THE UNIVERSITY *of* EDINBURGH

# Edinburgh Research Explorer

## European Language Grid: A Joint Platform for the European Language Technology Community

**Link:**
Link to publication record in Edinburgh Research Explorer

**Document Version:**
Publisher's PDF, also known as Version of record

**Published In:**
Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations

OPEN ACCESS

# European Language Grid:
# A Joint Platform for the European Language Technology Community

**Georg Rehm[1], Stelios Piperidis[2], Kalina Bontcheva[3], Jan Hajič[4], Victoria Arranz[5], Andrejs Vasiļjevs[6], Gerhard Backfried[7], José Manuel Gómez-Pérez[8], Ulrich Germann[9], Rémi Calizzano[1], Nils Feldhus[1], Stefanie Hegele[1], Florian Kintzel[1], Katrin Marheinecke[1], Julián Moreno-Schneider[1], Dimitrios Galanis[2], Penny Labropoulou[2], Miltos Deligiannis[2], Katerina Gkirtzou[2], Athanasia Kolovou[2], Dimitris Gkoumas[2], Leon Voukoutis[2], Ian Roberts[3], Jana Hamrlová[4], Dušan Variš[4], Lukáš Kačena[4], Khalid Choukri[5], Valérie Mapelli[5], Mickaël Rigault[5], Jūlija Meļņika[6], Miroslav Jánošík[7], Katja Prinz[7], Andrés García-Silva[8], Cristian Berrío[8], Ondrej Klejch[9], Steve Renals[9]**

[1]DFKI GmbH, Berlin, Germany; [2]R. C. Athena, ILSP, Greece; [3]University of Sheffield, UK;
[4]Charles University, Czechia; [5]ELDA, France; [6]Tilde, Latvia; [7]HENSOLDT Analytics, Austria;
[8]Expert System, Spain; [9]University of Edinburgh, UK

Corresponding author: georg.rehm@dfki.de

## Abstract

Europe is a multilingual society, in which dozens of languages are spoken. The only option to enable and to benefit from multilingualism is through Language Technologies (LT), i. e., Natural Language Processing and Speech Technologies. We describe the European Language Grid (ELG), which is targeted to evolve into the primary platform and marketplace for LT in Europe by providing one umbrella platform for the European LT landscape, including research and industry, enabling all stakeholders to upload, share and distribute their services, products and resources. At the end of our EU project, which will establish a legal entity in 2022, the ELG will provide access to approx. 1300 services for all European languages as well as thousands of data sets.

## 1 Introduction

Europe is a multilingual society with 24 EU Member State languages and dozens of additional languages including regional and minority languages and languages spoken by immigrants, trade partners and tourists. The only option to enable and to benefit from multilingualism is through Language Technologies (LT) including Natural Language Processing (NLP) and Speech Technologies (Rehm, 2017). While the European LT landscape is world class, it is also massively fragmented (Vasiljevs et al., 2019; Rehm et al., 2020d).

We describe Release 2 of the European Language Grid (ELG) cloud platform.[1] This scalable system is targeted to evolve into the primary

platform for LT in Europe. It will provide one umbrella platform for all LTs developed by the European LT landscape, including research and industry, addressing a gap that has been repeatedly raised by the European LT community for many years (Rehm and Uszkoreit, 2013; Rehm et al., 2016b; STOA, 2017; Rehm, 2017; Rehm and Hegele, 2018; European Parliament, 2018). ELG is meant to be a virtual home and marketplace for all products, services and organisations active in the LT space in Europe (Rehm et al., 2020a). The platform can be used by all stakeholders to showcase, share and distribute their products, services, tools and resources. At the end of the EU project ELG (2019-2022), which will establish a legal entity in early 2022, the platform will provide access to approx. 1300 commercial and non-commercial tools and services for all European languages, as well as thousands of language resources (LRs). ELG will enable the European LT community to deposit and upload their technologies and data sets and to deploy them through the grid. The ELG is also meant to support *digital language equality* in Europe (STOA, 2017; European Parliament, 2018), i. e., to create a situation in which all languages are supported through technologies equally well. The current imbalance is characterised by a stark predominance of LTs for English, while almost all other languages are only marginally supported and, thus, in danger of digital language extinction (Rehm and Uszkoreit, 2012; Kornai, 2013; Rehm et al., 2014, 2016a; ELRC, 2019).

Section 2 gives an overview of the ELG platform and related activities. Section 3 touches upon related work. Section 4 concludes the paper.

---

[1]https://www.european-language-grid.eu. We provide a screencast demo video at https://youtu.be/LD6QadkkZiM.

## 2 The European Language Grid

The European LT community has been demanding a dedicated LT platform for years. ELG concentrates on *commercial* and *non-commercial* LTs, both *functional* (processing and generation, written and spoken language) and *non-functional* (corpora, data sets etc.). We want to establish the ELG as the primary market place for the fragmented European LT landscape (Rehm et al., 2020d) to connect demand and supply. The ELG is based on robust, scalable and reliable open source technologies, enabling it to scale with the growing demand and supply. It contains records of all resources, service and application types, languages as well as LT companies, research organisations, projects, etc. (see Figure 1 and Figure 4 in the appendix).
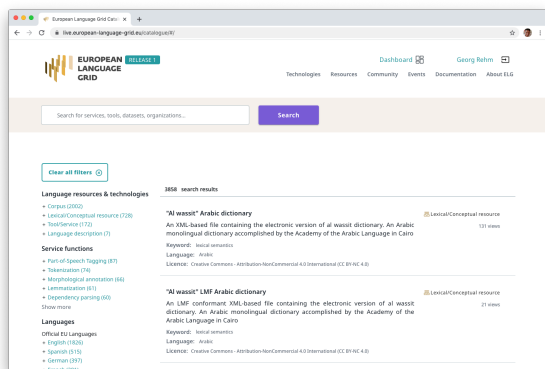


Figure 1: The ELG platform

### 2.1 Architectural Overview

ELG is a scalable platform with a web user interface, backend components and REST APIs. It offers access (search, discovery, etc.) to various kinds of LT-related resources such as functional services as well as corpora and data sets and organisations. An ELG functional service is an LT tool wrapped with the ELG LT Service API[2] and packaged in a Docker container; both steps have to be carried out by the LT provider. Then, the LT service container is integrated into the ELG (Section 2.7) so that it can be used through the web UI or APIs. The architecture consists of three layers: *base infrastructure*, *platform backend*, *platform frontend* (Figure 2).

The *base infrastructure* is operated on a Kubernetes[3] cluster in the data centre of a cloud provider located in Berlin, Germany, where all platform

---

[2]https://gitlab.com/european-language-grid/platform/
[3]https://kubernetes.io

components and all LT functional services run as Docker containers. The only components outside the cluster are the S3 storage, ReadtheDocs (ELG documentation), and any LT services deployed through external servers.
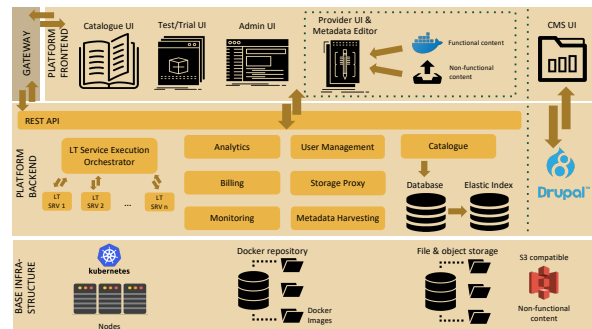


Figure 2: Technical architecture

The *platform backend* consists of (1) the *backend components* of the ELG catalogue, i. e., an inventory of all metadata records (Section 2.3). Users can browse and search the catalogue through queries or by utilising filters (e. g., language, service type, domain etc.). Users with the "LT provider" role can create new entries either by uploading XML descriptions or through a graphical metadata editor. The catalogue backend is implemented using Django, PostgreSQL and ElasticSearch. (2) The LT Service Execution Server offers a common REST API for executing functional services, also handling failures, time-outs etc. (3) The user management and authentication module is based on Keycloak, an identity and access management solution. (4) The Storage Proxy is used for interacting with the S3-compatible storage. (5) All integrated LT services. Additional components, especially for billing and monitoring purposes, are currently work in progress.

The *platform frontend* consists of UIs for the different types of users, e. g., LT providers, potential buyers and administrators (Section 2.6.2). These include (1) catalogue UIs (browse, search, view), provider and metadata editor UIs for uploading and registering functional and non-functional resources. They are implemented using React and packaged in the same container. (2) The administration pages are implemented using Django. (3) The test/trial UIs for functional services run in separate containers. The UIs are powered by the catalogue REST API, e. g., a resource's metadata record is returned as a JSON object and rendered as HTML. The frontend also includes a Drupal-based CMS for additional content (Section 2.6.2).

All core components of the ELG platform are built with robust, scalable, reliable and widely used technologies, e. g., Django, Angular and React. For managing LT service containers, ELG makes use of Knative[4], a layer on top of Kubernetes that handles auto-scaling.

## 2.2 Base Infrastructure

The base infrastructure consists of the nodes running the ELG platform, volume storage, networking facilities and S3-compatible object storage. We use *managed* Kubernetes, i. e., the maintenance and operation of Kubernetes itself is taken care of by the provider. The infrastructure also consists of a large set of Git repositories and Docker registries, hosted in a common group on GitLab[5] for all ELG source and configuration files. Many external registries are used to pull in third-party components, like database servers (MariaDB[6], PostgreSql[7]), authentication and identity management (Keycloak[8]), monitoring (Prometheus[9]), among others. Most LT services offered by the ELG platform are pulled from the Docker registries of their respective developers.

ELG uses a GitOps approach to deployment, with the cluster configuration stored in a dedicated Git repository as a set of Helm charts[10]. A continuous integration pipeline triggers a deployment with each check-in to this repository.

Eventually hosting more than one thousand LT services with different hardware needs, we are unable to keep all of them up concurrently as this would require hundreds of Gigabyte of RAM. KNative is used to automatically scale down services not currently in use to zero replicas. A service is scaled up further if a certain threshold of requests is exceeded. This setup is suitable for services with little traffic. For services intended to power actual applications, however, the time to spin up a container is likely too long. ELG will, later on, offer scaling profiles, which will keep a specific number of replicas online at all times.

Non-functional LT resources uploaded to the platform are made persistent to an S3 compatible object storage and can be downloaded from there.

## 2.3 Catalogue

The metadata records stored in the catalogue enable access to services and data resources. They are described using the ELG metadata schema (Labropoulou et al., 2020) and can be browsed and explored. The catalogue also includes a registry of stakeholders who develop LT services or products, and relevant projects, thus providing an overview of the whole European LT landscape. The ELG metadata schema builds upon, consolidates and updates the META-SHARE schema (Gavrilidou et al., 2012; Piperidis et al., 2018; Labropoulou et al., 2018), taking into account ELG's requirements, recent developments in the metadata domain (e. g., FAIR[11]), and the need for creating a common pool of resources through exchange mechanisms with collaborating initiatives.

The metadata schema caters for the description of the ELG core entities, i. e., Language Technologies (*tools/services*), including functional services and non-functional ones, and Data Language Resources, comprising *data sets* (corpora), *language descriptions* (i. e., models) and *lexical/conceptual resources* (e. g., gazetteers, ontologies, etc.). It also provides for related entities involved in the production, namely *actors* (*organizations*, *groups* and *persons*), *projects*, *documents*, and *licences/terms of use*. Metadata records are created by providers using the online editor (Section 2.6.1), or from other sources through harvesting and conversion APIs (Section 2.5), gradually enriched through (semi-)automatic processes and curated by persons who rightfully claim them.

## 2.4 Functional Services

The European LT landscape is broad and varied, with many providers of different classes of services and tools, exposed through different APIs and data formats. We attempt to bring more order to this varied landscape by identifying classes of related services, and providing a generic API for each class. So far, we have identified three classes. (1) *Machine Translation (MT)* services take text in one language and translate it into text in another language, possibly with additional metadata associated with each segment. (2) *Information Extraction (IE)* services take text and annotate it with metadata on specific segments. This class can cover a wide variety of services from basic NER through to complex sentiment analysis

---

[4]https://knative.dev
[5]https://gitlab.com/european-language-grid/
[6]https://mariadb.org
[7]https://www.postgresql.org
[8]https://www.keycloak.org
[9]https://prometheus.io
[10]https://helm.sh

[11]https://www.force11.org/group/fairgroup/fairprinciples

and domain-specific tools. (3) *Automatic Speech Recognition (ASR)* services take audio as input and produce text (e. g., a transcription) as output, possibly with metadata associated with each segment.

| | A | B | C | D |
|---|---|---|---|---|
| **ASR** | | | | |
| Speech Recognition | 12 | 3 | 9 | |
| **IE & Text Analysis** | | | | |
| Dependency Parsing | 24 | 7 | 13 | |
| Lemmatisation | 24 | 7 | 13 | |
| Morphological analyser | 24 | 7 | 13 | |
| Part of Speech tagging | 24 | 7 | 13 | |
| Tokenization | 24 | 7 | 13 | |
| Language identification | 22 | 6 | 14 | 13 |
| Named Entity Recognition | 16 | 5 | 11 | |
| Keyword extraction | 9 | 3 | 9 | |
| Sentiment Analysis | 8 | | 4 | |
| Key phrase Extraction | 7 | | 5 | |
| Polarity detection | 7 | | 4 | |
| Summarization | 7 | | 5 | |
| Other services (not shown here) | 30 | | | |
| **MT** (Source ↓ / Target →) | | | | |
| A | 30 | | 2 | 1 |
| C | 1 | | | |
| **Other** | | | | |
| Text to Speech | 7 | 1 | 1 | 2 |

Table 1: Language coverage per category of the services to integrate in ELG

Other clusters are emerging as we are preparing more services for integration, e. g., text-to-speech and text classification. Our goal is to provide services of all classes for all official EU languages and for other EU and non-EU languages that are of social or strategic interest in the EU. Table 1 shows the overall language coverage of each category of services across all consortium partners; languages have been divided into four groups: (A) EU official languages; (B) other EU languages without official status, plus languages from candidate countries and free trade partners; (C) languages spoken by immigrants or important trade and political partners; (D) languages that do not fit (A), (B), (C).

Release 1 of the platform (April 2020) targeted the languages spoken in the countries of the ELG consortium, with 141 IE and text analysis services, 24 MT, nine ASR, four TTS and two text categorisation services. Further services are being added on a regular basis with 200+ additional IE and text analysis services, 21 MT, eight ASR and nine TTS scheduled to be included by the time of ELG Release 2 in February 2021.

We aim to make it as simple as possible for LT providers to integrate their services, but in a way that avoids the proliferation of incompatible APIs for the same task, allowing users to access the widest range of services without being locked in to a single vendor. Our generic APIs use HTTP as the transport protocol and specific schemas of JSON-based messages as the payload. Providers who want to integrate their services into the ELG need to provide a Docker image that presents an HTTP endpoint that can receive requests and return responses in the specified format (user authentication, authorisation, etc. are handled by the platform). Once a service is integrated, it can be used via the public APIs and UIs (Section 2.6).

## 2.5 Data Sets and Language Resources

Already now ELG provides access to more than 2700 language resources. We ingested substantial resources from existing repositories, especially ELDA/ELRA, ELRC-SHARE (Lösch et al., 2018; Piperidis et al., 2018; Smal et al., 2020) and META-SHARE (Piperidis, 2012; Piperidis et al., 2014). We have also been working on 'external' repositories, about 220 of which have been identified so far. Some (e. g., Zenodo, Quantum Stat) are already being ingested together with two repositories related to ELG, LINDAT/CLARIAH-CZ and ELRA-SHARE-LRs (LRs published at LREC).

## 2.6 Access Methods and User Interfaces

Our main groups of users are: (1) LT/LR providers – companies or research organisations with tools, services or data that can be provided through the ELG; (2) Developers and integrators – companies and research institutions interested in using LT; (3) General LT information seekers; (4) Stakeholders who wish to provide information about events etc.; (5) Casual visitors. We provide three ways of access: REST APIs, web UIs, Python package.

### 2.6.1 REST APIs

The ELG exposes several REST APIs, which are used by all clients. They are exposed for (1) browsing and searching the catalogue, (2) creating, updating and retrieving metadata records, (3) executing services, (4) downloading resources. Authentication is performed through OAuth2 (OpenID-Connect) using JSON Web Tokens.

The catalogue API is based on a JSON serialisation of the metadata schema. The entry point is the search operation, which supports free text search as well as faceted browsing. The metadata record creation, update and retrieval API is controlled by

the catalogue module and associates each record with a creator and curator. The curator can edit and update the record until it is published.

The functional service API (internal LT API) provides a way of executing any functional service deployed in the ELG. *All* functional services of a given class (MT, ASR, etc.) are presented under a common API for that class, allowing the user to choose the best service for their requirements without being locked in to a single vendor.[12] The public-facing LT service API mirrors the internal LT service provider API (see above), being based around the same JSON message formats, but also offers simplified options. It is possible to HTTP POST plain text to an MT service, or binary audio to an ASR service, without having to wrap it in the full JSON envelope or multi-part MIME structure used by the internal API. Since the public and internal APIs are conceptually distinct, we can add and offer public APIs that use other technologies (e. g., gRPC). The LT Service Execution Server component translates requests between the public and internal APIs. An asynchronous interaction style is offered for services that require a longer run time to process a request, this works by returning an immediate response that directs the caller to another URL, which it can then poll to request the result.

### 2.6.2 Web Interface (GUI)

Angular 9.0 and Typescript were adopted for developing the Drupal CMS front-end which is used for presenting content such as news or conferences. For the catalogue UI we use React. Currently, both web applications (CMS and catalogue frontend) use client-side-rendering, i. e., they deliver a single HTML file, the rest of the application comes as Javascript files. User authorisation is ensured by adding a JSON Web Token (JWT) to data requests, where the user identity data is encoded and sent as an encrypted JSON object.

For LT services the catalogue record detail page includes a trial GUI, allowing users to experiment with the service in the browser. Generic trial UIs have been developed for the principal service types (ASR, MT, TTS, text annotation and classification services) but LT service providers can also supply their own GUI if the standard ones are not suitable. An example is the family of UDPipe dependency

```python
from elg import Catalog, Service
# Search for cz/en MT services
catalog = Catalog()
results = catalog.search(
    function='Machine Translation',
    languages=['en', 'cz'])
print(results)
# Create Service object from first result
service = Service.from_entity(results[0])
# Translate plain text input from English to Czech
translated = service(
    "Did Nikola Tesla live in Berlin?")
print(translated)
```

Figure 3: Python Client Package – code example

parser services, where the provider has created a custom UI to visualise dependency graphs.[13]

The web GUI also includes a metadata editor that supports different entities (LTs, organisations, etc.). It provides validation rules, lookup mechanisms that use values from previously filled-in metadata elements and an online help.

### 2.6.3 Python Client Package

The Python Client Package, available through the Python package manager pip[14], comprises a command line interface and utility scripts for querying the ELG catalogue and executing ELG-hosted services via REST API calls. For features that require authentication, e. g., calling services, the client prompts the user to enter a token which is received after successful authentication in a browser window (Figure 3). This simplifies the integration of ELG-hosted services into Python projects.

### 2.7 Contribution of Services and Resources

We want to enable commercial and non-commercial providers to adapt their LT services so that they can be integrated into the ELG and also to make this ingestion as simple as possible. Currently, the process consists of six steps: (1) adapt the service to fit the ELG API; (2) create a Docker image; (3) push the image into a Docker registry; (4) deploy the service by creating a Kubernetes configuration file; (5) create an ELG provider account; (6) register the service by creating a metadata record. For some of the ELG services, the integration took a few days, for others only a few hours. This effort was recently brought further down by adding Docker templates for the most common cases and introducing the metadata editor. Two alternative

---

[12]While workflows that consist of multiple services are currently not addressed by ELG, we do experiment with workflow composition and platform interoperability (Rehm et al., 2020b,c; Moreno-Schneider et al., 2020a,b).

[13]Trial UIs can include third-party code. They are sandboxed using an iframe and configured via JavaScript message passing.

[14]https://pypi.org/project/elg/

ways of integrating a service exist. It is possible to package the LT tool in a container that does not implement the ELG LT service API. In this case, a second container is required as an adapter, which implements the ELG LT service API and communicates with the LT tool container. It is also possible to run an LT service outside the cluster: here, a proxy container that implements the ELG LT service API is required and deployed in the cluster for accessing the external service. Libraries are available that produce skeleton code.

## 2.8 Key Stakeholders

The ELG is meant to be a joint umbrella platform for the whole European LT landscape including industry and research. ELG caters for *commercial LT providers* who want to showcase their products, services and their organisation. We want to provide *the* marketplace for European LT, which requires coverage of, ideally, all European provider companies. In December 2020 we populated the ELG catalogue with a list of 900 LT companies. Representatives of these organisations can claim (or delete) their record and take over maintenance of their ELG page, including upload of services or data sets. *Research centres and universities* are also LT providers but their interest is research-driven, providing data sets and experimental software. *LT users* are, e. g., organisations who want to make use of LT. They interact with the ELG in the role of a consumer or potential customer. ELG also collaborates with a number of EU-funded projects and initiatives (Rehm et al., 2020c,d) and set up a network of 32 National Competence Centres (NCCs), which function as bridges between the national and regional communities and the ELG.

## 2.9 Open Calls: Pilot Projects

ELG provides approx. 30% of its project budget to a number of pilot projects. The pilots either broaden ELG's portfolio (by developing services or resources), or demonstrate the ELG's usefulness. Financial support is awarded following an open, transparent and expert-driven evaluation process. The first call was published in March 2020, the second one in October 2020. The first set of projects started in July 2020, the second set starts in February 2021 with a duration of 9-12 months. In the first call, 110 proposals were accepted for evaluation with applicants from 29 countries. We received more proposals from SMEs (62) than research organisations (48). While 79 proposals fo-

cused on contributing services or resources, 31 proposals concentrated on developing applications using the ELG. We selected ten projects for funding, amounting to a sum of 1,363,915€ in total.[15] We received a total of 106 proposals to the second call with applicants from 28 countries. Again, we had more proposals from SMEs (61) than from research organisations (45). In February 2021, five projects were selected for funding.

## 2.10 Legal Entity

We will establish a not-for-profit legal entity in early 2022, which will take over operation of the ELG platform after the end of the current EU project (June 2022). The long-term operational model is currently under development.

## 3 Related Work

ELG builds upon previous work of the ELG consortium and the wider European LT community, especially META-NET/META and ELRC.

In addition, we have collected more than 30 platforms, projects or initiatives that can be considered relevant for ELG including, among others, UIMA (Ferrucci and Lally, 2003), CLARIN (Hinrichs and Krauwer, 2014), DKPro (Gurevych et al., 2007); Rehm et al. (2020a) provide an exhaustive comparison. They share at least one of the following goals with ELG, i. e., they provide: 1) a collection of LT/NLP tools or data sets; 2) a platform, which harvests metadata records from distributed sources, 3) a platform for the sharing of tools or data sets. While related projects do exist, the approach of ELG is unique. The platform that most closely resembles ELG is the National Platform for LT, operated by the Ministry of Electronics and Information Technology in India.[16]

Several global technology enterprises offer LT services. Among these are Amazon Comprehend[17] and Microsoft Azure Cognitive Services (Del Sole, 2018). Furthermore, Google recently (Sept. 2018) released a search platform for data sets.[18] Intento[19] offers commercial LT services from different providers for selected tasks.

---

[15]https://www.european-language-grid.eu/open-calls/
[16]https://nplt.in
[17]https://aws.amazon.com/en/comprehend/
[18]https://toolbox.google.com/datasetsearch
[19]https://inten.to

## 4 Conclusions and Future Work

It has been argued that Europe should not outsource its multilingual communication and language infrastructure to other continents since the European demands are unique and complex (Rehm and Uszkoreit, 2013; Rehm, 2017; Rehm et al., 2020d). Instead, Europe should make use of and support its own LT community. One of the obstacles to overcome is the creation of a joint technology platform. The ELG will foster LTs *for Europe* built *in Europe*. In its first two years, the ELG project has seen the demo of the MVP in October 2019, Release 1 in early 2020 and two successfully completed open calls for pilot projects. We have been improving and extending the platform and continuously added services and data sets. While Release 2 of the platform will follow in March 2021, Release 3 is foreseen for early 2022.

## Acknowledgments

## References

Alessandro Del Sole. 2018. Introducing Microsoft Cognitive Services. In *Microsoft Computer Vision APIs Distilled*, pages 1–4. Springer.

ELRC. 2019. ELRC White Paper: Sustainable Language Data Sharing to Support Language Equality in Multilingual Europe. https://lr-coordination.eu/sites/default/files/Documents/ELRCWhitePaper.pdf. European Language Resource Coordination (ELRC), Second online edition.

European Parliament. 2018. Report on language equality in the digital age. http://www.europarl.europa.eu/doceo/document/A-8-2018-0228_EN.html. (2018/2028(INI)). Committee on Culture and Education (CULT), Committee on Industry, Research and Energy (ITRE); Rapporteur: Jill Evans.

David Ferrucci and Adam Lally. 2003. Accelerating Corporate Research in the Development, Application, and Deployment of Human Language Technologies. In *Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*, pages 67–74. ACL.

Maria Gavrilidou, Penny Labropoulou, Elina Desipri, Stelios Piperidis, Haris Papageorgiou, Monica Monachini, Francesca Frontini, Thierry Declerck, Gil Francopoulo, Victoria Arranz, and Valerie Mapelli. 2012. The META-SHARE Metadata Schema for the Description of Language Resources. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*. European Language Resources Association (ELRA).

Iryna Gurevych, Max Mühlhäuser, Christof Müller, Jürgen Steimle, Markus Weimer, and Torsten Zesch. 2007. Darmstadt Knowledge Processing Repository based on UIMA. In *Proceedings of the First Workshop on Unstructured Information Management Architecture at Biannual Conference of the Society for Computational Linguistics and Language Technology, Tübingen, Germany*, page 89.

Erhard Hinrichs and Steven Krauwer. 2014. The CLARIN Research Infrastructure: Resources and Tools for eHumanities Scholars. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 1525–1531.

Andras Kornai. 2013. Digital Language Death. *PLoS ONE*, 8(10). https://doi.org/10.1371/journal.pone.0077056.

Penny Labropoulou, Dimitris Galanis, Antonis Lempesis, Mark Greenwood, Petr Knoth, Richard Eckart de Castilho, Stavros Sachtouris, Byron Georgantopoulos, Stefania Martziou, Lucas Anastasiou, Katerina Gkirtzou, Natalia Manola, and Stelios Piperidis. 2018. OpenMinTeD: A Platform Facilitating Text Mining of Scholarly Content. In *WOSP 2018 Workshop Proceedings, Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 7–12, Miyazaki, Japan. European Language Resources Association (ELRA).

Penny Labropoulou, Katerina Gkirtzou, Maria Gavriilidou, Miltos Deligiannis, Dimitris Galanis, Stelios Piperidis, Georg Rehm, Maria Berger, Valérie Mapelli, Michael Rigault, Victoria Arranz, Khalid Choukri, Gerhard Backfried, José Manuel Gómez Pérez, and Andres Garcia-Silva. 2020. Making Metadata Fit for Next Generation Language Technology Platforms: The Metadata Schema of the European Language Grid. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 3421–3430, Marseille, France. European Language Resources Association (ELRA).

Andrea Lösch, Valérie Mapelli, Stelios Piperidis, Andrejs Vasiļjevs, Lilli Smal, Thierry Declerck, Eileen

Schnur, Khalid Choukri, and Josef Van Genabith. 2018. European Language Resource Coordination: Collecting Language Resources for Public Sector Multilingual Information Management. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Julián Moreno-Schneider, Peter Bourgonje, Florian Kintzel, and Georg Rehm. 2020a. A Workflow Manager for Complex NLP and Content Curation Workflows . In *1st International Workshop on Language Technology Platforms (IWLTP 2020)*, Marseille. Submitted to IWLTP 2020.

Julián Moreno-Schneider, Georg Rehm, Elena Montiel-Ponsoda, Víctor Rodriguez-Doncel, Artem Revenko, Sotirios Karampatakis, Maria Khvalchik, Christian Sageder, Jorge Gracia, and Filippo Maganza. 2020b. Orchestrating NLP Services for the Legal Domain. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France. European Language Resources Association (ELRA). Accepted for publication.

Stelios Piperidis. 2012. The META-SHARE Language Resources Sharing Infrastructure: Principles, Challenges, Solutions. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Stelios Piperidis, Penny Labropoulou, Miltos Deligiannis, and Maria Giagkou. 2018. Managing Public Sector Data for Multilingual Applications Development. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Stelios Piperidis, Harris Papageorgiou, Christian Spurk, Georg Rehm, Khalid Choukri, Olivier Hamon, Nicoletta Calzolari, Riccardo Del Gratta, Bernardo Magnini, and Christian Girardi. 2014. META-SHARE: One Year After. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Georg Rehm, editor. 2017. *Language Technologies for Multilingual Europe: Towards a Human Language Project. Strategic Research and Innovation Agenda*. CRACKER and Cracking the Language Barrier federation. Version 1.0. Unveiled at META-FORUM 2017 in Brussels, Belgium, on November 13/14, 2017. Prepared by the Cracking the Language Barrier federation, supported by the EU-funded project CRACKER.

Georg Rehm, Maria Berger, Ela Elsholz, Stefanie Hegele, Florian Kintzel, Katrin Marheinecke, Stelios Piperidis, Miltos Deligiannis, Dimitris Galanis, Katerina Gkirtzou, Penny Labropoulou, Kalina

Bontcheva, David Jones, Ian Roberts, Jan Hajic, Jana Hamrlová, Lukáš Kačena, Khalid Choukri, Victoria Arranz, Andrejs Vasiļjevs, Orians Anvari, Andis Lagzdiņš, Jūlija Meļņika, Gerhard Backfried, Erinç Dikici, Miroslav Janosik, Katja Prinz, Christoph Prinz, Severin Stampler, Dorothea Thomas-Aniola, José Manuel Gómez Pérez, Andres Garcia Silva, Christian Berrío, Ulrich Germann, Steve Renals, and Ondrej Klejch. 2020a. European Language Grid: An Overview. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 3359–3373, Marseille, France. European Language Resources Association (ELRA).

Georg Rehm, Kalina Bontcheva, Khalid Choukri, Jan Hajic, Stelios Piperidis, and Andrejs Vasiljevs, editors. 2020b. *Proceedings of the 1st International Workshop on Language Technology Platforms*. Marseille, France. 16 May 2020, https://www.european-language-grid.eu/iwltp-2020/.

Georg Rehm, Dimitrios Galanis, Penny Labropoulou, Stelios Piperidis, Martin Welß, Ricardo Usbeck, Joachim Köhler, Miltos Deligiannis, Katerina Gkirtzou, Johannes Fischer, Christian Chiarcos, Nils Feldhus, Julián Moreno-Schneider, Florian Kintzel, Elena Montiel, Víctor Rodríguez Doncel, John P. McCrae, David Laqua, Irina Patricia Theile, Christian Dittmar, Kalina Bontcheva, Ian Roberts, Andrejs Vasiljevs, and Andis Lagzdiņš. 2020c. Towards an Interoperable Ecosystem of AI and LT Platforms: A Roadmap for the Implementation of Different Levels of Interoperability. In *Proceedings of the 1st International Workshop on Language Technology Platforms (IWLTP 2020, co-located with LREC 2020)*, pages 96–107, Marseille, France. 16 May 2020.

Georg Rehm, Jan Hajic, Josef van Genabith, and Andrejs Vasiļjevs. 2016a. Fostering the Next Generation of European Language Technology: Recent Developments – Emerging Initiatives – Challenges and Opportunities. In *Proceedings of the 10th Language Resources and Evaluation Conference (LREC 2016)*, pages 1586–1592, Portorož, Slovenia. European Language Resources Association (ELRA).

Georg Rehm and Stefanie Hegele. 2018. Language Technology for Multilingual Europe: An Analysis of a Large-Scale Survey regarding Challenges, Demands, Gaps and Needs. In *Proceedings of the 11th Language Resources and Evaluation Conference (LREC 2018)*, pages 3282–3289, Miyazaki, Japan. European Language Resources Association (ELRA).

Georg Rehm, Katrin Marheinecke, Stefanie Hegele, Stelios Piperidis, Kalina Bontcheva, Jan Hajic, Khalid Choukri, Andrejs Vasiļjevs, Gerhard Backfried, Christoph Prinz, José Manuel Gómez Pérez, Luc Meertens, Paul Lukowicz, Josef van Genabith, Andrea Lösch, Philipp Slusallek, Morten Irgens, Patrick Gatellier, Joachim Köhler, Laure Le Bars, Dimitra Anastasiou, Albina Auksoriūtė, Núria Bel, António Branco, Gerhard Budin, Walter Daelemans,

Koenraad De Smedt, Radovan Garabík, Maria Gavri-ilidou, Dagmar Gromann, Svetla Koeva, Simon Krek, Cvetana Krstev, Krister Lindén, Bernardo Magnini, Jan Odijk, Maciej Ogrodniczuk, Eiríkur Rögnvaldsson, Mike Rosner, Bolette Pedersen, Inguna Skadina, Marko Tadić, Dan Tufiş, Tamás Váradi, Kadri Vider, Andy Way, and François Yvon. 2020d. The European Language Technology Landscape in 2020: Language-Centric and Human-Centric AI for Cross-Cultural Communication in Multilingual Europe. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 3315–3325, Marseille, France. European Language Resources Association (ELRA).

Georg Rehm and Hans Uszkoreit, editors. 2012. *META-NET White Paper Series "Europe's Languages in the Digital Age"*. Springer, Heidelberg, New York, Dordrecht, London. 31 volumes on 30 European languages. http://www.meta-net.eu/whitepapers.

Georg Rehm and Hans Uszkoreit, editors. 2013. *The META-NET Strategic Research Agenda for Multilingual Europe 2020*. Springer, Heidelberg, New York, Dordrecht, London.

Georg Rehm, Hans Uszkoreit, Sophia Ananiadou, Núria Bel, Audronė Bielevičienė, Lars Borin, António Branco, Gerhard Budin, Nicoletta Calzolari, Walter Daelemans, Radovan Garabík, Marko Grobelnik, Carmen García-Mateo, Josef van Genabith, Jan Hajič, Inma Hernáez, John Judge, Svetla Koeva, Simon Krek, Cvetana Krstev, Krister Lindén, Bernardo Magnini, Joseph Mariani, John McNaught, Maite Melero, Monica Monachini, Asunción Moreno, Jan Odjik, Maciej Ogrodniczuk, Piotr Pęzik, Stelios Piperidis, Adam Przepiórkowski, Eiríkur Rögnvaldsson, Mike Rosner, Bolette Sandford Pedersen, Inguna Skadiņa, Koenraad De Smedt, Marko Tadić, Paul Thompson, Dan Tufis, Tamás Váradi, Andrejs Vasiļjevs, Kadri Vider, and Jolanta Zabarskaite. 2016b. The Strategic Impact of META-NET on the Regional, National and International Level. *Language Resources and Evaluation*, 50(2):351–374. 10.1007/s10579-015-9333-4.

Georg Rehm, Hans Uszkoreit, Ido Dagan, Vartkes Goetcherian, Mehmet Ugur Dogan, Coskun Mermer, Tamás Váradi, Sabine Kirchmeier-Andersen, Gerhard Stickel, Meirion Prys Jones, Stefan Oeter, and Sigve Gramstad. 2014. An Update and Extension of the META-NET Study "Europe's Languages in the Digital Age". In *Proceedings of the Workshop on Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era (CCURL 2014)*, pages 30–37, Reykjavik, Iceland.

Lilli Smal, Andrea Lösch, Josef van Genabith, Maria Giagkou, Thierry Declerck, and Stephan Busemann. 2020. Language Data Sharing in European Public Services – Overcoming Obstacles and Creating Sustainable Data Sharing Infrastructures. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, page 3443–3448, Marseille, France. European Language Resources Association (ELRA).

STOA. 2017. Language equality in the digital age – Towards a Human Language Project. STOA study (PE 598.621), IP/G/STOA/FWC/2013-001/Lot4/C2, March 2017. Carried out by Iclaves SL (Spain) at the request of the Science and Technology Options Assessment (STOA) Panel, managed by the Scientific Foresight Unit (STOA), within the Directorate-General for Parliamentary Research Services (DG EPRS) of the European Parliament. http://www.europarl.europa.eu/stoa/.

Andrejs Vasiljevs, Khalid Choukri, Luc Meertens, and Stefania Aguzzi. 2019. Final study report on CEF Automated Translation value proposition in the context of the European LT market/ecosystem. DOI 10.2759/142151. A study prepared for the European Commission, DG Communications Networks, Content & Technology by Crosslang, Tilde, ELDA, IDC.
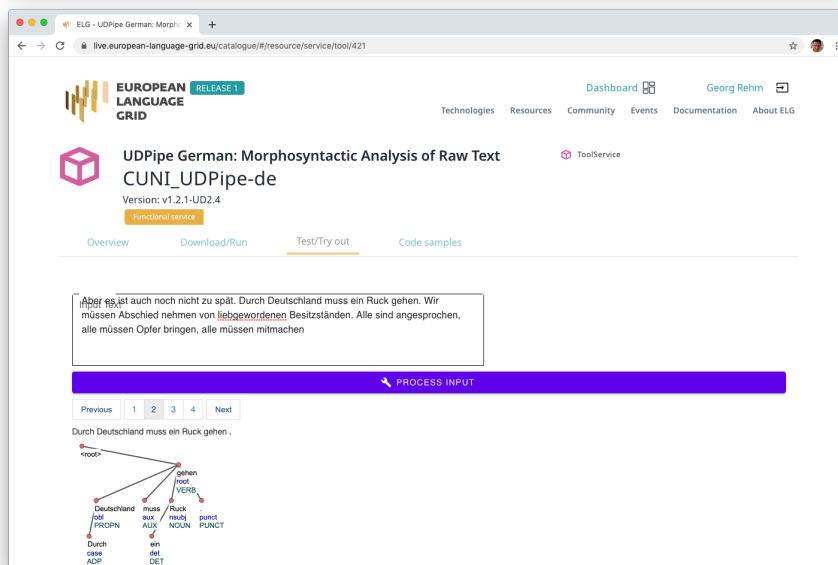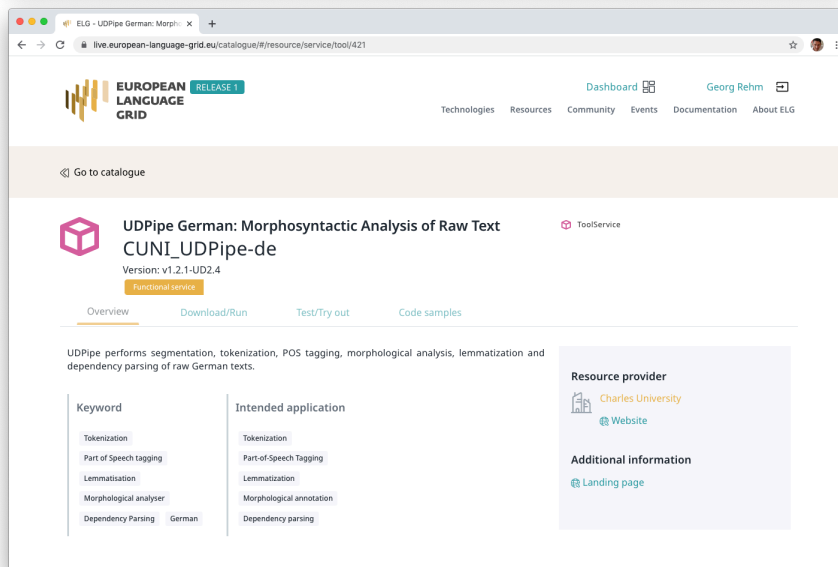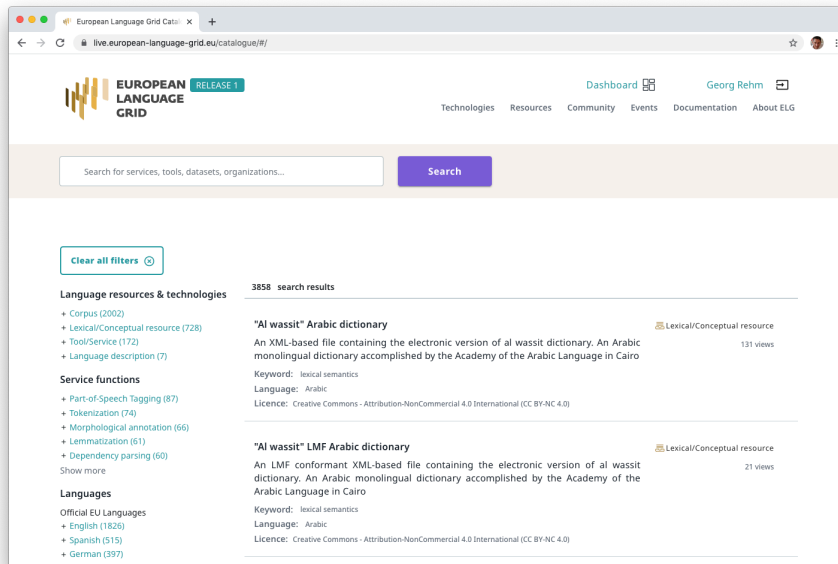
# A  Selected Screenshots

Figure 4: Selected screenshots of the European Language Grid (ELG)