Welkom bij Berlijn in 2019 en 2020 met de hulp van Microsoft. Angela Merkel was in 2016 in
    Berlijn." .
```

Table 2. Example of annotated named entities (Turtle format) for each language (English, German, Spanish and Dutch)

2.1.2.3 CRF and BiLSTM

In order to adapt the CRF and BiLSTM approaches to the needs of the Lynx project, i.e., the legal domain, we developed a dataset containing annotated Legal Entities. This dataset, German Legal Documents for Named Entity Recognition, consists of 750 German court decisions published on the portal "Rechtsprechung im Internet".[10] The source text was collected from the XML documents, split into sentences and words by SoMaJo [Proisl2016] and annotated manually in WebAnno [Eckart2016]. The

[10] http://www.rechtsprechung-im-internet.de

dataset[11] is freely available for download under CC BY 4.0 license.[12] The data is released in CoNLL-2002 format. A deeper description of the dataset and the adaptation process of the CRF and BiLSTM can be found in [Leitner2019]. The whole list of entity types can be seen in Figure 1. Besides showing the different types of entities, the table describes the annotations included in the training set used for training the CRF and BiLSTM methods.

Coarse-grained classes			#	%	Fine-grained classes			#	%
1	**PER**	Person	3,377	6.30	1	**PER**	Person	1,747	3.26
					2	**RR**	Judge	1,519	2.83
2	**LOC**	Location	2,468	4.60	3	**AN**	Lawyer	111	0.21
					4	**LD**	Country	1,429	2.66
					5	**ST**	City	705	1.31
					6	**STR**	Street	136	0.25
					7	**LDS**	Landscape	198	0.37
3	**ORG**	Organization	7,915	14.76	8	**ORG**	Organization	1,166	2.17
					9	**UN**	Company	1,058	1.97
					10	**INN**	Institution	2,196	4.09
					11	**GRT**	Court	3,212	5.99
					12	**MRK**	Brand	283	0.53
4	**NRM**	Legal norm	20,816	38.81	13	**GS**	Law	18,520	34.53
					14	**VO**	Ordinance	797	1.49
					15	**EUN**	European legal norm	1,499	2.79
5	**REG**	Case-by-case regulation	3,470	6.47	16	**VS**	Regulation	607	1.13
					17	**VT**	Contract	2,863	5.34
6	**RS**	Court decision	12,580	23.46	18	**RS**	Court decision	12,580	23.46
7	**LIT**	Legal literature	3,006	5.60	19	**LIT**	Legal literature	3,006	5.60
							Total	53,632	100

Figure 1. Named entities available in the LER dataset

In Table 2 an example of annotated entities for each language is shown.

```
@prefix ns1: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix ns2: <http://www.w3.org/2005/11/its/rdf#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07#offset_13_19>
    a ns1:Annotation, ns1:OffsetBasedString ;
    ns1:anchorOf "Berlin" ;
    ns1:annotationUnit [
        a ns1:AnnotationUnit ;
        ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/res/LER> ;
        ns2:taClassRef <http://dbpedia.org/ontology/ST>
    ] ;
    ns1:beginIndex "13"^^xsd:nonNegativeInteger ;
    ns1:endIndex "19"^^xsd:nonNegativeInteger ;
    ns1:referenceContext <http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07> .

<http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07#offset_32_41>
```

[11] https://github.com/elenanereiss/Legal-Entity-Recognition

[12] https://creativecommons.org/licenses/by/4.0/deed.en

```
    a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;
    ns1:anchorOf "Microsoft" ;
    ns1:annotationUnit [
        a ns1:AnnotationUnit ;
        ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/res/LER> ;
        ns2:taClassRef <http://dbpedia.org/ontology/UN>
    ] ;
    ns1:beginIndex "32"^^xsd:nonNegativeInteger ;
    ns1:endIndex "41"^^xsd:nonNegativeInteger ;
    ns1:referenceContext <http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07> .

<http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07#offset_58_70>
    a <http://lkg.lynx-project.eu/def/LynxAnnotation>, ns1:Annotation, ns1:OffsetBasedString ;
    ns1:anchorOf "Herr Schmidt";
    ns1:annotationUnit [
        a ns1:AnnotationUnit ;
        ns2:taAnnotatorsRef <http://lkg.lynx-project.eu/res/LER> ;
        ns2:taClassRef <http://dbpedia.org/ontology/RR>
    ] ;
    ns1:beginIndex "58" ^^xsd:nonNegativeInteger ;
    ns1:endIndex "70"^^xsd:nonNegativeInteger ;
    ns1:referenceContext <http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07> .

<http://lynx-project.eu/res/c0b89d21-1d5b-4c19-9189-358461781d07>
    a <http://lkg.lynx-project.eu/def/LynxDocument>, ns1:Context ;
    ns1:beginIndex "0"^^xsd:nonNegativeInteger ;
    ns1:endIndex "70" ^^xsd:nonNegativeInteger ;
    ns1:isString "Wilkommen in Berlin am 2018 bei Microsoft mit den Richter Herr Schmidt" .
```

Table 3. Example of annotated legal named entities (Turtle format)

2.2 TEMPORAL EXPRESSION ANALYSIS

2.2.1 General Description of Method

The Temporal Expression Extraction service is responsible for the identification and normalization of temporal expressions, including any word or sequence of words referring to a time instant (e.g., ‘five o’clock’) or a time interval (e.g., ‘from nine to ten’). Temporal expressions frame events or happenings implicitly or explicitly mentioned in the document. Following the ISO-TimeML standard [Pustejovsky2010] we distinguish among dates, times, durations and sets. In addition, intervals are currently being implemented, but for now just for the Spanish language. Once properly tested, they will be added to the service and extended at least to English.

- DATE: Calendar expressions such as 'October 7, 1991', '22/01/2018', or '1992'; also relative expressions like 'Two days ago'.
- TIME: Points in time ('At seven o'clock', '22:30', '3.30pm'...), absolute or relative ('Half an hour ago', 'In two minutes and three seconds').
- DURATION: Amounts of time like 'Two days', 'Three years and six months', 'Two centuries', 'One hour and 20 minutes' or 'Half an hour'.
- SET: Repetitions in time (such as 'Monthly', 'Twice a week', 'Every Monday', 'Three times a year', 'Every first of the month'...).
- INTERVAL: Period between two temporal expressions (‘from 14h to 20h’, ‘from Monday to Friday’…).

The service is rule-based, and is 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, Dutch and Italian use default functionality of a third-party library, HeidelTime [Strötgen2010].

2.2.2 Description of Service within Lynx

The Temporal Expression Extraction service accepts both NIF and plain text POST requests and returns the annotations in NIF format or as TIMEX3 tags. Just the input text, its language and an optional reference date are needed. If the reference date (usually the creation date of the text to analyse) is not provided, the current date is used. The tables below show the output of the service for the following example sentence, using as reference date "2020-04-27":

*The trial will begin **tomorrow** and will last **two days and three hours**.*
*There will be reports **twice an hour**.*

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix lkg: <http://lkg.lynx-project.eu/def/> .

<URL>
	a		lkg:LynxDocument, nif:Context ;
	nif:isString	"The trial will begin tomorrow and will last two days and three hours. There will be reports twice an hour.
"@en .

<URL#offset_21_29>
	a		nif:OffsetBasedString, lkg:LynxAnnotation ;
	nif:referenceContext	<URL> ;
	nif:anchorOf	"tomorrow" ;
	nif:beginIndex	"21"^^xsd:nonNegativeInteger ;
	nif:endIndex	"29"^^xsd:nonNegativeInteger ;
	nif:annotationUnit [
		a nif:AnnotationUnit ;
		itsrdf:taAnnotatorsRef <http://annotador.oeg-upm.net/> ;
		itsrdf:taClassRef lkg:DATE;
		itsrdf:taConfidence 0.9 ;
		rdf:value "2020-04-28"
	] .

<URL#offset_44_68>
	a		nif:OffsetBasedString, lkg:LynxAnnotation ;
	nif:referenceContext	<URL> ;
	nif:anchorOf	"two days and three hours" ;
	nif:beginIndex	"44"^^xsd:nonNegativeInteger ;
	nif:endIndex	"68"^^xsd:nonNegativeInteger ;
	nif:annotationUnit [
		a nif:AnnotationUnit ;
		itsrdf:taAnnotatorsRef <http://annotador.oeg-upm.net/> ;
		itsrdf:taClassRef lkg:DURATION;
		itsrdf:taConfidence 0.9 ;
		rdf:value "P2DT3H"
	] .

<URL#offset_94_107>
	a		nif:OffsetBasedString, lkg:LynxAnnotation ;
	nif:referenceContext	<URL> ;
	nif:anchorOf	"twice an hour" ;
	nif:beginIndex	"94"^^xsd:nonNegativeInteger ;
	nif:endIndex	"107"^^xsd:nonNegativeInteger ;
	nif:annotationUnit [
```

```
        a nif:AnnotationUnit ;
        itsrdf:taAnnotatorsRef <http://annotador.oeg-upm.net/> ;
        itsrdf:taClassRef lkg:SET;
        itsrdf:taConfidence 0.9 ;
        rdf:value "PT1H"
    ] .
```

Table 4. NIF output of the example sentence

The trial will begin <**TIMEX3** tid="t1" type="DATE" value="2019-02-15">tomorrow</**TIMEX3**> and will last <**TIMEX3** tid="t2" type="DURATION" value="PT2D3H">two days and three hours</**TIMEX3**>. There will be reports <**TIMEX3** tid="t3" type="SET" value="1H" freq="2X">twice an hour</**TIMEX3**>

Table 5. Output of the example sentence with TIMEX3 tags

Spanish and English Temporal Expression Analysis service

This service works on the CoreNLP library, using its tokenizer, sentence splitter, POS tagging, lemmatizer, NER (excluding the SUTime service) and the TokensRegex[13], in charge of managing the rules to detect temporal expressions. For English, the default POS and lemmatizer services were used, for Spanish, the IxaPipes [Agerri2014] service was injected.

Regarding the rules, a set of around 100 rules for each language were developed. Each of them is activated iteratively with different priorities and in different stages of the processing, and targets different temporal expressions. The rules identify them and provide the information needed for normalizing them afterwards. They also 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 date in bold is part of a reference to a legal document, not a date referring to the narrative of the text) and the wrong normalization it implies for the surrounding temporal expressions (for instance, if in the previous example we had considered "5 April 1993" as a temporal expression, any surrounding anchored expression such as "the following month" would be considered by most taggers as anchored to it and therefore referring to May 1993). To learn how the service should deal with these kinds of particularities, a Temporal Expressions usage Questionnaire (see Annex 2) was sent to the industrial partners of each of the pilots in order to gather temporal specific needs for each of the use cases. In addition, calls were conducted when needed to clarify the requirements, and feedback from experts in Spanish was collected in order to decide how to deal with them. As a result, a new version of the rules for Spanish is currently under development, and once completed, will be integrated into the service. The lessons learnt will also be used to improve also the set of rules for English. Some examples of the new contributions of the rules, and the feedback received from the result of the Questionnaire, are the following:

- Addition of INTERVALs, not present in the standard used. An example of the standard output (current rule set) and the improved extended annotation (rules under construction) on the sentence "Entre los días 23 y 27 de mayo." ("Between the 23rd and the 27th of May") can be found in Table 6 and Table 7.
- Need of identification of modifiers like „no inferior a" (at least), or specific legal anchor dates, such as „fecha de conciliación" (date of the conciliation).
- Interest on preserving dates included in legal references.

Entre <**TIMEX3** tid="t1" type="DATE" value="2020-05-23">los días 23</**TIMEX3**> y <**TIMEX3** tid="t2" type="DATE" value="2020-05-27">27 de mayo</**TIMEX3**>.

[13] https://nlp.stanford.edu/software/tokensregex.html

Table 6. Standard output

```
<INTERVAL iid="i1" tidBegin="t1" typeBegin="DATE" valueBegin="2020-05-23" tidEnd="t2" typeEnd="DATE"
valueEnd="2020-05-27">Entre los días 23 y 27 de mayo</INTERVAL>.
```

Table 7. Improved rules output

Regarding the evaluation of the service, for English we used the TempCourt corpus [Navas-Loro2019], while for Spanish we used the TempEval2[14] dataset, both publicly available. While the latter is generic and widely used in the temporal tagging community, the TempCourt corpus comprehends several judgments from different courts that includes specific legal temporal annotations. In addition, for Spanish, despite of being no available legal corpora to test the service (the lack of expert annotators prevented the materialization of the ongoing corpus mentioned in D3.3), the result of the annotation using the new rules is currently being manually reviewed by legal experts.

The results of the current set of rules are shown below. We consider for this evaluation three features: the **extent** of the tag, the correctness of the normalized **value** and, optionally, the **type** classification if reported for other state-of-the-art systems. To obtain the measures Precision (P), Recall (R) and F-value (F1), both in lenient and strict mode, we used the software GATE[15].

TempCourt corpus (English Evaluation)

TempCourt corpus[16] is a set of thirty court decisions gathered from the European Court of Justice (ECJ), the European Court of Human Rights (ECHR) and the United States Supreme Court (USSC). In the evaluation we just consider the European sources, this is, the first two courts. The documents were annotated using two different criteria, and producing therefore two different annotation sets:

- StandardTimeML - This set of annotations follows the previously mentioned TimeML guidelines, annotating all the Temporal Expressions found.
- LegalTimeML - This set of annotations avoids some of the standard annotations, such as for instance mentions to years when citing legal articles. Even when they are temporal references, they are not temporal references in the narrative of the text.

We compared the output of the tool integrated in our service, called Annotador[17], against these two annotation sets for each of the two European sources of documents. We split the results on the source of the documents (10 from each court), as done in the reference paper, in order to make easier the task of comparing the results reported in it and the ones of the service. Tables (Table 8 and Table 9) present the results of our service, while Figures (Figure 2 and Figure 3) the results reported in [Navas-Loro2019] for other temporal taggers.

<u>ECHR</u>

	Lenient			Strict			Lenient+value			Strict+value		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
StandardTimeML	0.9746	0.9583	0.9664	0.9407	0.9250	**0.9328**	**0.9068**	**0.8917**	**0.8992**	**0.8814**	**0.8667**	**0.8739**
LegalTimeML	0.8729	0.9717	0.9196	**0.8305**	**0.9245**	**0.8750**	**0.8051**	**0.8962**	**0.8482**	**0.7712**	**0.8585**	**0.8125**

[14] http://semeval2.fbk.eu/semeval2.php?location=tasks&taskid=5

[15] https://gate.ac.uk/

[16] https://tempcourt.github.io/TempCourt/

[17] http://annotador.oeg-upm.net/

Table 8. Result of Annotador, the tool used in our service, in the ECHR documents in TempCourt.

	Lenient			Strict			Lenient + value			Strict + value		
A	P	R	F1	P	R	F1	P	R	F1	P	R	F1
HE	**0.99**	**0.99**	**0.99**	0.84	0.84	0.84	0.78	0.78	0.78	0.78	0.78	0.78
	0.88	**0.99**	**0.93**	0.71	0.80	0.75	0.67	0.75	0.71	0.64	0.72	0.68
SU	0.88	0.87	0.88	0.85	0.84	0.84	0.78	0.78	0.78	0.76	0.75	0.75
	0.76	0.85	0.80	0.71	0.80	0.76	0.66	0.74	0.79	0.64	0.72	0.68
GU	0.96	0.93	0.94	**0.95**	0.92	**0.93**	**0.86**	0.84	0.85	**0.86**	0.84	**0.85**
	0.84	0.92	0.88	**0.83**	**0.92**	**0.87**	0.74	0.82	0.78	**0.74**	0.82	**0.78**
CA	0.88	0.87	0.87	0.83	0.82	0.82	0.78	0.78	0.78	0.75	0.75	0.75
	0.75	0.85	0.80	0.70	0.79	0.74	0.65	0.74	0.69	0.64	0.72	0.67
CL	0.92	0.78	0.85	0.34	0.32	0.35	–	–	–	–	–	–
	0.80	0.77	0.78	0.33	0.32	0.33	–	–	–	–	–	–
SY	0.98	0.93	0.96	0.83	0.79	0.81	0	0	0	0	0	0
	0.86	0.93	0.90	0.70	0.76	0.73	0	0	0	0	0	0
TE	0.94	0.95	0.95	0.92	**0.93**	0.92	**0.86**	**0.88**	**0.87**	0.85	**0.86**	**0.85**
	0.83	0.95	0.89	0.80	**0.92**	0.85	**0.75**	**0.86**	**0.80**	0.72	**0.83**	0.77
TI	0.78	0.85	0.81	0.64	0.70	0.67	0.64	0.71	0.67	0.63	0.69	0.66
	0.69	0.86	0.76	0.62	0.77	0.69	0.64	0.79	0.71	0.61	0.76	0.68
US	0.73	0.61	0.67	0.69	0.58	0.63	0	0	0	0	0	0
	0.65	0.62	0.64	0.61	0.58	0.60	0	0	0	0	0	0
UW	0.90	0.53	0.67	0.51	0.30	0.38	0.55	0.33	0.41	0.51	0.30	0.38
	0.86	0.58	0.69	0.48	0.32	0.38	0.51	0.34	0.41	0.48	0.32	0.38

Figure 2. Result of different temporal taggers in ECHR documents in the TempCourt corpus, as reported in [Navas-Loro2019]. The white row represents the results against the StandardTimeML annotation set, while the grey one does so for the LegalTimeML annotation set. The temporal taggers evaluated were, in order of appearance in the table: Heideltime (HE), Sutime (SU), GUTime/TARSQI (GU), CAEVO (CA), ClearTK (CL), Syntime (SY), TERNIP (TE), TIPSem (TI), USFD2 (US) and UWTime (UW).

ECJ

	Lenient			Strict			Lenient+value			Strict+value		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
StandardTimeML	**0.9793**	0.9403	**0.9594**	0.9556	0.9176	0.9362	0.9586	0.9205	**0.9391**	0.9408	0.9034	**0.9217**
LegalTimeML	**0.5414**	0.9433	**0.6880**	0.5237	0.9124	**0.6654**	**0.5296**	0.9227	**0.6729**	0.5118	0.8918	**0.6504**

Table 9. Result of Annotador, the tool in our service, in the ECJ documents in TempCourt.

	Lenient			Strict			Lenient + value			Strict + value		
A	P	R	F1	P	R	F1	P	R	F1	P	R	F1
HE	0.48	0.95	0.64	0.47	0.94	0.63	0.47	**0.94**	0.62	0.47	**0.93**	0.62
	0.27	0.97	0.42	0.26	**0.96**	0.41	0.26	**0.94**	0.40	0.26	**0.93**	0.40
SU	0.81	0.97	0.88	0.79	**0.95**	0.86	0.78	0.93	0.85	0.77	0.92	0.84
	0.44	0.95	0.60	0.43	0.93	0.58	0.41	0.90	0.57	0.41	0.89	0.56
GU	**0.97**	0.87	0.91	**0.97**	0.86	**0.91**	0.94	0.84	0.89	0.94	0.84	0.88
	0.51	0.82	0.63	0.50	0.82	0.62	0.48	0.78	0.60	0.48	0.78	0.60
CA	0.89	0.74	0.81	0.85	0.70	0.77	0.86	0.71	0.77	0.85	0.70	0.77
	0.49	0.74	0.59	0.46	0.70	0.56	0.46	0.70	0.56	0.46	0.69	0.55
CL	0.77	0.88	0.82	0.32	0.36	0.34	–	–	–	–	–	–
	0.42	0.88	0.57	0.18	0.37	0.24	–	–	–	–	–	–
SY	0.89	**0.99**	**0.93**	0.81	0.90	0.85	0	0	0	0	0	0
	0.49	**0.98**	0.65	0.46	0.92	0.61	0	0	0	0	0	0
TE	**0.97**	0.88	0.92	0.96	0.88	**0.91**	**0.96**	0.87	**0.91**	**0.95**	0.87	**0.91**
	0.54	0.89	**0.67**	**0.53**	0.88	**0.66**	**0.53**	0.88	**0.65**	**0.52**	0.87	**0.65**
TI	0.72	0.81	0.76	0.64	0.72	0.68	0.62	0.70	0.65	0.61	0.69	0.65
	0.41	0.83	0.54	0.37	0.75	0.49	0.35	0.71	0.47	0.34	0.70	0.46
US	0.31	0.54	0.39	0.29	0.51	0.37	0.02	0.04	0.03	0.02	0.03	0.02
	0.20	0.65	0.31	0.19	0.61	0.29	0.02	0.06	0.03	0.02	0.05	0.02
UW	–	–	–	–	–	–	–	–	–	–	–	–
	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3. Result of different temporal taggers in ECJ documents in the TempCourt corpus, as reported in [Navas-Loro2019]. See Fig. 7 for more information.

TempEval2 corpus (Spanish Evaluation)

TempEval2 was a challenge held during SemEval 2010 where a corpus of 35 news was annotated by experts in order to evaluate the capabilities of the state-of-the-art temporal taggers. We present below the results of the tool used by our service (Annotador) to find temporal expressions in Spanish, against the two main currently operative state-of-the-art temporal taggers that can process Spanish: HeidelTime and SUTime[18].

[18] It must be taken into account that SUTime has rules for Spanish, but they are not complete and do not cover all the types of temporal expressions. This is why its results show very good precision but low recall.

		strict			lenient			average		
Tagger	Attribute	P	R	F1	P	R	F1	P	R	F1
Annotador	value	0.8021	**0.7778**	0.7897	0.8281	**0.8030**	0.8154	0.8151	**0.7904**	0.8026
	type	0.8438	**0.8182**	**0.8308**	0.9063	**0.8788**	**0.8923**	**0.8750**	**0.8485**	**0.8615**
	extent	0.8646	**0.8384**	0.8513	0.9323	**0.9040**	**0.9179**	0.8984	**0.8712**	**0.8846**
Heidel	value	**0.8418**	0.7525	**0.7947**	**0.8644**	0.7727	**0.8160**	**0.8531**	0.7626	**0.8053**
	type	**0.8531**	0.7626	0.8053	0.8870	0.7929	0.8373	0.8701	0.7778	0.8213
	extent	**0.9040**	0.8081	**0.8533**	0.9435	0.8434	0.8907	**0.9237**	0.8258	0.8720
SUTime	value	0.6377	0.2222	0.3296	0.8261	0.2879	0.4270	0.7319	0.2551	0.3783
	type	0.6522	0.2273	0.3371	**0.9275**	0.3232	0.4794	0.7899	0.2753	0.4082
	extent	0.6667	0.2323	0.3446	**0.9565**	0.3333	0.4944	0.8116	0.2828	0.4195

Figure 4. Result of state-of-the-art temporal taggers for Spanish in the TempEval2 dataset, as reported in [Navas-Loro2020]. Annotador is the one used by our service.

More detailed information about this evaluation, as well as about the inner work of the Spanish rules, can be found in a recent publication [Navas-Loro2020].

German Temporal Expression Analysis service

The aim of this service is a good automatic recognition and semantic interpretation (normalization) of temporal expressions in German-language legal texts, especially court decisions and legislative texts. The definition of temporal expressions includes dates such as “1. Januar 2000” (1st January, 2000), as well as durations like “fünf Kalenderjahre” (five calendar years) and repeating time intervals like “jeden Monat” (every month). Such expressions should not only be identified, but also normalized by translating them into a standardized ISO format. Since no suitable corpus exists yet, a small text collection is annotated with temporal expressions using the timex3 tag according to the TimeML standard.

One of the specifics of the domain are references to other legal texts which contain (alleged) dates (Richtlinie 2008 / 96 /EG, Directive 2008 / 96 /EG). Other peculiarities of the domain and/or language are frequent use of compounds such as “Kalenderjahr”, “Fälligkeitsmonat” or “Bankarbeitstag” (calendar year, due month, banking day), generic usages 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, HeidelTime [Strötgen2010] was adapted to the domain. A final evaluation showed that the adjustments made to HeidelTime [Strötgen2010] significantly improved its performance. Particularly noteworthy is the recall, which rose by around 10 percentage points. Normalization, on the other hand, remains problematic, which is also due to generic or event-based uses of temporal expressions as well as legal references.

	strikt			partiell			strikt+value				partiell+value			
	P	R	F1	P	R	F1	P	R	F1	Acc	P	R	F1	Acc
HT	87,3	74,8	80,6	93,2	79,2	85,7	85,8	73,5	79,2	98,3	89,4	76,4	82,4	96,5
HT nV	92,3	86,4	89,2	95,9	89,5	92,6	90,2	84,4	87,2	97,7	92,3	86,2	89,1	96,3
+	5,0	11,6	8,7	2,7	10,3	6,9	4,4	10,9	8,0	−0,6	2,9	9,7	6,7	−0,2

Figure 5. Comparison of the results of the original version of HeidelTime (HT) with the modified (HT nV) on the dev-corpus. The last line shows the improvement

Dutch and Italian Temporal Expression Analysis service

For these languages, the HeidelTime [Strötgen2010] library is used.

2.3 GEOGRAPHICAL INFORMATION RECOGNITION (GEOLOCATION)

This service is responsible for the annotation and linking of geographical information in documents from the legal domain. Although this module is not completely finished, a working version is already available.

2.3.1 General Description of Method

This service is based on three different methods for annotating geographical entities: (i) Language Models; (ii) Dictionaries; and (iii) Rules.

2.3.1.1 Language Models

Statistical models

The language model method uses the same approach as described in named entity recognition based on OpenNLP (see Section *2.1.1.1 Statistical Language Model*). The linking of entities differs, because it uses a different source of external URIs. In the case of locations, the system points to Geonames URIs.[19] We use a SPARQL query against the Geonames ontology to retrieve the latitude and longitude of entities of the type location that we identify in the text.

Apart from that, we are experimenting with three other tools:

- **Hidden Markov Model:** The Hidden Markov Model is a statistical method working with Markov-chains which is often used in Natural Language Processing. Based on observations and transition probabilities, it infers the internal states. For our dataset, we used the *'hmm'* module by *nltk*.[20]
- **Conditional Random Fields:** on the contrary as in the named entity recognition where we used the *'sklearn-crfsuite'*, here we are using the *'CRFTagger'* by *nltk*.
- **BERT:** in that case we are using the same approach as in NER.

2.3.1.2 Dictionary based method

If a lexicon, dictionary or list of words is available, we use the dictionary for lexicon-based proper noun identification (with limited mechanisms for disambiguation). This method is based on the DictionaryNameFinder[21] module of OpenNLP. This module allows the spotting of entities defined in dictionaries.

2.3.1.3 Rules based Method

This approach uses a set of manually defined rules to identify geographical entities. The rules are written in a BNF format or directly written as regular expressions that are checked against the text using the RegExNameFinder[22] module of OpenNLP.

2.3.2 Description of Service within Lynx

This service accepts a text as input (both in plain text or NIF format). This text is analysed using one or several of the methods described above, and it returns a NIF based format document containing annotations for each of the geographic entities (itsrdf:taClassRef <http://dbpedia.org/ontology/Location>). Apart from the Entity Type annotation, it will also include an

[19] http://www.geonames.org

[20] https://www.nltk.org/

[21] https://opennlp.apache.org/docs/1.7.0/apidocs/opennlp-tools/opennlp/tools/namefind/DictionaryNameFinder.html

[22] https://opennlp.apache.org/docs/1.8.4/apidocs/opennlp-tools/opennlp/tools/namefind/RegexNameFinder.html

annotation linking the entity with an external linked data source (itsrdf:taIdentRef) for every entity. Finally, the service will assign a latitude and a longitude to the document computed as the average values of the latitude and longitude of all the found and linked entities.

An example of annotated named entities is shown in the listing below.

```
@prefix dbo:   <http://dbpedia.org/ontology/> .
@prefix eli:   <http://data.europa.eu/eli/ontology#> .
@prefix dct:   <http://purl.org/dc/terms/> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix lkg:   <http://lkg.lynx-project.eu/def/> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix nif:   <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
<http://lkg.lynx-project.eu/res/13a29d67#offset_11_17>
        a nif:OffsetBasedString , lkg:LynxAnnotation ;
        nif:anchorOf        "Berlin" ;
        nif:annotationUnit    [
                a nif:AnnotationUnit ;
                <http://www.w3.org/2003/01/geo/wgs84_pos/lat> "52.516666666666666" ;
                <http://www.w3.org/2003/01/geo/wgs84_pos/long> "13.383333333333333" ;
                itsrdf:taAnnotatorsRef  <http://dfki.lynx-project.eu/annotators/GEOLinker> ;
                itsrdf:taClassRef      dbo:Location ;
                itsrdf:taIdentRef      dbr:Berlin
        ] ;
        nif:annotationUnit    [
                a nif:AnnotationUnit ;
                itsrdf:taAnnotatorsRef  <http://dfki.lynx-project.eu/annotators/GEO> ;
                itsrdf:taClassRef      dbo:Location
        ] ;
        nif:beginIndex       "11"^^xsd:nonNegativeInteger ;
        nif:endIndex        "17"^^xsd:nonNegativeInteger ;
        nif:referenceContext  <http://lkg.lynx-project.eu/res/13a29d67> .
<http://lkg.lynx-project.eu/res/13a29d67>
        a lkg:LynxDocument , nif:Context ;
        lkg:metadata    [
                eli:id_local  "13a29d67" ;
                <http://lynx-project.eu/ontologies/nif#averageLatitude> 52.516666666666666 ;
                <http://lynx-project.eu/ontologies/nif#averageLongitude> 13.383333333333333 ;
                <http://lynx-project.eu/ontologies/nif#standardDeviationLatitude> 0.0 ;
                <http://lynx-project.eu/ontologies/nif#standardDeviationLongitude> 0.0 ;
                dct:language  "en"
        ] ;
        nif:beginIndex  "0"^^xsd:nonNegativeInteger ;
        nif:endIndex    "26"^^xsd:nonNegativeInteger ;
        nif:isString    "Welcome to Berlin in 2016." .
```

Table 10. Example of an annotated geographical entity in output NIF format

The three methods that have been previously described are going to be used in the Lynx project to annotate geographical entities in the legal domain.

2.3.2.1 Language Model Methods

This approach aims to identify more general rather than domain specific entities. Therefore, we have trained two different models using OpenNLP for the Lynx project using the training data provided by Nothman [Nothman2013]. The two models cover two languages, English and German: (i) English-LOC and (ii) German-LOC.

Experimentally, we are working on the development of a fine-grained classification of Geographical Entities. This classification has 14 more fine-grained location subcategories (such as city, country, state, park etc.) plus three main categories (organization, person, other) and it is used to annotate manually a small English dataset consisting of 92 texts on the Berlin district of Moabit which were crawled from the internet. The dataset has a total number of 3432 sentences and 57067 tokens. A sentence consists of 16.6 tokens on average. We have used this dataset to evaluate the two approaches described previously: HMM and CRF (BERT is still under development). To retrieve the F-score, the dataset was divided into a train set (2745 sentences) and a test set (687 sentences), respectively. The evaluation results are shown in Table 11.

	Precision	Recall	F-Measure
HMM	81.42	90.23	85.59
CRF	93.55	94.62	93.85

Table 11. Numeric evaluation of HMM, CRF and BERT using the manually annotated Moabit Dataset.

2.3.2.2 Dictionary based method

This approach is going to be mainly used in scenarios where specific geographical entities are available as lists. This approach is mainly relevant in the domain of Labour Law and Geothermal projects, i.e., a set of entities that would not be covered by general domain approach (as the language model).

For the moment being, there is one dictionary included in the module, which is listed in Table 12.

Dictionary Name	Dictionary Description	Number of Entries
SpanishCompanies	A dictionary containing names and website addresses of Spanish companies.	338607 names of companies

Table 12. Dictionaries currently available in the Lynx platform through the GEO service.

2.3.2.3 Rule based Method

This method is going to be used mainly in Scenario 1 “Contract Analysis”. In this scenario, and considering that the contracts are rental contracts, geographic information is essential, given that all the processing of the contract can be influenced by the location of the property (or object).

The two previous methods are not suitable for very fine-grained geographic entities, so we have chosen to use a set of rules for the identification of specific geographic entities in the analysis of contracts, because it proved difficult to identify specific addresses with language models or dictionaries, since the streets can have various names. Therefore, we worked on the implementation of a set of rules that allow us to identify several specific geographic information that is beyond the capacity of the two previous methods.

Address

The textual form of addresses has mainly a fixed structure, which makes it suitable for being recognized using regular expressions (rules). Therefore, we have defined specific rules (shown in Table 13) for the identification of addresses in the four languages that are relevant in the project: English, German, Spanish and Dutch.

Language	Rule
English (UK) Addresses	(^|,\s*)((?<number>(\d+[\s-–]*[\d]*)))\s+(?<street>[\wßäöüÄÖÜñ]+)(,?)\s+(?<city>(['\-a-zA-ZßäöüÄÖÜñ]+))\s+(?<zip>([\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>([\wßÁÉÍÓÚáéíóúäöüÄÖÜñ\s]+))($)
Dutch Addresses	(^|,)(?<street>[\wßäöüÄÖÜñ\s]+)\s+((?<number>(\d+[\s-–]*[\d]*)))?\s+(?<zip>(\d+))\s+(?<regioncode>([\wßäöüÄÖÜñ\s]+))\s+(?<city>(['\-\wßäöüÄÖÜñ\s]+))\s+(?<country>([\wßäöüÄÖÜñ\s\.]+))($)

Table 13. Rules defined for the recognition of addresses in four languages: EN, DE, ES and NL.

This approach is useful, but we already know that there are still many variations in addresses that cannot be covered by these rules. Therefore, we are still improving the rule based method in order to have a better recall recognizing a wider variety of addresses. Some examples of Spanish addresses that cannot be recognized are shown in the Table 14.

Language	Address	Description
Spanish	Calle Ledesma, 20, 4.º bis	The number of the building includes a specific text such as 'bis', which in Spanish means that this street has to number 4 buildings.
	C/ Florida, 19, 3.º izda.	The letter of the flat is described with more than one character: 'izdq' instead of 'I'.
	Manuel Iradier, 7, 3.º, dpto. 2	Flat identification includes 'dpto'.
	Ctra. Gasteiz-Irun, km 5	This is an address referring to a location on the countryside, which uses road names and kilometre points instead of street names and building numbers.
	Plaza Vicente Goikoetxea, 1, 1.º, 1.ª ofic.	This address includes the number of an office (1.ª ofic.) instead of a flat number or identifier.

Table 14. Examples of Spanish addresses that are not recognized by the defined rules.

3 SUMMARISATION SERVICES

In order to enable users to get a quick overview of the main ideas of a specific piece of content (paragraph, text, document, and multiple documents), methods for single document and also multi-document summarisation will be integrated into the Lynx platform. The goal is to add additional layers of useful annotations that enable the human experts to better and faster comprehend a document. This section describes the current state of the Summarisation Service implemented based on extractive and abstractive summarization methods and integrated into the Lynx platform.

3.1 GENERAL DESCRIPTION OF METHOD

3.1.1 Extractive summarization

The Centroid Summarisation is an unsupervised extractive summarisation method which is suitable for single or multiple documents ([Ghalandari2017]; [Rossiello2017]). It defines a sentence representation model, by assigning a score to each sentence.

A central element of this approach is utilizing the compositional properties of word embeddings. Word embeddings are continuous vector representations of words, which capture syntactic and semantic information. The Euclidean distance (or cosine similarity) between two-word vectors provides an effective method for measuring the linguistic or semantic similarity of the corresponding words, so that conjugation, synonyms or related concepts are close to each other in the embedding space. Furthermore, the learned embeddings have a meaningful linear substructure, so that the vector difference from man to woman is roughly similar to the one between king and queen (representing the underlying concept sex/gender) (see Figure 6).

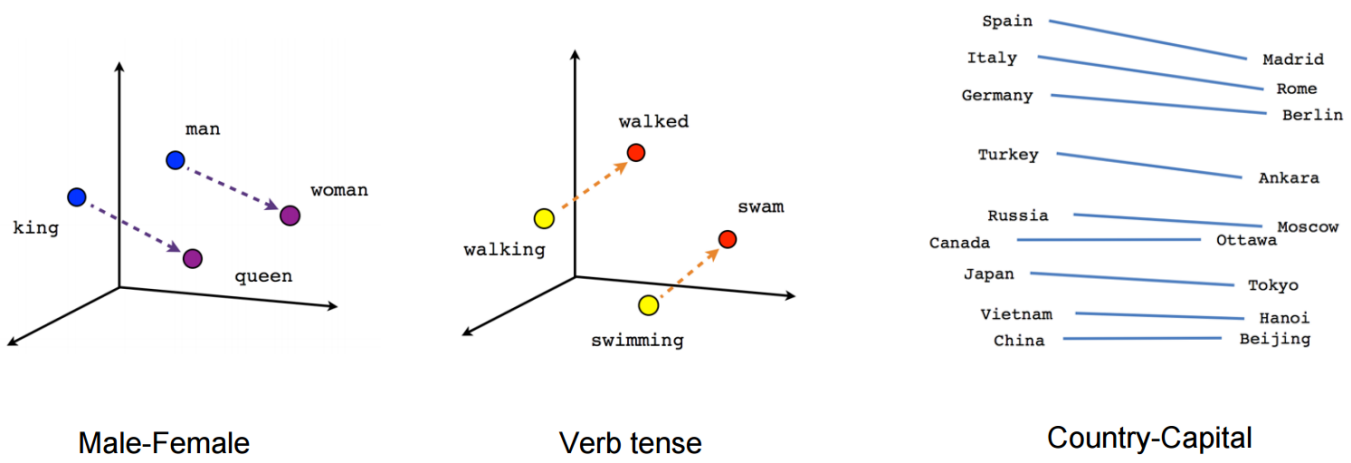

Figure 6. Word embedding 2D visualization displaying the vector difference between concepts

With the advance of neural network language models, several well performing methods were developed ([Pennington2014]; [Mikolov2013]). In many languages there is a variety of pre-trained word embeddings available, which were usually trained on billions of words.

The other crucial part is the term frequency-inverse document frequency, in short TFIDF [Neto2000]. It is a measure consisting of the product of two statistics. One is Term Frequency (TF), which is the number of times a term *t* occurs in a document *d*.

The other is the Inverse Document Frequency (IDF). It measures how often a word appears across all documents that were provided. This is done by taking the log of the total number of documents divided by the number of documents in which the term appears.

By taking the product of TF and IDF, we can calculate a measure for every term in a text which reflects how important it is in the document. The assumption is, that words that appear often within a document, but rarely in other given documents, must be a central element of the document. This way the most relevant words of a text can be extracted.

For our summarisation approach, we first collected a reference corpus that consisted of documents from the same field. If news articles were to be summarized, then the reference corpus would entail articles from different newspapers. With this data, we learn the IDF scores over the reference corpus after removing all stopwords. For single or multiple documents to be summarized, we calculated the TFIDF 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 selected all words with a weight above a certain threshold and got their embeddings. The properties of word embeddings were used to create a so-called centroid vector for one or multiple documents. This centroid represents the condensed meaningful information of one or more documents and is calculated by adding up word embeddings of the most relevant words.

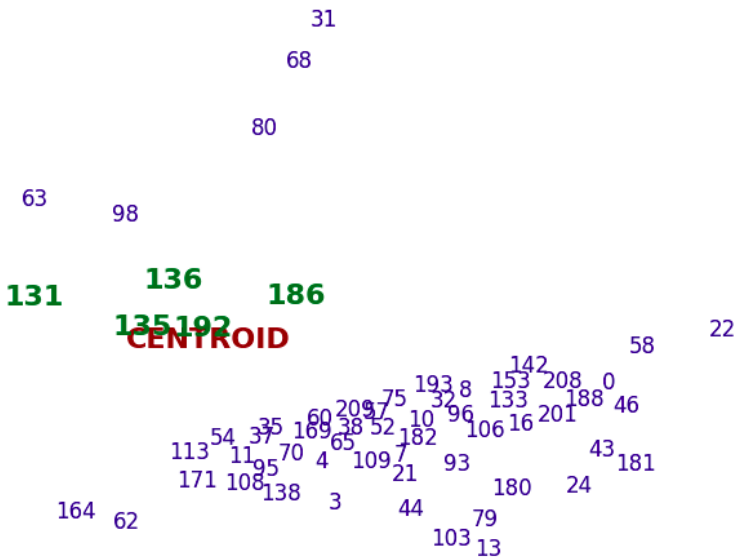

Figure 7. Centroid and sentence embedding 2D visualization [Rossiello2017]. Selected sentences are marked green.

In order to narrow down the number of sentences to extract from, we then calculated the relevance of each sentence. We used a combination of two measures. First the TFIDF [Neto2000] values of all words in the sentences were added up and divided by the sentence length. Additionally, we used the so called new-TFIDF measure. For every word that was used to calculate the centroid vector (and therefore represents some crucial information in the document), we checked in which sentence it was first used and then weighted these sentences. The reasoning behind this is, that normally when new terms or concepts are introduced, they are explained. Hence those sentences should be more relevant for the summarisation.

By adding up the word embeddings, the selected sentences were then embedded. Those sentence embeddings together with the centroid were then projected in the embedding space (see Figure 7). The closeness of the sentence embeddings to the centroid embedding represents their relevance to summarizing the document. To create the summary first the sentences closest to the centroid was 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

then a set threshold, it is assumed that it would not add much new information to the summary and it is therefore skipped.

3.1.2 Abstractive summarization

Our text summarization model is based on the Transformer architecture. This architecture adopts the original model of Vaswani et al. [Vaswani2017]. On top of the decoder, we use a Pointer-Generator (Equation 1) to increase the extractive capabilities of the network (we later refer to this architecture as CopyTransformer).

Equation 1. Pointer-Generator Formula

$$p(w) = p_{gen} P_{copy}(w) + (1 - p_{gen}) P_{softmax}(w),$$

where $P_{copy}(w)$ is the probability of copying a specific word w from the source document, $P_{softmax}(w)$ is the probability of generation a word calculated by the abstractive summarization model and p_{gen} is the probability of copying instead of generation.

Convolutional Self-Attention

The Transformer, like any other self-attention network, has a hierarchical multi-layer architecture. In many experiments it was shown that this architecture tends to learn lexical information in the first layers, sentence-level patterns in the middle and the semantics in the upper layers [Raganato2018, Tenney2019]. The disadvantage of this approach is that during the attention operation it considers all tokens as equally important, whereas syntactic information is mostly concentrated in certain local areas. This problem is usually specified as the problem of locality modelling. As syntactic information can help in identifying more important words or phrases, it could be beneficial to focus attention on these regions.

A successful approach to the locality modelling task is the so-called convolutions (local) self-attention networks [Yang2019]. Essentially, the problem is dealt with by the application of a 1-dimensional convolution to the self-attention operation at the network's lower layers. This strengthens dependencies among neighbouring elements and makes the model distance-aware when it searches for low-level patterns in a sequence. In other words, it restricts the attention scope to the window of neighbouring elements. A complete description can be found in [Aksenov2020].

BERT-Conditioned Encoder

The main task of the encoder is to remember all the semantic and syntactic information from the input text which should be used by the decoder to generate the output. Knowledge transfer from the language model should theoretically improve its ability to remember the important information due to the much larger corpus used in its pre-training phase compared to the corpus used in the text summarization training phase. We thus condition our encoder on the BERT language model.

BERT-Windowing

One of the key features of our approach is its ability to overcome the length limitations of BERT, allowing it to deal with longer documents. BERT's maximum supported sequence length is 512 tokens[23], which is smaller than the average size of texts used in most summarization datasets. Our method relies on the well-known method of windowing which to our knowledge was never used before neither in the BERT-based models nor in abstractive text summarization research (Figure 8). We apply BERT to the windows

[23] These are not tokens in the traditional sense, but so-called WordPiece tokens, see [Devlin2018].

of texts with strides and generate *N* matrices, every matrix embedding one window. Then we combine them by doing the reverse operation. The vectors at the overlapping positions are averaged (by summing and dividing by the number of overlapping vectors). As a result, we have the matrix of embeddings with the shape of the hidden size times the length of the text. The drawback of this approach is that we reduce the size of the context as each resulted vector is calculated based on maximum twice the window size number of tokens. Besides, the split of the text to equal size windows will aggravate the consistency of the input as some sentences will be split in an arbitrary manner between two adjacent windows. Despite this drawback, we assume that this procedure will nevertheless improve the accuracy of the encoder trained on the non-truncated texts. We set the window size to the maximum size of 512 tokens and the stride to 256. We consider this stride size optimal due to a trade-off between the average context size and computational requirements of the model (number of windows). By this trade we ensure every token to have a 768 tokens-context except for the 256 initial and final tokens, which only have 512 tokens-context.

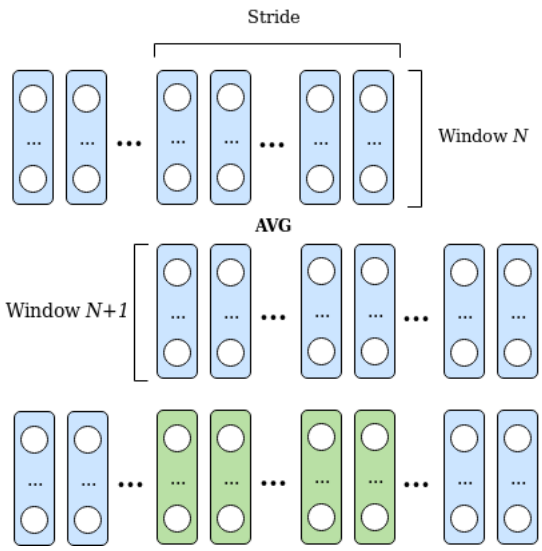

Figure 8. Integration of BERT-generated contextual representations from two windows

BERT-Conditioned Decoder

In the decoder, pre-training was applied in a similar way. The main difference is that instead of the final output of BERT we use only its word embedding matrix (without positions). The reason behind this is that in the decoder the generated probability distribution is conditioned on the incomplete text (previous summary draft output) while BERT implicitly assumes consistent and completed input [Zhang2019]. As context-independent embeddings are not enough to represent the minimum set of features to make a meaningful prediction, the custom Transformer decoder is always stacked on top of BERT.

Our whole BERT-based model is similar to One-Stage BERT [Zhang2019] and BertSumAbs [Liu2019] but differs in the usage of the four last hidden states of BERT to create contextualized representation, in presence of Pointer Generator and capabilities to process long texts. In Figure 9 we show the schema of the basic model with the BERT-conditioned convolutional self-attention encoder and BERT-conditioned decoder.

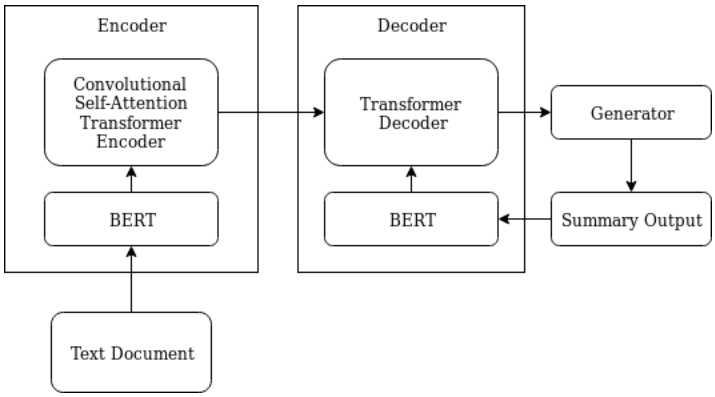

Figure 9. Model overview

Integration of BERT and Convolutional Self-Attention

We evaluated two different ways of integrating the BERT-conditioning with the convolutional self-attention of the model's encoder (Figure 10).

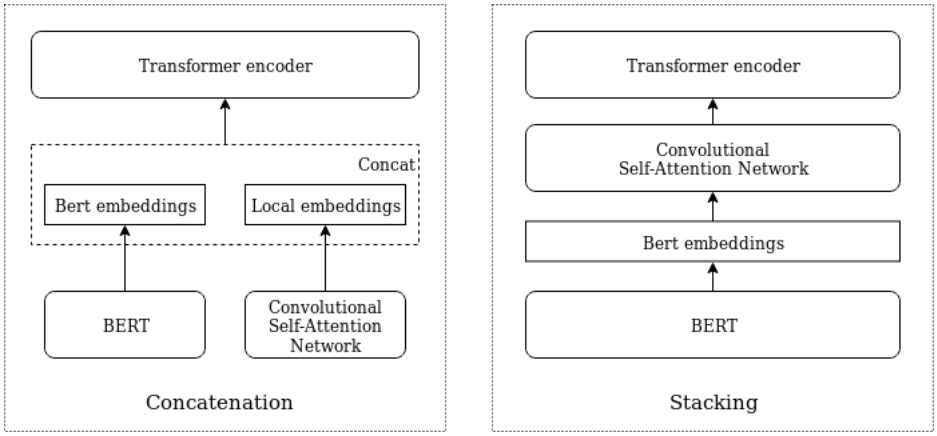

Figure 10. Two different approaches for the integration of the BERT-conditioning with Convolutional Self-Attention

- Stacking: This approach comprises feeding the BERT-generated embeddings to the convolutional self-attention Transformer encoder. A potential problem with this approach is that convolutional self-attention is assumed to be beneficial when applied in the lower layers as its locality modelling feature should help in modelling of local dependencies (e.g., syntax). At the same time, BERT is a hierarchical model where the last layers target high-level patterns in the sequences (e.g., semantics). We assume that the application of the network detecting the low-level patterns on BERT's output can undermine its generalization abilities.
- Concatenation: Because of the considerations raised above, we also develop a second approach which we call Concatenation. We split the convolutional self-attention Transformer encoder into two networks where the first one uses only convolutional self-attention and the second original self-attention (identical to the Transformer encoder). Then we feed the original sequences into BERT and into the convolutional self-attention network in parallel. The resulting embedding vectors are concatenated and fed into the Transformer encoder. In this way, we model the locality at the lower layers of the encoder at the cost of a smaller depth of the network (assuming the same number of layers).

A complete description of this method can be found in [Aksenov2020].

3.2 DESCRIPTION OF SERVICE WITHIN LYNX

3.2.1 Extractive summarization

The usage of the Summarisation service in Lynx is different depending on every business use case. In the case of Contract Analysis a single-document approach has to be used, while in Labour Law and Geothermal Project Analysis a multi-document summarisation approach is needed. Apart from the approach, the format in which the information is provided also differs from one use case to the other: NIF document (Contract Analysis) and JSON (others).

To evaluate our model, we tested it against several other methods on Task 2 of the DUC 2004 Conference [DUC2004]. The corpus used covers 50 topics each with 10 newspaper articles. For validation, several manually written summaries for each topic were provided. To measure the quality of the summarisation we calculated the recall based rouge score (Lin 2004), which compares generated and human summaries on the basis of n-gram over- laps. We tested for Rouge-1 and Rouge-2 scores using the original Pearl script with the following settings ROUGE-1.5.5 with options -c 95 -b 665 -m -n 2 -x. The summary length was set to be 665 bytes long, longer generated summaries were cut off after 665 bytes. As an absolute baseline, we used LEAD, which is simply the first 665 bytes from the most recent article of each cluster. Additionally, we compared against the popular probabilistic model called SumBasic [Nenkova2005] and LexRank [Erkan2011], another frequently used summarisation algorithm which analyses connections between sentences. Finally, we compared our method against the traditional Centroid methods using bag of words instead of embeddings (**C_BOW**) and the improved version [Rossiello2017] using googles pretrained wordembeddings (**C_GNEWS**). For comparability with **C_GNEWS** we used the same pretrained wordembeddings, we set the topic threshold of our model to be 0.1 and the similarity threshold to 0.9 and picked the top 3 sentences of each newspaper according to our TFIDF preselection. As seen in Table 15 we could improve over all the compared methods, both in terms of the Rouge-1 and Rouge-2 score.

MODEL	ROUGE-1	ROUGE-2
LEAD	32.42	6.42
SUMBASIC	37.27	8.58
LEXRANK	37.58	8.78
C_BOW	37.76	8.08
C_GNEWS	37.91	8.45
OURS	**38.41**	**9.26**

Table 15. Results of the Summarisation Service evaluation using the Task 2 of the DUC 2004 Conference [DUC2004].

Currently the extractive summarisation module is working with English language texts.

3.2.2 Abstractive summarization

We aim to develop a system that works in a language-independent way. It assumes that either the upstream components are available in the respective language, or they are themselves language-independent, such as the multi-lingual version of BERT. Since most summarization datasets are in English however, we use English for the evaluation and additionally include German to check if of our model can be applied to another language. The evaluation was performed using two different datasets: (1) the **CNN/Daily Mail Dataset** [Hermann2015; Nallapati2016], containing in a collection of news articles paired with multi-sentence summaries published on the CNN and Daily Mail websites; and (2) the **SwissText Dataset**, created for the 1st German Text Summarization Challenge at the 4th Swiss Text Analytics Conference – SwissText 2019 [ZHAW2019], used to evaluate the efficiency of the model in a multi-lingual, multi-domain environment.

Our system is built on the OpenNMT library. For training, we use cross-entropy loss and the Adam optimizer with the Noam decay method [Kingma2014]. Regularization is made via dropout and label smoothing. For evaluation, we calculate the F1-scores for ROUGE using the files2rouge library. The ROUGE evaluation is made on the sequences of WordPiece tokens.

Locality Modelling

To evaluate the effect of convolution on self-attention we introduce it in the first layer of the encoder. We use the same kernel sizes as in [Yang2019]. In these experiments, to accelerate the training process, we use a small model with a hidden size of 256, four self-attention heads and three layers in the encoder and decoder. All models are trained for 90,000 training steps with the Coverage Penalty. As a baseline, we use our implementation of CopyTransformer. In contrast to [See2017], we do not re-use the attention layer for the decoder but train a new Pointer- Generator layer from scratch.

The results are presented in Table 16. We see that both convolutions over tokens and over attention heads improve the ROUGE scores. Standard convolution outperformed circular convolution on ROUGE-1, ROUGE-2 and ROUGE-L by 0.06, 0.13 and 0.09 percent, respectively.

Method	ROUGE-1	ROUGE-2	ROUGE-L
CopyTransformer	31.95	14.49	30.02
+ 1D conv.	32.62	14.99	30.74
+ 2D conv.	**32.72**	**15.12**	**30.85**
+ 2D Circular conv.	32.68	15.01	30.76

Table 16. Ablation study of model with Convolutional Self- Attention on the CNN/Daily Mail dataset (kernel sizes are 11 and 3)

BERT Conditioning

To find the optimal architecture of the BERT-based ab- stractive summarizer we conducted an ablation study (Table 17). All hyperparameters were set equal to the ones in experiments in convolutional self-attention. On CNN/Daily Main dataset we test three different models: BERT encoder+Transformer Decoder, BERT-Transformer encoder+Transformer decoder and BERT-Transformer encoder+BERT-Transformer decoder. The version of BERT used in the experiments is BERT-Base. As the baseline, we use the Transformer without Pointer Generator. From the results, we observe that BERT improves the efficiency of the model when it is used in both encoder and decoder. Besides, BERT in the encoder is more effective when it is used to produce embeddings to be used by the standard Transformer encoder than when it is used solely as an encoder. Even without a Pointer Generator, our model outperformed the CopyTransformer baseline by 1.28, 0.5 and 1.24 on ROUGE-1, ROUGE-2 and ROUGE-L.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	24.82	6.27	22.99
CopyTransformer	31.95	14.49	30.02
Bert Encoder + Transformer Decoder	31.3	13.37	29.46
Bert-transformer Encoder + Transformer Decoder	32.5	14.68	30.68
Bert-transformer Encoder + Bert-transformer Decoder	**33.23**	**14.99**	**31.26**
Transformer (full text)	23.18	5.15	21.48
Bert-transformer Encoder + Transformer Decoder (full text)	**31.51**	**14.1**	**29.77**

Table 17. Ablation study of the BERT-based model on truncated and original CNN/Daily Mail dataset

To evaluate our BERT-windowing method we conducted the experiments on the full text. Our approach outperforms the baseline, which proves that the method can be successfully applied to texts longer than

512 tokens. The final performance of this model is still lower than that of the model trained on the truncated text, but as the same pattern can be observed for the baselines we assumed this relates to the specifics of the dataset that is prone to having important information in the first sentence of a text.

On SwissText data we use the multilingual version of BERT-Base. We evaluated two models with Bert-transformer encoder and Transformer and BERT-Transformer decoders (Table 18). The introduction of BERT into the transformer increased the ROUGE-1, ROUGE-2 and ROUGE-L scores by 7.21, 8.91 and 7.51 percent, respectively. At the same time, the usage of BERT in the decoder decreased the overall score. We assume that the reason behind this is that in multilingual BERT, due to its language-independence, the embedding matrix outputs less precise contextualized representations which undermines their benefits for the summarization task.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Transformer	36.40	20.69	34.14
CopyTransformer	39.44	25.11	37.16
Bert-transformer Encoder + Transformer Decoder	**44.01**	**29.60**	**41.65**
Bert-transformer Encoder + Bert-transformer Decoder	43.22	29.01	40.84
Transformer (full text)	34.76	18.65	32.61
Bert-transformer Encoder + Transformer Decoder (full text)	**45**	**30.49**	**42.64**

Table 18. Ablation study of the BERT-based model on the truncated and original SwissText dataset

On the non-truncated texts, usage of the Bert-transformer encoder increased the ROUGE scores by 10.23, 11.84 and 10.03 percent. Furthermore, it gives us higher scores com- pared to the same model on truncated texts. This demonstrates the usability of BERT-windowing for this particular dataset. We assume that the difference in performance on the CNN/Daily Mail datasets reflects the difference in distribution of the useful information within the text. Particularly, that in the SwissText dataset, it is spread more uniformly than in the CNN/Daily Mail dataset. We con- ducted a small experiment comparing the average ROUGE score between a golden summary and the head and the tail of a document (taking the first or last n sentences, where n correlates to the length of the gold summary) on both datasets. The difference between taking the head and a tail on the SwissText dataset (ROUGE-L of 34.79 vs. 20.15, respectively) was much smaller than on CNN/Daily Mail (ROUGE-L of 16.95 vs. 12.27, respectively) which confirms our hypothesis.

Integration Strategies

To evaluate the integration strategies, we trained two models with the respective BERT-based baselines. Both models have in their encoder two Transformer layers and one Convolutional Transformer layer placed on top of BERT or in parallel, respectively (Table 19).

Method of Integration	Model	ROUGE-1	ROUGE-2	ROUGE-L
Stacking	BERT+CopyTransformer	35.28	**17.12**	33.31
	BERT+Convolutional CopyTransformer	**35.4**	16.82	**33.31**
Concatenation	BERT+CopyTransformer	34.82	16.46	32.79
	BERT+Convolutional CopyTransformer	**35.26**	**16.79**	**33.22**

Table 19. Different strategies for integrating language models with convolutional Self-Attention (CNN/Daily Mail dataset)

The method of stacking does not provide any significant improvement. With the introduction of convolutional self- attention only ROUGE-1 increased by 0.12 percent, while ROUGE-2 dropped by 0.3 and ROUGE-L remained the same. Considering that in many domains ROUGE-2 maximally correlates with human assessment, we dismiss this method. The concatenation strategy convolution is shown to be much more efficient, increasing ROUGE scores by 0.44,0.33 and 0.43 percent. This con- firms our hypothesis that locality modelling is the most efficient when applied at the bottom on the non-contextualized word representations. Unfortunately, this model failed to outperform the stacking baseline. We conclude that

the concatenating architecture undermines the performance of the Transformer model, and the convolutional self-attention is not beneficial when used together with pre-trained language models. Hence, we decided to train our two final models separately.

Model Comparison

For the final comparison of our model to other state-of-the-art methods we conducted experiments on the CNN/Daily Mail dataset. We set the hidden state to 512, the number of Transformer layers in the encoder and layers to six and the number of self-attention heads to eight. Hence, our baseline is smaller than the original CopyTransformer [Gehrmann2018], which may be the reason why it performs slightly worse (Table 20). BERT-conditioning was used in both the encoder and decoder. The sizes of convolution kernels are set to 13 and three. The networks were trained for 200,000 training steps on a single NVIDIA GeForce GTX 1080 Ti. The generation of the summary was made via the Beam search algorithm with the Beam size set to four. Finally, the generated summaries were detokenized back to the sequences of words separated by spaces.

Method	ROUGE-1	ROUGE-2	ROUGE-L
BiLSTM + Pointer-Generator + Coverage (See et al., 2017)	39.53	17.28	36.38
ML + Intra-Attention (Paulus et al., 2018)	38.30	14.81	35.49
CopyTransformer (Gehrmann et al., 2018)	39.25	17.54	36.45
Bottom-Up Summarization (Gehrmann et al., 2018)	41.22	18.68	38.34
One-Stage BERT (Zhang et al., 2019)	39.50	17.87	36.65
Two-Stage BERT (Zhang et al., 2019)	41.38	19.34	38.37
ML + Intra-Attention + RL (Paulus et al., 2018)	39.87	15.82	36.90
Key information guide network (Li et al., 2018)	38.95	17.12	35.68
Sentence Rewriting (Chen and Bansal, 2018)	40.88	17.80	38.54
BertSumAbs (Liu and Lapata, 2019)	**41.72**	**19.39**	**38.76**
CopyTransformer (our implementation)	38.73	17.28	35.85
Convolutional CopyTransformer	38.98	17.69	35.97
BERT+CopyTransformer (enc., dec.)	**40**	**18.42**	**37.15**

Table 20. ROUGE scores for various models on the CNN/Daily Mail test set. The first section shows different state-of-the-art models, the second section presents our models and baseline.

For the BERT-based model, we set the minimum length of a generated summary to 55 as we found that without such restriction the model was prone to generate shorter sequences than in the test dataset. The model outperformed the baseline by 1.27 on ROUGE-1, 1.14 on ROUGE-2 and 1.3 on ROUGE-L. This is better than the scores of One- Stage BERT but still worse than the two-stage and Bert- SumAbs models.

For the convolutional CopyTransformer we use convolutional self-attention in the first three layers of the encoder. It increased ROUGE-1, ROUGE-2 and ROUGE-L by 0.25, 0.41 and 0.12.

Furthermore, we present the first publicly available benchmark for the SwissData dataset (Table 21).

Method	ROUGE-1	ROUGE-2	ROUGE-L
CopyTransformer (our implementation)	39.5	22.36	36.97
Convolutional CopyTransformer	40.54	23.62	38.06
BERT+CopyTransformer (enc.)	**42.61**	**25.25**	**39.85**

Table 21. ROUGE scores for our models on the SwissText test set

All parameters are equal to the CNN/Daily Mail baseline. BERT-conditioning was used only in the encoder. The networks were trained on the truncated texts in 90,000 training steps. From the results we see that the convolutional CopyTransformer showed much more efficiency than on CNN/Daily Mail dataset, outperforming the baseline by 1.04 percent on ROUGE-1, 1.26 on ROUGE-2 and 1.09 on ROUGE-L. The BERT-based model achieved the highest scores.

Currently the abstractive summarisation module is working for English and German language. The summary generated by this service is included into the NIF document as a main document annotation. An example of such a NIF annotation output (in bold) is shown in Table 22.

```
@prefix eli:   <http://data.europa.eu/eli/ontology#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#> .
@prefix lkg:   <http://lkg.lynx-project.eu/def/> .
@prefix nif:   <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dct:   <http://purl.org/dc/terms/> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://lynx-project.eu/res/doc7840>
        a lkg:LynxDocument , nif:Context ;
        lkg:metadata    [
                eli:id_local  "d45a28ce" ;
                lkg:summary   "tomas medina caracas was a member of the high command of the fuerzas armadas de
                    colombia . he had been in the cross - hairs of the u . s . and manufacturing . he was charged with
                    conspiracy to import cocaine into the united states .\n\n\n" ;
                dct:language  "en"
        ] ;
        nif:beginIndex  "0"^^xsd:nonNegativeInteger ;
        nif:endIndex    "2203"^^xsd:nonNegativeInteger ;
        nif:isString    "bogota , colombia -lrb- cnn -rrb- -- a key rebel commander and fugitive from a u.s. drug trafficking
            indictment was killed over the weekend in an air attack on a guerrilla encampment , the colombian military said
            monday . alleged cocaine trafficker and farc rebel tomas medina caracas in an interpol photo . tomas medina
            caracas , known popularly as `` el negro acacio , '' was a member of the high command of the fuerzas armadas
            revolucionarias de colombia and , according to colombian and u.s. officials , helped manage the group 's
            extensive cocaine trafficking network . he had been in the cross-hairs of the u.s. justice department since 2002 .
            he was charged with conspiracy to import cocaine into the united states and manufacturing and distributing
            cocaine within colombia to fund the farc 's 42-year insurgency against the government . u.s. officials alleged
            medina caracas managed the rebel group 's sales of cocaine to international drug traffickers , who in turn
            smuggled it into the united states . he was also indicted in the united states along with two other farc
            commanders in november 2002 on charges of conspiring to kidnap two u.s. oil workers from neighboring
            venezuela in 1997 and holding one of them for nine months until a $ 1 million ransom was paid . officials said
            the army 's rapid response force , backed by elements of the colombian air force , tracked medina caracas down
            at a farc camp in the jungle in the south of the country . `` after a bombardment , the troops occupied the camp
            , and they 've found 14 dead rebels so far , along with rifles , pistols , communications equipment and ... four
            gps systems , '' defense minister juan manuel santos said at a news conference . `` the death of ` el negro acacio
            ' was confirmed by various sources , including members of farc itself . '' medina caracas commanded farc 's 16th
            front in the southern departments of vichada and guainia . established in 1964 as the military wing of the
            colombian communist party , farc is colombia 's oldest , largest , most capable and best-equipped marxist rebel
            group , according to the u.s. department of state . e-mail to a friend . journalist fernando ramos contributed to
            this report ." .
```

Table 22. Example of NIF output of the Summarisation Service using the abstractive model for English

4 CONCLUSIONS AND FUTURE WORK

The semantic annotation and summarisation services described in this document are part of the building blocks that have been developed in the Lynx project. All of them are available and fully functional through the Lynx platform. At the time of writing this document, results provided by tests carried out during the implementation stage are promising.

The services were implemented with the requisite that they can be managed by the Curation Workflow Manager (described in deliverables *D4.4 Initial Implementation and Report of Data and Content Curation Services* [LynxD44] and *D4.5 Final implementation and report of Data and Content Curation Services* that will be released in month M36). Therefore, they are microservices that are able to scale, if needed, and are living and being executed in Docker containers. They run independently from other services communicating through REST APIs. A summary of the services developed, and the locations of their deployments and code repositories can be found below (in Table 23).

Acronym	Name	Temporary Deployment URL	Code URL
NER	Named Entity Recognition	http://dfkiner-88-staging.cloud.itandtel.at	https://gitlab.com/superlynx/dfki_ner
TIMEX	Temporal Expression Analysis	http://upmtimex-88-staging.cloud.itandtel.at	https://gitlab.com/superlynx/upm_timex
GEO	Geographical Information Extraction	http://geolocation-88-staging.cloud.itandtel.at	https://gitlab.com/superlynx/geolocation
SUMM	Summarisation	http://summarization-88-staging.cloud.itandtel.at	https://gitlab.com/superlynx/summarization

Table 23. List of services together with its documentation and deployment URLs

In the case of named entity recognition, we have a working system including a specific model for each required language in the project: English, German, Spanish and Dutch. These models are based on BERT, one of the newest technologies for NLP, which provides interesting results for the recognition of general domain entities of four types: PERSON, ORGANIZATION, LOCATION and MISCELLANEA.

Regarding Temporal Expression Analysis, the service in all languages needed for the project is already up and running. The latest developments in this service have been focused on three points. First, an analysis of the main needs of specific legal temporal expressions (such us “five working days”) and additional representations to the standard used, such as the identification of intervals for some of the languages. Then, an agreement on representation of these expressions has been reached (since the TimeML standard offers no support to this kind of expressions), and it is currently being documented. Finally, the expansion of the set of rules in the service in order to cover them has been determined.

Geolocation has been further developed to offer two new capabilities: dictionaries and rules. The generation and usage for recognition of specific dictionaries, especially for the Geothermal Project Analysis and the Labour Law scenarios, where specific lists of entities (or even concepts) are common practice. The main contribution to the Lynx platform from the second new functionality of the GEO service is the definition and the implementation of rules for fine-grained annotation of Geographic entities such as addresses, especially interesting in the Contract Analysis scenario. These rules have been defined in order to recognize addresses, which as mentioned, can have complicated structures. Therefore, these rules will be further improved.

The summarisation service has been completely developed for English and German texts, using two different approaches: extractive and abstractive summarization. These modules are based on neural networks and, as previously described, obtain remarkable results in comparison with state-of-the-art technology.

ANNEX 1. API DESCRIPTIONS

The API description of the Lynx services can be found here: http://lynx-project.eu/doc/api/.

Below, we present the list of API endpoints exposed by each of the services.

NER

A full documentation of this service can be found here: http://lynx-project.eu/doc/api/ner.html, but the methods relevant to the Lynx project are the following:

- GET /ner/listmodels, which returns a JSON object containing all available models for performing annotation of named entities.
- POST /ner/analyzetext, which process the input text (in plain text or NIF format defined in the ***'Content-Type'*** header) and enriches it with semantic annotation about named entities. It will always return a NIF document (RDF Turtle or JSON-LD defined in the ***'Accept'*** header) including the annotations. The input document is provided in the body of the request. Apart from that, the required parameters include:
  - Language of the input text (en, de, es, nl).
  - Models: name of the models to be used in the processing. The available models are:
    - ***ner-wikinerEn_ORG***: a general domain language model for the recognition of organizations in English.
    - ***ner-wikinerEn_PER***: a general domain language model for the recognition of persons in English.
    - ***ner-de_aij-wikinerTrainORG***: a general domain language model for the recognition of organizations in German.
    - ***ner-de_aij-wikinerTrainPER***: a general domain language model for the recognition of persons in German.
  - Mode: mode of the processing: **'spot'** for spotting, **'link'** for linking and **'all'** for both.

TIMEX

A full documentation of this service can be found here: http://upmtimex-88-staging.cloud.itandtel.at/swagger-ui.html, while a demonstrator of its functionality is available here: http://annotador.oeg-upm.net/, but the method relevant to the Lynx project so far is the following:

- POST /annotate/temporal, which process the input text (NIF or plain text) and returns an annotated version, following the TIMEX or the NIF format.

GEO

A full documentation of this service can be found here: http://lynx-project.eu/doc/api/geo.html, but the methods relevant to the Lynx project so far are the following:

- GET /geolocation/listmodels, which returns a JSON object containing all available models for performing annotation of Geographical entities.
- POST /geolocation/analyzetext, which process the input text (in plain text or NIF format defined in the ***'Content-Type'*** header) and enriches it with semantic annotation about geographical entities. It will always return a NIF document (RDF Turtle or JSON-LD defined in the ***'Accept'*** header) including the annotations. The input document is provided in the body of the request. Apart from that, the required parameters include:

- Language of the input text.
- Analysis: identify the type of recognition: Dictionary approach – **dict**, Language Model approach – **language** or Rule-based approach – **rules**.
- Models: name of the models to be used in the processing. The available models for each analysis type are:
  - Language:
    - ***ner-wikinerEn_LOC***: a general domain language model for English.
    - ***ner-de_aij-wikinerTrainLOC***: a general domain language model for German.
  - Dict:
    - ***DICTSpanishCompanies***: a dictionary containing names and website addresses of Spanish companies.
  - Rules: there are four models (one for each important language in the project) to recognize addresses. The models are: ***'rules-address-es'*** (for Spanish), ***'rules-address-en'*** (for English), ***'rules-address-de'*** (for German), and ***'rules-address-nl'*** (for Dutch).
- Mode: mode of the processing: **'spot'** for spotting, **'link'** for linking and **'all'** for both.

SUMM

A full documentation can be found here: http://lynx-project.eu/api/doc/summ.html, but the methods relevant to the Lynx project are the following:

- GET /summarization/listmodels, which returns a JSON object containing all available models for performing summarization.
- POST /summarization/summarizetext, which allows querying the API with plain text or a NIF formatted document (defined in the ***'Content-Type'*** header) and generates a summary for the provided text included in the output document (RDF Turtle or JSON-LD defined in the ***'Accept'*** header). The input document is provided in the body of the request. Apart from that, the required parameters include:
  - Language of the input text.
  - Models: name of the models to be used in the processing. The available models are:
    - ***German***: single document abstractive summarization model for German documents.
    - ***English***: single document abstractive summarization model for English documents.

ANNEX 2. QUESTIONNAIRE ON TEMPORAL EXPRESSIONS

On July 2019, the following questionnaire was sent to each of the use cases in the project in order to gather temporal expression requirements to improve the Temporal Expression Analysis service.

QUESTIONNAIRE

Introduction on temporal expressions

Temporal Expressions are any word or sequence of words referring to a time instant (e.g. 'five o'clock') or a time interval (e.g. 'from nine to ten'). Temporal expressions frame events or happenings implicitly or explicitly mentioned in the document. Following the ISO-TimeML standard, we distinguish among dates, times, durations and sets; additionally, we also plan to add intervals.

- DATE: Calendar expressions such as 'October 7, 1991', '22/01/2018', or '1992'; also relative expressions like 'Two days ago'.
- TIME: Points in time ('At seven o'clock', '22:30', '3.30pm'...), absolute or relative ('Half an hour ago', 'In two minutes and three seconds').
- DURATION: Amounts of time like 'Two days', 'Three years and six months', 'Two centuries', 'One hour and 20 minutes' or 'Half an hour'.
- SET: Repetitions in time (such as 'Monthly', 'Twice a week', 'Every Monday', 'Three times a year', 'Every first of the month'...).
- INTERVAL: Period between two temporal expressions ('from 14h to 20h', 'from Monday to Friday'…).

In general, we can consider a temporal expression is any expression than answers "when" and "how long".

Nevertheless, current temporal taggers (the tools that identify and normalize, this is, give a standard value to these expressions) are not prepared to deal with legal documents. They are not able, for instance, to detect expressions such as "five working days", since they are not usually found in other domains.

To be able to properly improve the Lynx service to detect these expressions, we would like you to answer the following questions about your needs and interests regarding temporal expressions.

1. Temporal expressions of interest.

Is there any specific temporal expression you would like to find that is not common to other domains? (such as "working days"). Please don't hesitate to contact us for doubts. Anything you would like to find/has any time-relate value to you is of interest.

Temporal expression	Different ways to express it (languages, different ways to express it)	Example/Other comments
five working days	días laborables (es), X working days	...
years of contribution	worked years (en), years of contribution (en), años cotizados (es)	...
the date of signature	...	...
...	...	...

2. Temporal expressions to refer.

Usually temporal expressions are relative, such as if for instance we say “two days ago”. Temporal taggers tend to normalize with regard to the present day or to the date of creation of the document. Is there any other type of date you would be interested to use as anchor date, such as the date of publication of a document or the date of signature? It can vary depending on the type of document. Please let us know any relevant metadata related to time you would like to consider to this aim, or other considerations we are not taking into account. If this information is not available as metadata, any hints on where to find it within the text is also helpful.

Temporal expression	Type of document/other info	Where to find this information in the text if not available as metadata	Other comments
Date of signature	Contracts	End of the document	...
Date of the decision	worked years (en), years of contribution (en), años cotizados (es)	End of the document	...
...	...	...	...

3. Useful temporal expressions

Maybe not all the temporal expressions in a text are necessarily useful. For instance, some legal references, such as “European Council of the 3 June 2018” or “Real Decreto-ley 10/2018, de 24 de agosto” can include them without being actually a date in the text, but part of a legal reference. Do you have any preference on how to deal with it?

☐ I would mark them all.

☐ I would mark them all, but distinguishing them somehow (such as having different types or timelines in the text).

☐ I would just mark expressions from the text, not the ones that are part of a legal reference.

☐ No strong opinion, not important.

4. Other comments

Please let us know any other comments on how to deal with temporal expressions in the Lynx service:

REFERENCES

[Aksenov2020] Dmitrii Aksenov, Julián Moreno-Schneider, Peter Bourgonje, Robert Schwarzenberg, Leonhard Hennig, and Georg Rehm. Abstractive Text Summarization based on Language Model Conditioning and Locality Modeling. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Christopher Cieri, Khalid Choukri, Thierry Declerck, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020), Marseille, France, 5 2020. European Language Resources Association (ELRA).

[Benikova2015] Benikova, D., Yimam, S.M., Santhanam, P., Biemann, C.: Germaner: Free opengerman named entity recognition tool. In: Proceedings of the International Confer-ence of the German Society for Computational Linguistics and Language Technol-ogy, GSCL 2015, University of Duisburg-Essen, Germany, 30th September - 2ndOctober 2015. pp. 31–38 (2015).

[Devlin2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional trans- formers 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), pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

[DocBook2010] Walsh, N., Hamilton, R. L. (2010). DocBook 5: The Definitive Guide. Sebastopol, CA: O'Reilly. ISBN: 978-0-596-80502-9.

[DUC2004] Document Understanding Conference 2004. https://duc.nist.gov/duc2004/.

[Eckart2016] Eckart de Castilho, R., M´ujdricza-Maydt,´ E., Yimam, S.M., Hartmann, S.,Gurevych, I., Frank, A., Biemann, C.: A web-based tool for the integrated annota-tion of semantic and syntactic structures. In: Hinrichs, E.W., Hinrichs, M., Trippel,T. (eds.) Proceedings of the Workshop on Language Technology Resources andTools for Digital Humanities, LT4DH@COLING, Osaka, Japan, December 2016.pp. 76–84. The COLING 2016 Organizing Committee (2016).

[Erkan2011] Erkan, Gunes and Dragomir Radev (2011). “LexRank: Graph-based Lexical Centrality As Salience in Text Summarization”. In: Journal of Artificial Intelligence Research - JAIR 22. doi: 10.1613/jair.1523.

[Gehrmann2018] Gehrmann, S., Deng, Y., and Rush, A. (2018). Bottom-up abstractive summarization. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4098–4109.

[Ghalandari2017] Ghalandari, Demian Gholipour (2017). “Revisiting the Centroid-based Method: A Strong Baseline for Multi-Document Summarization”. In: arXiv preprint arXiv:1708.07690.

[Hermann2015] Hermann, K. M., Kovisky´, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15, pages 1693–1701, Cambridge, MA, USA. MIT Press.

[Hochreiter1997] Hochreiter, S. & Schmidhuber, Jü. (1997). Long short-term memory. Neural computation, 9, 1735--1780.

[Kingma2014] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. International Conference on Learning Representations, 12.

[Leitner2019] Leitner, Elena & Rehm, Georg & Moreno-Schneider, Julián. (2019). Fine-grained Named Entity Recognition in Legal Documents. SEMANTICS 2019. Submission currently under review.

[Lin2004] Lin, Chin-Yew (2004). "ROUGE: A Package for Automatic Evaluation of summaries". In: p. 10.

[Liu2019] Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

[LynxD11] Jorge González-Conejero, Emma Teodoro, & Pompeu Casanovas. (2018). Lynx D1.1 Functional Requirements Analysis Report. Zenodo.

[LynxD41] Julián Moreno-Schneider, & Georg Rehm. (2018). D4.1 Pilots Requirements Analysis Report. Zenodo.

[LynxD42] Julián Moreno-Schneider, & Georg Rehm. (2018). D4.2 Intermediate version of Workflow definition. Zenodo.

[LynxD43] Julián Moreno-Schneider, & Georg Rehm. (2019). D4.3 Final version of Workflow definition. Zenodo.

[LynxD44] Julián Moreno-Schneider, & Georg Rehm. (2019). D4.4 Initial Implementation and Report of Data and Content Curation Services. Zenodo.

[Mikolov2013] Mikolov, Tomas et al. (2013). "Distributed representations of words and phrases and their compositionality". In: Advances in neural information processing systems, pp. 3111–3119.

[Nallapati2016] Nallapati, R., Zhou, B., dos Santos, C., Gúlçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pages 280–290, Berlin, Germany, August. Association for Computational Linguistics.

[Navas-Loro2019] Navas-Loro, María et al. (2019). TempCourt: Evaluation of temporal taggers on a new corpus of court decisions. The Knowledge Engineering Review, 34, E24. doi:10.1017/S0269888919000195

[Navas-Loro2020] Navas-Loro, María & Rodríguez-Doncel, Víctor (2019). "Annotador: a Temporal Tagger for Spanish". To be published in the Journal of Intelligent and Fuzzy Systems (expected Q2-2020).

[Nenkova2005] Nenkova, Ani and Lucy Vanderwende (2005). "The impact of frequency on summarization". In:

[Neto2000] Neto, Joel Larocca et al. (2000). "Document clustering and text summariza- tion". In: Mikolov, Tomas et al. (2013). "Distributed representations of words and phrases and their compositionality". In: Advances in neural information processing systems, pp. 3111–3119.

[Nothman2013] Nothman, Joel & Ringland, Nicky & Radford, Will & Murphy, Tara & R. Curran, James. (2013). Learning multilingual named entity recognition from Wikipedia. Artificial Intelligence. 194. 151–175. 10.1016/j.artint.2012.03.006.

[Pennington2014] Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: Proceedings of the 2014 con- ference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

[Proisl2016] Proisl, T., Uhrig, P.: Somajo: State-of-the-art tokenization for german web andsocial media texts. In: Cook, P., Evert, S., Sch¨afer, R., Stemle, E. (eds.) Proceedingsof the 10th Web as Corpus Workshop, WAC@ACL 2016, Berlin, August 12, 2016.pp. 57–62. Association for Computational Linguistics (2016)

[Raganato2018] Raganato, A. and Tiedemann, J. (2018). An analysis of encoder representations in transformer-based machine translation. In Proceedings of the 2018 EMNLP Work- shop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 287–297, Brussels, Belgium, November. Association for Computational Linguistics.

[Raj2020] Development and Evaluation of a BERT model for NER. Kamal Raj. Visited at 20/05/2020. https://github.com/kamalkraj/BERT-NER.

[Rehm2019] Georg Rehm, Julian Moreno-Schneider, Jorge Gracia, Artem Revenko, Victor Mireles, Maria Khvalchik, Ilan Kernerman, Andis Lagzdins, Marcis Pinnis, Artus Vasilevskis, Elena Leitner, Jan Milde, and Pia 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, June 2019. Co-located with NAACL 2019. 7 June 2019. In print.

[Reimers2014] Reimers, N., Eckle-Kohler, J., Schnober, C., Kim, J., Gurevych, I.: Germeval-2014:Nested named entity recognition with neural networks. In: Faaß, G., Ruppenhofer,J. (eds.) Workshop Proceedings of the 12th Edition of the KONVENS Conference.pp. 117–120. Universit¨atsverlag Hildesheim (Oktober 2014)

[Reimers2017a] Reimers, N., Gurevych, I.: Optimal hyperparameters for deep lstm-networks forsequence labeling tasks. CoRRabs/1707.06799(2017)

[Reimers2017b] Reimers, N., Gurevych, I.: Reporting score distributions makes a difference: Per-formance study of lstm-networks for sequence tagging. In: Proceedings of the 2017Conference on Empirical Methods in Natural Language Processing. pp. 338–348.Association for Computational Linguistics (2017)

[Rossielo2017] Rossiello, Gaetano, Pierpaolo Basile, and Giovanni Semeraro (2017). “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, pp. 12–21.

[See2017] See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. Proceedings of the 55th Annual Meeting of the Associa- tion for Computational Linguistics (Volume 1: Long Papers).

[Tenney2019] Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.

[Yang2019] Yang, B., Wang, L., Wong, D. F., Chao, L. S., and Tu, Z. (2019a). Convolutional self-attention networks. Proceedings of the 2019 Conference of the North.

[Zhang2019] Zhang, H., Cai, J., Xu, J., and Wang, J. (2019). Pretraining-based natural language generation for text summarization. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pages 789–797, Hong Kong, China, November. Association for Computational Linguistics.

[ZHAW2019] ZHAW. (2019). German text summarization challenge. Swiss Text Analytics Conference. Available online.