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

D3.3 Intermediate 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	Ēriks Ajausks, Andis Lagzdiņš (Tilde), Christian Sageder (OLS)
VERSION | STATUS	V1 | Final
NATURE	Other
DISSEMINATION LEVEL	Public
DOCUMENT DOI	10.5281/zenodo.3235752
DATE	31/05/2018 (M18)

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 draft version	10/04/2019	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.2	First draft of structure	18/04/2019	
0.3	Include DFKI annotation services	07/05/2019	
0.4	Include DFKI Summarization Service	08/05/2019	
0.5	Include UPM TIMEX service	09/05/2019	Maria Navas
0.6	Exec. summary, introduction, conclusions	10/05/2019	Julián Moreno Schneider (DFKI), Georg Rehm (DFKI)
0.9	Include reviewers comments	24/05/2019	
1.0	Final remarks	29/05/2019	

LIST OF ACRONYMS

BB	Building Block
BiLSTM	Bilateral LSTM
CRF	Conditional Random Fields
GEO	Geographical Location
LKG	Legal Knowledge Graph
LSTM	Long-Short Term Memory
NER	Named Entity Recognition
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 an overview of the intermediate summarisation and annotation services (developed under the Task 3.3 of WP3 in the Lynx project). This report describes several services that are divided into Annotation Services, which goal is enriching documents by annotating semantic information on them; and Summarisation Service, which aim at generating a new and shorter piece of a text from one or several texts (documents or parts of documents).

The description of the services is composed of two parts: for each of the services, first, its general approach is presented. 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.

Being an intermediate report, the described services are still under development, and will still experience changes and improvements in the following months. However, large amounts of work have been done in the interoperability aspect (data interchange format) and the conversion of the services for working in a docker microservice architecture. The fact that the services are already properly running and working in Openshift is a success.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

This report aims to describe the status of services as of the end of M18 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 points in which more work must be done; (iii) the risk factors in the future development. The second is the documentation of the status 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 containers in order to deploy them all under the same platform (using Openshift). 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 intermediate 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 current status of semantic annotation services which aim at annotating and enriching documents within the legal domain.

Section 3 gives an overview of the current status of the summarisation service which aims at generating summaries from documents.

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 at the end in the appendices.

2 ANNOTATION SERVICES

This section describes three annotation services developed in the Lynx project. The goal is the enrichment of documents in different business use case (see deliverable D4.1 and D4.2 [LynxD41, LynxD42]). First, a general description of the service functionality is given, 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 three different approaches: (i) language model; (ii) Conditional Random Fields (CRF); and (iii) Bilateral Long Short Term Memory Neural Networks (BiLSTM).

2.1.1.1 Language Model

The recognition of Named Entities based on language models was implemented using the Name Finder module[1] 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 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[2] and DBPedia spotlight.[3] If a URI is retrieved (the most likely reasons for not retrieving one are either because no Wikipeda 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.[4]

2.1.1.2 Conditional Random Fields (CRF) method

Conditional Random Fields present a statistical modeling method used for structured prediction. CRFs can be considered as a sequence modeling 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,[5] 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] https://opennlp.apache.org/docs/1.8.3/apidocs/opennlp-uima/opennlp/uima/namefind/NameFinder.html

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

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

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

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

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.

2.1.1.3 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 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 (NEs) 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 Scenario 1 “Contract Analysis” and Scenario 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 the language models approach, while the CRF and BiLSTM methods have not been included in the deployed service yet. Including them into the development is already foreseen in the next steps of the project.

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 training models for other languages.

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

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#>.
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84 pos/>.

<http://link.omitted/documents/document1#offset_0_36>
    a nif:RFC5147String , nif:String , nif:Context;
    nif:beginIndex "0"^^xsd:nonNegativeInteger;
    nif:endIndex "36"^^xsd:nonNegativeInteger;
    nif:isString "Angela Merkel was in Berlin in 2016."^^xsd:string;
    dfkinif:averageLatitude "52.516666666666666"^^xsd:double;
    dfkinif:averageLongitude "13.383333333333333"^^xsd:double;
    dfkinif:standardDeviationLatitude "0.0"^^xsd:double;
    dfkinif:standardDeviationLongitude "0.0"^^xsd:double;
    nif:meanDateRange "20160101010000 20170101010000"^^xsd:string.

<http://link.omitted/documents/#offset_0_12>
    a nif:RFC5147String , nif:String;
    nif:anchorOf "Angela Merkel¨^^xsd:string;
    nif:beginIndex "0"^^xsd:nonNegativeInteger;
    nif:endIndex "12"^^xsd:nonNegativeInteger;
    itsrdf:taClassRef <http://dbpedia.org/ontology/Person>;
    nif:referenceContext <http://link.omitted/documents/#offset_0_36>;
    itsrdf:taIdentRef <http://dbpedia.org/resource/Angela_Merkel>.
```

Figure 1. Example of annotated named entities in NIF format

2.1.2.2 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, Legal Entity Recognition (LER), consists of 750 German court decisions published on the portal "Rechtsprechung im Internet".[6] The source text was collected from the XML documents, split into sentences and words by SoMaJo [Proisl2016] and annotated manually in WebAnno [Eckart2016]. The dataset[7] is freely available for download under CC BY 4.0 license.[8] 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

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

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

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

list of entity types can be seen in Figure 2. 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 2. Named Entities available in the LER dataset

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, 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'…).

The service is currently 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 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 the NIF format or as TIMEX3 tags. Just the input text, its language and an optional reference date are needed. The figures below show the output of the service for the example sentence below:

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

```
@prefix nif-ann: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-annotation#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@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#> .

<#offset_92_105>  a           nif:RFC5147String , nif:String ;
        nif:anchorOf          "twice an hour" ;
        nif:beginIndex        "92"^^xsd:nonNegativeInteger ;
        nif:endIndex          "105"^^xsd:nonNegativeInteger ;
        nif:normalizedDate    "1H" ;
        nif:referenceContext  <#offset_0_106> ;
        itsrdf:taClassRef     <http://www.w3.org/2006/time#TemporalEntity> .

<#offset_0_106>  a      nif:RFC5147String , nif:String , nif:Context ;
        nif:beginIndex  "0"^^xsd:nonNegativeInteger ;
        nif:endIndex    "106"^^xsd:nonNegativeInteger ;
        nif:isString    "The trial will begin tomorrow and will last two days and three
hours. There will be reports twice an hour."@en .

<#offset_21_29>  a            nif:RFC5147String , nif:String ;
        nif:anchorOf          "tomorrow" ;
        nif:beginIndex        "21"^^xsd:nonNegativeInteger ;
        nif:endIndex          "29"^^xsd:nonNegativeInteger ;
        nif:normalizedDate    "2019-02-15" ;
        nif:referenceContext  <#offset_0_106> ;
        itsrdf:taClassRef     <http://www.w3.org/2006/time#TemporalEntity> .

<#offset_44_68>  a            nif:RFC5147String , nif:String ;
        nif:anchorOf          "two days and three hours" ;
        nif:beginIndex        "44"^^xsd:nonNegativeInteger ;
        nif:endIndex          "68"^^xsd:nonNegativeInteger ;
        nif:normalizedDate    "PT2D3H" ;
        nif:referenceContext  <#offset_0_106> ;
        itsrdf:taClassRef     <http://www.w3.org/2006/time#TemporalEntity> .
```

Figure 3. 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>

Figure 4. 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[9], 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).

Regarding the evaluation of the service, for English we will use the Tempcourt [10] corpus and the TempEval3[11] corpus, 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. For Spanish, a new legal corpus is currently under development in order to test the performance of the service.

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.

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

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

[11] https://www.cs.york.ac.uk/semeval-2013/task1/

	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. Since the rules can be extended (as done in the German service), they might be eventually extended regarding its performance.

2.3 GEOGRAPHICAL INFORMATION RECOGNITION (GEOLOCATION)

This service is responsible for the annotation and linking of geographical information in documents from the legal domain. This module is currently under development, although 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
(iii) Rules

2.3.1.1 Language Model Method

The language model method uses the same approach as described in Named Entity Recognition based on OpenNLP (see Section 2.1.1.1). 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.[12] 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.

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[13] 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 and converted into a set of regular expressions that are checked against the text using the RegExNameFinder[14] module of OpenNLP.

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

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

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

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 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 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 rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix itsrdf: <http://www.w3.org/2005/11/its/rdf#>.
@prefix nif: <http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84 pos/>.

<http://link.omitted/documents/document1#offset_0_26>
    a nif:RFC5147String , nif:String , nif:Context;
    nif:beginIndex "0"^^xsd:nonNegativeInteger;
    nif:endIndex "26"^^xsd:nonNegativeInteger;
    nif:isString "Welcome to Berlin in 2016."^^xsd:string;
    dfkinif:averageLatitude "52.516666666666666"^^xsd:double;
    dfkinif:averageLongitude "13.383333333333333"^^xsd:double;
    dfkinif:standardDeviationLatitude "0.0"^^xsd:double;
    dfkinif:standardDeviationLongitude "0.0"^^xsd:double.

<http://link.omitted/documents/#offset_11_17>
    a nif:RFC5147String , nif:String;
    nif:anchorOf "Berlin¨^^xsd:string;
    nif:beginIndex "11"^^xsd:nonNegativeInteger;
    nif:endIndex "17"^^xsd:nonNegativeInteger;
    itsrdf:taClassRef <http://dbpedia.org/ontology/Location>;
    nif:referenceContext <http://link.omitted/documents/#offset_0_26>;
    geo:lat "52.516666666666666"^^xsd:double;
    geo:long "13.383333333333333"^^xsd:double;
    itsrdf:taIdentRef <http://dbpedia.org/resource/Berlin>.
```

Figure 6. 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. The adaptations done are described in the next sections.

2.3.2.1 Language Model Method

This approach aims to identify more general rather than domain specific entities. Therefore, we have trained two different models 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.

Although we have only generated models for German and English, the Wikiner collection includes also data in other languages such as Spanish, so training models for other languages could be done.

2.3.2.2 Dictionary based method

This approach is going to be mainly used in Scenario 3 “Geothermal Project Analysis” in order to identify specific Geographical Entities that are only relevant in the domain of Geothermal projects, i.e., a set of entities that would not be covered by general domain approach (as the language model). The concrete dictionaries that are going to be included have not been defined yet, but they will be included in the final report (deliverable D3.8 Summarisation and annotation services).

2.3.2.3 Rules based Method

This method is going to be used mainly in Scenario 1 “Contract Analysis”. In this scenario, and taking into account 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 are working 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.

Currently, we are working on the definition of the necessary rules that will be completely described in the deliverable D3.8.

3 SUMMARISATION SERVICE

In order to enable users to get a quick overview of the main ideas of a specific piece of content (paragraph, text, document, multiple documents), methods for 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.

3.1 GENERAL DESCRIPTION OF METHOD

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 7).

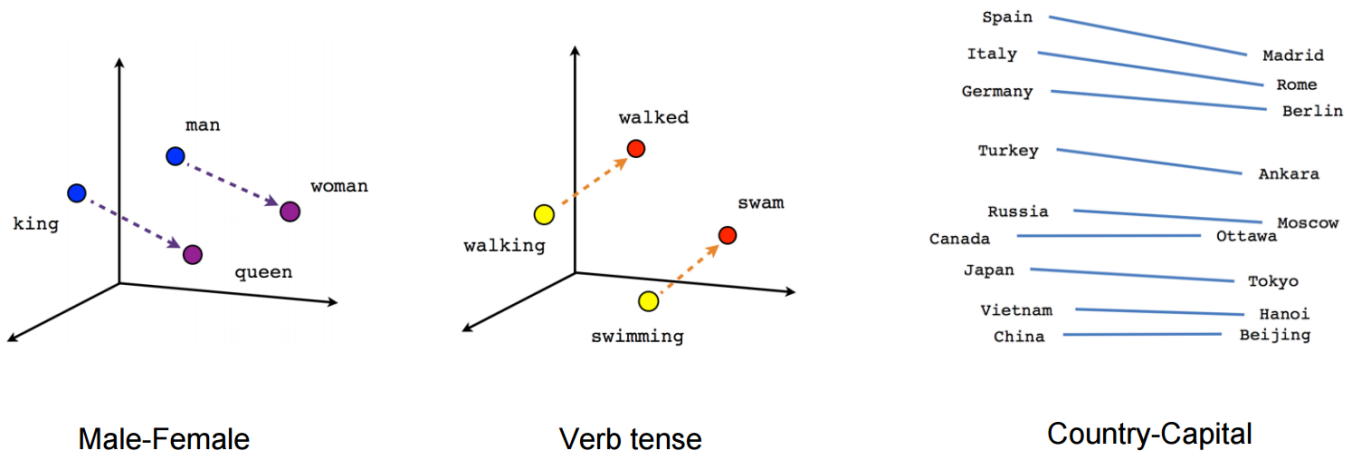

Figure 7. Word embedding 2D visualization displaying the vector difference between concepts

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.

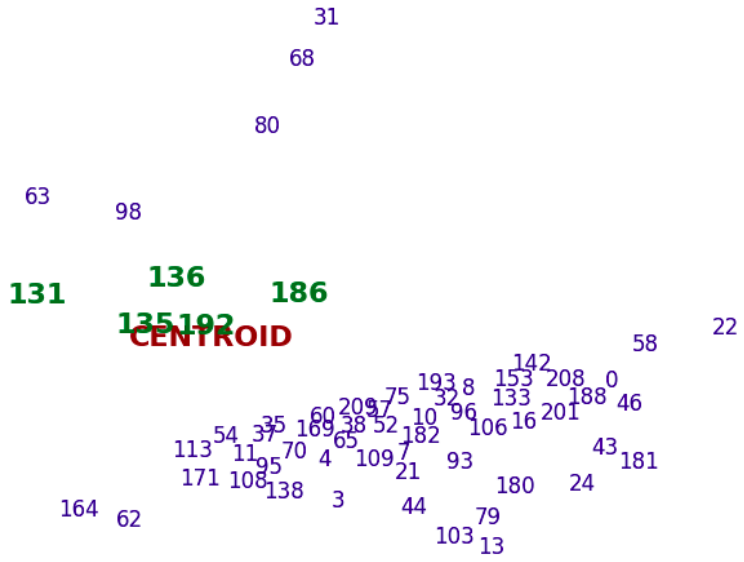

Figure 8. Centroid and sentence embedding 2D visualization [Rossiello2017]. Selected sentences are marked green.

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 8). 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.

3.2 DESCRIPTION OF SERVICE WITHIN LYNX

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 analyzes 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 1 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 1. Results of the Summarisation Service evaluation using the Task 2 of the DUC 2004 Conference [DUC2004].

Currently the summarisation module is only working with English language texts, but it is planned to train the different components for all the languages of the Lynx project: German (DE), Spanish (ES) and Dutch (NL). If there is time and resources available, Italian (IT) will also be considered.

The summary generated by this service is included into the NIF document (in the case of single-document) summarisation as a main document annotation. An example of such a NIF annotation output (in bold) is shown in Figure 9. The output for the multi-document summarisation is still under definition and it will be completely described in D3.8.

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

<http://lynx-project.eu/documents/#offset_0_1134>
a nif:RFC5147String , nif:String , nif:Context ;
nif:beginIndex "0"^^xsd:nonNegativeInteger ;
nif:endIndex "1134"^^xsd:nonNegativeInteger ;
nif:isString "COUNCIL REGULATION (EC) No 881/2002\\n\\nof 27 May 2002\\n\\nimposing certain specific restrictive measures directed against certain persons and entities associated with the ISIL (Da'esh) and Al-Qaida organisations\\n\\n▼B\\n\\n\\n\\nArticle 1\\n\\nFor the purpose of this Regulation, the following definitions shall apply:\\n\\n1. ‘funds’ means financial assets and economic benefits of every kind, including but not limited to cash, cheques, claims on money, drafts, money orders and other payment instruments; deposits with financial institutions or other entities, balances on accounts, debts and debt obligations; publicly and privately traded securities and debt instruments, including stocks and shares, certificates presenting securities, bonds, notes, warrants, debentures, derivatives contracts; interest, dividends or other income on or value accruing from or generated by assets; credit, right of set-off, guarantees, performance bonds or other financial commitments; letters of credit, bills of lading, bills of sale; documents evidencing an interest in funds or financial resources, and any other instrument of export-financing;\\n\\n." ;
nif:summary "Text of the Summary" .

Figure 9. Example of NIF output of the Summarisation Service

4 CONCLUSIONS AND FUTURE WORK

The various services described in this deliverable are part of the processing building blocks that are under development in the Lynx project. They are focused on semantic annotation and summarisation. While they are still in a preliminary stage, all of them are already available through the Lynx platform and can be used. The results are promising.

The services were implemented with the requisite that they can be managed by the Curation Workflow Manager (described in deliverables D4.4 and D4.5). Therefore, they are microservices that will be able to scale if needed and will be living and executed in Docker containers. They will run independently from other services communicating through REST APIs.

A summary of the services developed up to this moment, and the locations of their preliminary deployments and code repositories can be found below (in Table 2).

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

Table 2. List of services together with its documentation and deployment URLs

In the case of Named Entity Recognition and Geolocation, future work consists of training models for the other languages of the project, mainly Spanish. Offering training capabilities for future generations of new models (for other languages or specific domains) is also foreseen. Specifically, for the Geolocation service, two other next steps are under consideration: the generation of specific dictionaries for the Geothermal Project Analysis scenario and the definition and the implementation of rules for fine-grained annotation of Geographic entities, especially in the Contract Analysis scenario.

Regarding Temporal Expression Analysis, a first version of the service in all languages needed for the project is already up and running. Next steps include three points. First, an analysis of the main needs of specific legal temporal expressions (such us “five working days”) and anchor dates (usually in other fields the reference date is the date of creation of a document, but in the legal domain are different dates to consider such as “date to enter in to force” or “date of publication”) is needed. Then, it is important to reach an agreement on representation of these expressions (since the TimeML standard offers no support to this kind of expressions). Finally, the expansion of the set of rules in the service in order to cover them has to be determined.

The summarisation service is under development. The multi-document summarisation has not been developed and deployed yet. Also, this service is not yet completely compatible with NIF format regarding the input and output.

All of the missing implementations are underway. Further enhancements will be implemented in July.

Important future work entails the definition of a legal corpus for the different use cases and languages that can be used for testing the different services in the legal domain. This will provide deeper and more detailed evaluation measurements.

ANNEX 1. API DESCRIPTIONS

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

Below, we present the list of API endpoints exposed by each of the services for which such an API has already been implemented.

NER

A full documentation of this service can be found here: http://lynx-project.eu/api/doc/ner.html, but the methods relevant to the Lynx project so far 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) and enriches it with semantic annotation about named entities. It will always return a NIF document including the annotations. Possible parameters include:
  - Language of the input text.
  - Analysis: identify the type of recognition: Dictionary approach – **dict**, Language Model approach – **language** and all available models – **all**.
  - Models: name of the models to be used in the processing.
  - 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-dev-int.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/api/doc/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) and enriches it with semantic annotation about geographical entities. It will always return a NIF document including the annotations.

SUMM

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

- POST /summarization/summarizetext, which allows querying the API with plain text or a NIF formatted document and generates a summary for the provided text. Possible parameters include:
  - lengthPercentage: defines the length of the summary (in sentences) based on the length of the input text (in sentences).

REFERENCES

[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).

[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.

[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.

[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.

[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.

[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.

[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.

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

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

[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.

[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)

[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.