Proceedings of the 1st International Workshop on Language Technology Platforms (IWLTP 2020), pages 81–88
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

A Processing Platform Relating Data and Tools for Romanian Language

Vasile Păiş, Radu Ion, Dan Tufiş
Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy
CASA ACADEMIEI, 13 “Calea 13 Septembrie”, Bucharest 050711, ROMANIA
{vasile, radu, tufis}@racai.ro

Abstract

This paper presents RELATE (http://relate.racai.ro), a high-performance natural language platform designed for Romanian language. It is meant both for demonstration of available services, from text-span annotations to syntactic dependency trees as well as playing or automatically synthesizing Romanian words, and for the development of new, annotated corpora. It also incorporates the search engines for the large CoRoLa reference corpus of contemporary Romanian and the Romanian wordnet. It integrates multiple text and speech processing modules and exposes their functionality through a web interface designed for the linguist researcher. It makes use of a scheduler-runner architecture, allowing processing to be distributed across multiple computing nodes. A series of input/output converters allows large corpora to be loaded, processed and exported according to user preferences.

Keywords: natural language processing, web platform, Romanian language processing

1. Introduction

Today’s natural language processing challenges require the use of very complex pipelines applied on huge datasets. In this context, existing pipelines must be integrated and adapted for usage inside high performance environments such as clusters, grids or even in the cloud. The entire flow needs to be supervised and resume mechanisms must be in place in order to recover processing in case of unforeseen hardware or software errors.

Even though existing Romanian language resources are an order of magnitude less than those existing for English language, several new large data sets become available each year. For each new project that we are involved in, we are faced with processing hundreds of thousands of text files, in the several gigabytes range. Due to large sizes involved, combined with the pipeline’s complexity, this usually implies many days of processing time. Thus, the ability to distribute processing across multiple computing nodes becomes a necessity in order to reduce the required processing time. Furthermore, in order to allow scientists to focus on their research and not on technical issues, a user-friendly interface was needed, allowing easy interaction with the system.

RELATE is a Romanian language technology platform developed at the Institute for Artificial Intelligence of the Romanian Academy, integrating different state-of-the art tools and algorithms for processing Romanian language, developed either in-house or by our partners in different research projects. It evolved from our previous TEPROLIN platform (Ion, 2018) from a demonstrative, single file multi-level processing pipeline, to a more complex platform allowing for user-friendly interaction with Romanian language technologies as well as storage, processing, visualizing and downloading of large sets of annotated data. It was constructed using a task-based approach, where the user can load a corpus (usually as an archive) then start a number of annotation tasks and finally export the resulting data. The platform hides the complexities of distributing the load across the available processing nodes, waiting for data to be processed, error recovery and final gathering of results. Instead, the user is presented with an easy to use web interface where she/he can interact with the already annotated files and see the status of the entire annotation process. RELATE was constructed with the goal of making it accessible to at least two types of researchers: 1) theoretical linguists, Romanian language teachers and anyone interested in studying Romanian language by providing a nice visualization of the automatic analysis for any Romanian sentence and 2) NLP researchers wishing to either have access to off-the-shelf Romanian annotators or evaluating Romanian language technologies.

2. Related work

Speaking of language resource *inventories and search engines,* META-SHARE[1] (Federmann et al., 2012) together with CLARIN[2] are the biggest, publicly available European websites for research and development in the field. ELRC-SHARE[3] (European Language Resource Coordination Share) is another website dedicated to European language resources, specifically for machine translation. Both ELRC-Share and META-SHARE offer search boxes through which one can easily find various language resources (language tool, annotated, text or audio corpora, etc.) for any (European) language. Beside language resources for Romanian, our language of interest, there are *complex processing pipelines* such as NLP-Cube (Boroş et al., 2018) or TTL (Ion, 2007) that are able to do tokenization, POS tagging, lemmatization, chunking and dependency parsing. To use them, one has to be tech savvy, know Python 3 or Perl programming and be comfortable installing required open-source libraries (actually, this is the story of any open-source language technology tool, thus limiting its use to those that possess the knowledge to take the required steps).

To make the composition of the language processing chains more user-friendly, GATE (Cunningham, 2002) and TextFlows (Perovšek et al., 2016) allow for dragging and dropping text processing widgets into a graphical processing workflow to create the processing pipelines that the likes of NLP-Cube and TTL require computer programming to achieve. While graphically composing language processing chains is a big step towards the usability of the respective language technologies, their output is not enhanced with specialized visualization tools that allow access into the computational resources used for annotation.

[1] http://www.meta-share.org
[2] https://www.clarin.eu
[3] https://elrc-share.eu/

RELATE aims specifically at doing automatic text processing, with annotations at multiple levels, along with annotation visualization and expansion into the corresponding linguistic computational resources. Compared to other platforms, such as (Wanxiang et al., 2010), our platform does not focus on exposing APIs, even though such text processing APIs do exist, either directly from the different components or as an indirect result of integrating several components. Instead, RELATE is designed to be an integrated environment accessible via the web interface. In some ways it is similar to (Morton and LaCivita, 2003) work, with the addition of the web interface and parallel processing capabilities. Currently, the RELATE platform does not contain yet any functionality for automatic training of new models, such as more recent platforms like (Gardner et al., 2018). Furthermore, compared to the WebLicht (Hinrichs et al, 2010) platform, developed within the CLARIN project, RELATE is focused on Romanian language tools. Even more, besides integrating tagging capabilities, the platform also integrates other tools, such as WordNet, translation, speech recognition and synthesis.

The processing workflow is guided via addition of tasks which, by design, can work with the internal format produced by any other tasks. Thus, no workflow editor, such as the one used in (Perovšek et al., 2016), was needed at the moment. Tasks can be chained together, one after another, without the need for complex "wiring".

3. Platform Architecture

RELATE has two main areas (see Figure 1): a public area and a private area. The public area allows running most of the annotation tasks as well as exploring other platform features without any data storage facilities. Therefore, this is intended either for familiarizing a user with the platform or for small scale annotations (like single sentences or small files which do not require long term storage in the platform). The private part requires a user name and password[4] to be provided for user authentication and allows access to all platform features, including annotation of large corpora and storage of both raw and annotated data.

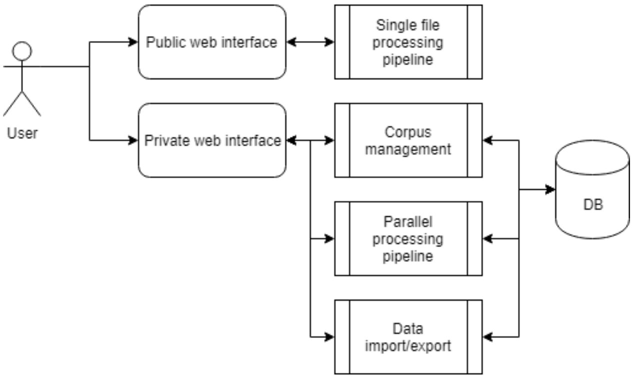


Figure 1: RELATE public and private areas

3.1 Platform components

The RELATE platform was constructed using an approach based on multiple interconnected layers. From the user's perspective, the first layer is the web front-end. It is in charge of displaying data to the user and employs visualizations such as: text views (for displaying raw text files as well as annotated files if the user opts for a text like visualization), data grids (for visualizing table information, such as annotation results in different formats), tree-views (useful for displaying dependency parsing information), integration of Brat rapid annotation tool (Stenetorp et al., 2012) for named entity visualization. Furthermore, the visualization layer interconnects with visualizations made available from other projects, such as the interrogation tools from the Reference Corpus for Contemporary Romanian Language (CoRoLa) (Mititelu et al., 2018).
The second layer of the platform is the back-end layer. This is in charge of orchestrating user requests between the various integrated modules. In turn, this happens either via an ephemerous flow, with results communicated directly to the web front-end, or via the task system with final storage in the platform's file system. The multi-layer architecture is presented in Figure 2.

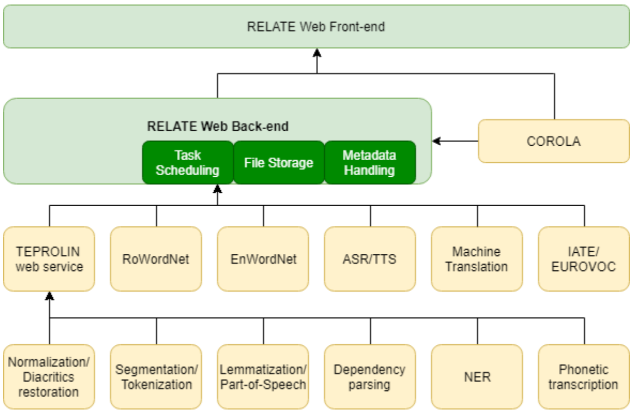


Figure 2: RELATE multiple layers architecture

Components integrated in the RELATE platform are written in different programming languages, such as: C/C++, Java, Python, scripts (bash, php). Furthermore, most of them were not exposing any web API and the few who had such an API available used completely different invocation flows. This created serious integration challenges, as described in more detail in (Păiș et al., 2019). Basically, we had to either create a web API wrapper for the tools or execute them as separate processes and collect the produced temporary files. In order to guarantee a uniform interrogation for multiple, related, modules, we used the TEPROLIN web service which integrates modules written in Python and other programming languages and exposes them in the same web API. This is in turn consumed by the back-end layer modules. Different modules are integrated either for textual annotation, as detailed below in the "Available annotations" sub-section, or only for enhancing the user visualization experience and allowing the researcher to make additional enquiries. Such is the case for integrating the Romanian WordNet aligned with the English WordNet which allows the user to research cross-lingually various senses of annotated words.

[4] The credentials are provided free of charge by request sent to one of the authors.

3.2 Available annotations

TEPROLIN and its web service[5] interface is a text preprocessing platform for Romanian (Ion, 2018) that currently offers 15 types of text transformation/annotations, from text-span annotations to syntactic dependency trees. Below is a brief account of these modules and corresponding annotations:

1. **Text normalization**: removal of multiple consecutive spaces and Romanian diacritical codes normalization;
2. **Diacritics restoration**: automatic detection of texts lacking Romanian diacritics and automatic diacritic insertion;
3. **Word hyphenation** (Stan et al., 2011);
4. **Word stressed syllable identification** (Stan et al., 2011);
5. **Word phonetic transcription** (Stan et al., 2011) using the SAMPA phonemes for Romanian[6];
6. **Numeral rewriting** (Stan et al., 2011): automatic transformation of number to their written form, useful in text-to-speech synthesis (e.g. 93 → "ninety-three");
7. **Abbreviation rewriting** (Stan et al., 2011): automatic expansion of abbreviations or acronyms to their full form, also useful for text-to-speech synthesis (e.g. art. → "article" or AI → "Artificial Intelligence");
8. **Sentence splitting** (Ion, 2007; Boroş et al., 2018);
9. **Tokenization** (Ion, 2007; Boroş et al., 2018);
10. **POS tagging** (Ion, 2007; Boroş et al., 2018) using the Morpho-Syntactic Descriptors (MSD) for Romanian tag set[7];
11. **Lemmatization** (Ion, 2007; Boroş et al., 2018);
12. **Named entity recognition (NER)** with four labels: person-`PER`, location-`LOC`, organization-`ORG` and time -`TIME` (Păiş 2019);
13. **Biomedical NER** (Boroş et al., 2018) with four labels: disorder (`DISO`), anatomical part (`ANAT`), medical procedure (`PROC`) and chemical (`CHEM`). The sequence labeler was trained on the MoNERo corpus (Mitrofan et al., 2018), (Carp (Mitrofan), 2019);
14. **Chunking** (Ion, 2007) with four types of non-recursive syntactic phrases: noun (`Np`), verb (`Vp`), adjectival/adverbial (`Ap`) and prepositional (`Pp`);
15. **Dependency parsing** (Boroş et al., 2018) with the Romanian Universal Dependencies label set[8].

Each module was adapted and made available for integration as part of the ReTeRom project[9]. Development of individual modules was realized by the ReTeRom partners, as indicated in the references and on the project's website.

TEPROLIN is a Python 3 module that integrates various NLP applications by requiring them to implement the TEPROLIN application programming interface:

- Resource loading, which usually takes from tens of seconds to minutes when the NLP application starts, is only allowed inside a specialized method which is called once when the implementing object is instantiated;
- If the NLP application is not written in Python 3, TEPROLIN expects that the application runs on the same machine as the platform; the communication with the resident process is done via an established inter-process communication mechanism (e.g. sockets or named pipes).

When adding a new NLP application, the software engineer has to insert its name and operations in the TEPROLIN operation graph. Using this graph, TEPROLIN is able to automatically resolve the requirements of the new operation (e.g. before doing POS tagging, the text has to be tokenized first).

Pushing the "DEMO" button in the TEPROLIN Web Service/Complete Flow menu entry will run the full (all 15 operations) processing chain on two sample Romanian sentences. These two sentences were chosen such that every annotation that TEPROLIN is able to give is present and can be visualized. The output of this run can be visualized in computer readable formats: JSON, CoNLL-U[10], CoNLL-X, XML, and as well as graphically: in "Tree" mode (the most informative) and in "Entities" mode where NER annotations can be visualized graphically.

3.3 Task-based processing

In order to achieve better performance by harnessing the CPU resources available on different servers, the RELATE platform uses a task-based scheduler engine which in turn distributes the load across the available computing nodes. Since we targeted a mixed environment, with computing nodes of different sizes and performances, as well as a mixture of operating systems, we decided to develop our own task-engine for the purposes of the platform. It has two components: the scheduler, which is the first to receive a new task and decides where it should be executed, and the task runners which take care of actually running the task and storing final files on the file system.

Each task runner process keeps track of the files already processed so that it can resume processing in case of a system failure. Furthermore, the process is activated via a cron job which ensures automatic restart in case the task runner itself encounters a fatal error. Even more, logging is performed at operating system level ensuring all relevant messages are recorded and available for investigation. However, this is not displayed to the end user, being considered a very technical information, useful for platform developers. Entire processing pipelines are kept in-memory and accessed by task runners via URL endpoints. This ensures the possibility to distribute the tasks on any computing nodes, regardless of their location: same local area network (similar to a cluster environment), multiple networks (a grid environment) or across the Internet (cloud environment). Of course, the location of the computing resources can influence the overall processing time due to the differences in transfer speeds. Nevertheless, in case of large corpora, we

[5] http://relate.racai.ro:5000

[6] https://www.phon.ucl.ac.uk/home/sampa/rom-uni.htm

[7] http://nl.ijs.si/ME/V4/msd/html/msd-ro.html

[8] http://universaldependencies.org/ro/index.html

[9] http://www.racai.ro/p/reterom/index_en.html

[10] https://universaldependencies.org/format.html

consider the parallelization outweighs transfer times, thus reducing the total time required to process the files.

In order to avoid costly synchronization issues that usually occur in distributed systems, the RELATE platform does not make use of any shared resources. The scheduler process allocates disjunct slices of the corpus to each of the task runners. This allows for parallel computation throughout the pipelines without the need to synchronize with other processes.

Finally, the last runner who finishes work related to any particular task will also be in charge of composing the final result if needed. Even though, most of the tasks do not require final assembly of data since each annotation happens on a separate file. The scheduler and runners architecture is presented in Figure 3.

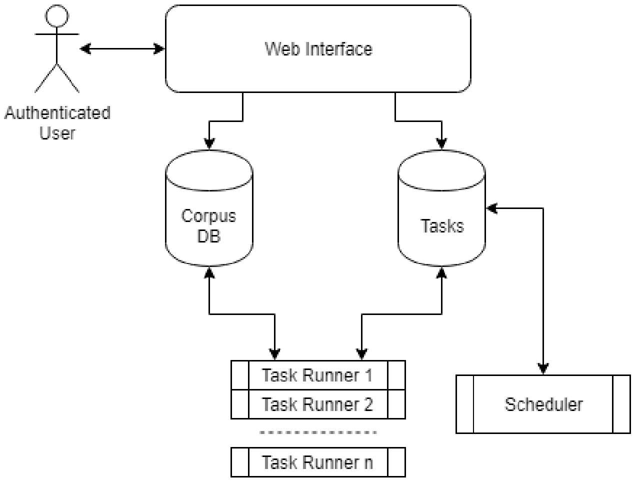


Figure 3: Task execution inside the RELATE platform

3.4 File formats

The platform was designed to allow corpora to be uploaded in the user's format, then processed and annotated to an internal platform format and finally exported to another user specified format. For this purpose, the platform has an import/export interface which can be extended with new functionality to export different user specified formats. Currently, the input format is either raw text files or comma separated files (CSV). In the second case, the user can specify in the interface the column or columns containing text data.

The internal format used throughout the platform is CoNLL-U Plus[11] format. This is a tab separated set of columns, usually considered to be an extension of the CoNLL-U format. In order to allow for a greater compatibility with CoNLL-U aware applications (and users), we have decided to keep the first 10 columns in the order of CoNLL-U specification and extend this with additional information available in the platform, such as named entities, IATE and EUROVOC annotations (Coman et al., 2019).

For output, the platform allows for a number of formats to be used, including: JSON, CoNLL-U (with limited annotations), CoNLL-U Plus variations, XML. In the case of CoNLL-U Plus, one possibility is to export the internal format, containing all the produced annotations, or a subset of those as required for different projects. The actual annotations available in the output file depend on the annotations tasks that were executed.

Since different modules in the pipeline require additional internal formats, other converters are available internally inside the platform, but are not exposed as input/output options.

3.5 Available visualizations

Apart from the annotation options described in section 3.2 above, RELATE integrates several visualization components, allowing the researcher user to better interact with the data. These components can be accessed either directly, via the proper links present in the platform's main menu, or via action buttons made available when interacting with the annotated data.

The "Tree" visualization mode is the most comprehensive of all, displaying generated annotations as well as *on the fly query results of other Romanian computational resources for the selected word*. In other words, the user can *relate* (hence the name of the portal) the output of the automatic language processing chain with information stored in the associated Romanian computational resources, thus seeing if the resource contains (or not) the relevant information and whether this information is useful when studying Romanian or how could it inform other automated Romanian processing algorithms. The "Tree" visualization mode has the dependency tree of a sentence in the center of the frame (one can see individual sentences using the arrows on the left/right of the current sentence). Dependency label names can be seen on the relations. If the user clicks a node in the tree, a panel of information about that word is opened to the right of the dependency tree: search in the CoRoLa corpus, search in the Romanian WordNet, listen to the native pronunciation of the word (if it is stored in the corpus) or synthesizing it (if not existing in the speech corpus), using the SSLA Text-to-Speech module[12] (Boroș et al., 2018b).

Besides linking other Romanian computational language resources and language tools, token annotations can also be inspected in the "Tree" view (e.g. POS tag, lemma, chunk membership, etc.) "Similar Words" will display up to 10 most similar words to the clicked word, computed using word embeddings extracted from the CoRoLa corpus (Păiș and Tufiș, 2018). A lemma with POS version of the similar words list is also available.

Romanian wordnet, RoWordNet, as described in (Tufiș and Mititelu, 2014), is made available for interrogation in the platform, either by itself or aligned with the English wordnet (Miller, 1995). The second option involves searching for a Romanian lemma in the wordnet, seeing the identified synsets and, based on the synset id, the corresponding English information is also displayed.

CoRoLa (Mititelu et al., 2018) which was constructed as a priority project of the Romanian Academy, between 2014 and 2017, contains both written texts and oral recordings. For each of these components, dedicated query interfaces were made available. These were also integrated in the RELATE platform, allowing words to be researched for occurrences in CoRoLa. In the case of written data, interrogation is performed by integration of the KorAP corpus management platform, developed at the Institute for German Language (Leibniz-Institut für Deutsche

[11] https://universaldependencies.org/ext-format.html

[12] http://slp.racai.ro/index.php/ssla/

Sprache) in Mannheim (Bański et al. 2014; Diewald et al. 2016). Similarly, for interrogation of audio transcriptions aligned with voice recordings, the Oral Corpus Query platform (OCQP) (Boroș et al., 2018b) developed for CoRoLa was integrated allowing the user to listen for the pronunciation of different words.

Since only a fraction of Romanian words are available in the audio component of the CoRoLa corpus, two speech synthesis components were integrated in the platform, allowing to user to listen for pronunciation of other words as well. One such system is the Speech Synthesis for Lightweight Applications (SSLA), described in (Boroș and Dumitrescu, 2015). Another, more recent development, is a system derived from our ROBIN project. Furthermore, from the ROBIN project resulted also an automatic speech recognition component which was also integrated in the RELATE platform.

In the case of text automatically recognized from speech, this can be automatically processed through the RELATE platform text annotation components, even though at this moment we lack the integration of an automatic capitalization and punctuation restoration component. Therefore, this particular integration currently has its use only in the case of small sentences.

A machine translation component is also available for interrogation within the RELATE platform. This is derived from the project “CEF Automated Translation toolkit for the Rotating Presidency of the Council of the EU”, TENtec no. 28144308, led by TILDE, a linguistic technology company specializing in neural automatic translation. As part of this project, the translation system (Ro-En and En-Ro)[13] was developed in partnership with the Institute of Research for Artificial Intelligence “Mihai Draganescu” and is available for short translations within the RELATE platform.

Apart from the dedicated components, the platform makes use of advanced data grids whenever such a display option makes sense. For this purpose, we integrated the PqGrid[14] component which allows for features like: maximized view of data grid, column reordering, sorting, searching and integration with JSON based APIs. Furthermore, dependency parsed sentences are displayed in a tree-like visualization which is enhanced with action buttons allowing exploration of words within the other visualization components as detailed above.

3.6 Statistics

For each corpus, a dedicated task can be started for computing corpus statistics. These are computed at various levels: entire corpus, word form, lemma. After being computed, they can be visualized in the RELATE interface or downloaded as CSV files. Similar to other tasks, the statistics task makes use of the parallel runners in order to reduce the overall time required.

Corpus level statistics include: number of raw documents, number of annotated documents, number of sentences, number of tokens, number of “words” (strings separated by space characters), number of lines, number of characters. For each named entity type, the identified number of entities of that type is computed. Similarly, for each universal part of speech tag the corresponding number of occurrences is computed.

Word form (token) statistics include number of tokens, number of unique tokens and for each unique word form the total number of occurrences as well as the total number of files containing the particular word form are computed. Furthermore, the statistics task computes the number of words occurring only once in the entire corpus (also known as “hapax legomena”), the words occurring only two times and the words occurring only three times.

Lemma statistics include number of unique lemmas as well as the number of occurrences for each lemma.

4. Case Study: Annotation of Romanian Legal Corpus

Within the ”Multilingual Resources for CEF.AT in the legal domain” (MARCELL)[15] project, the seven participating teams cooperated in order to produce a comparable corpus aligned at the top-level domains identified by EUROVOC descriptors[16]. For Romanian language, the legal database created includes more than 140K legislative documents issued starting with 1881. These were gathered from the Romanian legislative portal[17] and converted from HTML to raw text format. This resulted in 2.7GB of raw text. During the conversion process certain metadata was also retrieved from within the HTML pages, but only information required for the project’s use cases was stored (such as the publication year of the document).

For upload in the RELATE platform, the raw text was compressed into a zip archive, which had the size of 550Mb. After uploading to the platform, it was automatically decompressed by a task runner and its content was made available through the interface. Following a quick visual inspection to ensure the files were properly imported, an annotation task was launched.

Given the large size of the corpus, the annotation process took about one month on the two physical servers which were made available for project’s purposes. Allocation of text files to pipeline components was orchestrated by the RELATE platform using the scheduler-runners approach described in 2.3 above. During this time, one server restart occurred due to a power outage which demonstrated the platform’s ability to recover in case of unexpected errors and resume annotation. Furthermore, during task running, annotated files started to become available in the interface as they were finished. This allowed the researchers involved in the project to look at the produced annotations and identify potential issues.

Once the basic annotation task ended, a separate, dedicated task was started for IATE[18] and EUROVOC annotations, using the method described in (Coman et al., 2019). This was again orchestrated by the RELATE platform and split across 10 processes which managed to process the entire corpus in less than half hour. Similar to the previous step, annotations were made available in the RELATE web interface and were consulted by the project’s team. Figure 5 shows a data grid visualization of one of the annotated files. This is performed using the CoNLL-U Plus format.

The large difference in the required time for the two annotation tasks is due to the number of processes

[13] https://ro.presidencymt.eu/#/text

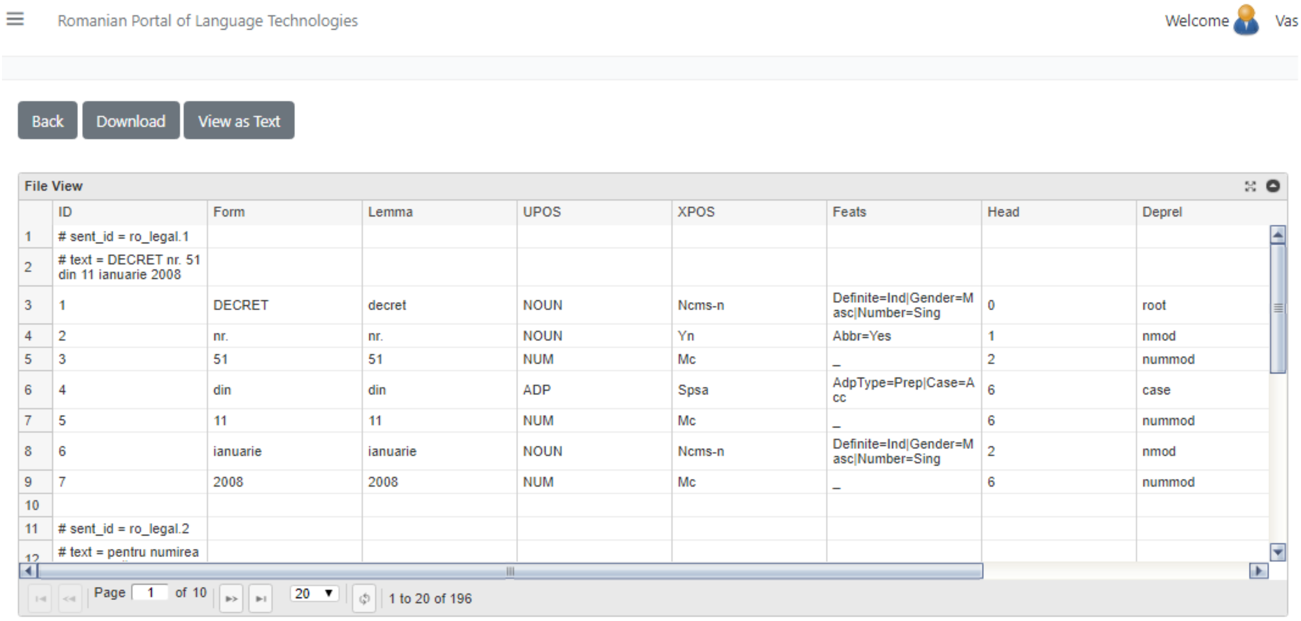
[14] https://paramquery.com/

[15] https://marcell-project.eu/

[16] https://eur-lex.europa.eu/browse/eurovoc.html

[17] http://legislatie.just.ro/

[18] https://iate.europa.eu/home



	ID	Form	Lemma	UPOS	XPOS	Feats	Head	Deprel
1	# sent_id = ro_legal.1							
2	# text = DECRET nr. 51 din 11 ianuarie 2008							
3	1	DECRET	decret	NOUN	Ncms-n	Definite=Ind|Gender=Masc|Number=Sing	0	root
4	2	nr.	nr.	NOUN	Yn	Abbr=Yes	1	nmod
5	3	51	51	NUM	Mc	_	2	nummod
6	4	din	din	ADP	Spsa	AdpType=Prep|Case=Acc	6	case
7	5	11	11	NUM	Mc	_	6	nummod
8	6	ianuarie	ianuarie	NOUN	Ncms-n	Definite=Ind|Gender=Masc|Number=Sing	2	nmod
9	7	2008	2008	NUM	Mc	_	6	nummod
10								
11	# sent_id = ro_legal.2							
12	# text = pentru numirea							

Figure 5: Datagrid visualization of an annotated file from the Romanian legal corpus

involved and their respective complexity. The IATE/EUROVOC annotator used the already tokenized and annotated documents from the previous step. More important, the Aho-Corasick algorithm (Aho and Corasick, 1975) used for detecting the corpus occurrences of the terms stored in the trie dictionary made of IATE Romanian terms runs in linear-time (see details in (Coman et al., 2019)).

Following the two annotation stages, a statistics task was executed, in order to compute the overall statistics on the corpus, useful for reporting purposes. This was executed using 13 processes orchestrated by the platform and took about one hour and half to compute the statistical indicators described in section 2.6 above. Table 1 presents some of the computed statistics.

Number of documents	144,131
Number of tokens	456,079,723
Unique tokens	1,528,228
Unique lemmas	1,195,484
Tokens occurring only once	772,141

Table 1: Statistics from the Romanian legal corpus obtained using the RELATE platform

Finally, a MARCELL specific preparation task was executed, ensuring the output format agreed within the project. This is also a CoNLL-U Plus based format. Each document begins with a line describing the columns followed by a "newdoc" marker holding the file id (# newdoc id = ro.legal). Each sentence in a document is labelled by a unique ID (example: "# sent id = ro legal.4"), followed by the text of the respective sentence (# text = ...). Following is a tab separated list of 14 columns, according to the first descriptor line in the file. It contains the word id, word form, lemma, universal part of speech tag, language specific part of speech tag, list of morphological features, head of the current word, universal dependency relation, underscore in columns nine and ten (since we don't use any enhanced dependency graph features or miscellaneous features), named entities in BIO format, NP chunk information, IATE and EUROVOC annotations.

The entire annotated corpus has a size of 29GB and was archived using an archiving task, resulting a zip archive of 4.3GB, downloadable through the platform and was later stored in the MARCELL repository.

5. Conclusion

This paper presented an integrated, high performance platform for Romanian language, called RELATE. It allows researchers to upload a large corpus and perform annotations as well as complex analysis on the data. To achieve parallelization of time-consuming annotation operations, the platform uses a scheduler-runners mechanism. This allows CPU-intensive operations to be distributed across multiple processing nodes across a network or even across the Internet.

By integrating current state of the art modules for processing Romanian language, developed by different research partners, the RELATE platform strives to become a national reference portal.

Multiple input and output file formats are supported, while the internal format used by the platform is the CoNLL-U Plus format. Large archives can be uploaded, processed and finally downloaded in a standard annotated format.

The platform is loosely coupled with the processing pipelines, by means of URLs accessed by the task runner processes, thus complying with a micro-services architecture. Therefore, one of the key future developments for the platform is envisaged to be its containerization in the form of multiple docker containers: one for the interface and one for the processing pipeline. This would allow for quick deployment on new processing nodes as well as increased durability when faced with operating system updates or changes in external libraries.

RELATE will be further enhanced with new Romanian language technologies/computational resources as they become available. While we do not aim at standardizing language technologies interoperation or annotation visualization, thus admitting supplementary programming effort for each new addition, our *focus* is to keep thinking on how to best visualize and link automatically generated annotations with their supporting computational resources in such a way that the widest interested audience is best served doing their work.

In the spirit of European Language Grid, as National Center of Competence for Romania, we will try to persuade all the developers of technologies and resources for Romanian to adhere and contribute to the RELATE portal with new tools and data-sets.

6. Acknowledgements

Part of this work was conducted in the context of the ReTeRom project. Part of this work was conducted in the context of the Marcell project. Speech recognition and synthesis components referenced in this work were developed within the ROBIN project. Machine translation components referenced in this work were developed within the ”CEF Automated Translation toolkit for the EU Council Presidency”, TENtec no. 28144308.

7. Bibliographical References

Aho, A. and Corasick, M. (1975). Efficient string matching: An aid to bibliographic search. *Commun. ACM*. 18:6, 333-340.

Bański, P., Diewald, N., Hanl, M., Kupietz, M. and Witt, A. (2014). Access Control by Query Rewriting. The Case of KorAP. In *Proceedings of the Ninth Conference on International Language Resources and Evaluation (LREC’14).* Reykjavik, European Language Resources Association (ELRA).

Boroș, T., Dumitrescu, D.Ș. and Burtică, R. (2018). NLP-Cube: End-to-End Raw Text Processing With Neural Networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, pages 171-179.

Boroș, T., Dumitrescu, D.Ș. and Păiș, V. (2018b). ”Tools and resources for Romanian text-to-speech and speech-to-text applications”, in *Proceedings of the International Conference on Human-Computer Interaction – RoCHI 2018,* pp 46-53.

Boroș, T. and Dumitrescu, D.Ș. (2015). Robust deep learning models for text-to-speech synthesis support on embedded devices. In *Proceedings of the 7th International Conference on Management of computational and collective IntElligence in Digital EcoSystems (MEDES'15)*, Caraguatatuba, Brasil.

Carp (Mitrofan) M. (2019). Extragere de cunoștințe din texte în limba română și date structurate cu aplicații în domeniu medical. PhD Thesis, Romanian Academy, 144 pages.

Che, W., Li, Z. and Liu, T. (2010). LTP: a Chinese Language Technology Platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations (COLING '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 13-16.

Coman, A., Mitrofan, M. and Tufis, D. (2019). Automatic identification and classification of legal terms in Romanian law texts, In *Proceedings of ConsILR 2019*, Cluj, România, pp 39-49.

Cunningham, H. (2002). GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36(2):223—254.

Diewald, N., Hanl, M., Margaretha, E., Bingel, J., Kupietz, M., Bański, P. and Witt, A. (2016). KorAP Architecture – Diving in the Deep Sea of Corpus Data. In: Calzolari, Nicoletta et al. (eds.): *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16),* Portoroz, European Language Resources Association (ELRA).

Federmann, C., Giannopoulou, I., Girardi, C., Hamon, O., Mavroeidis, D., Minutoli, S. and Schröder, M. (2012). META-SHARE v2: An Open Network of Repositories for Language Resources including Data and Tools. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, Turkey, pages 3300-3303.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). AllenNLP: A Deep Semantic Natural Language Processing Platform. arXiv:1803.07640.

Hinrichs, M., Zastrow, T., and Hinrichs, E. W. (2010). WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*, pp 489-493.

Ion, R. (2007). Word Sense Disambiguation Methods Applied to English and Romanian. PhD Thesis, Romanian Academy, 148 pages (in Romanian).

Ion, R. (2018). TEPROLIN: An Extensible, Online Text Preprocessing Platform for Romanian. In *Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2018),* November 22-23, 2018, Iași, Romania.

Miller, G.A. (1995). WordNet: A Lexical Database for English, *Communications of the ACM*, Vol. 38, No. 11:39-41.

Mititelu, B.V., Tufiș, D. and Irimia, E. (2018). The Reference Corpus of Contemporary Romanian Language (CoRoLa). In *Proceedings of the 11th Language Resources and Evaluation Conference – LREC’18,* Miyazaki, Japan, European Language Resources Association (ELRA).

Morton, T. and LaCivita, J. (2003). WordFreak: an open tool for linguistic annotation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations - Volume 4 (NAACL-Demonstrations '03),* Vol. 4. Association for Computational Linguistics, Stroudsburg, PA, USA, 17-18. DOI: https://doi.org/10.3115/1073427.1073436.

Păiș, V. (2019). Contributions to semantic processing of texts; Identification of entities and relations between textual units; Case study on Romanian language. PhD Thesis, Romanian Academy, 114 pages.

Păiș, V., Tufiș, D. and Ion, R. (2019). Integration of Romanian NLP tools into the RELATE platform. In *Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language – CONSILR 2019*, pages 181-192.

Perovšek, M., Kranjc, J., Erjavec, T., Cestnik, B. and Lavrač, N. (2016). TextFlows: A visual programming platform for text mining and natural language processing, *Science of Computer Programming*, Volume 121, Pages 128-152.

Stan, A., Yamagishi, J., King, S. and Aylett, M. (2011). The Romanian Speech Synthesis (RSS) corpus: building a high quality HMM-based speech synthesis system using a high sampling rate. *Speech Communication* 53(3):442—450.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S. and Tsujii, J. (2012). brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations Session at EACL 2012*

Tufiş, D. and Mititelu, B.V. (2014). The Lexical Ontology for Romanian. In Nuria Gala, Reinhard Rapp, Gemma Bel-Enguix (eds) Recent Advances in Language Production, Cognition and the Lexicon, pages 491–504, Springer**,** 2014