RegMiner - Tutorial

Karolin Winter[1], Manuel Gall[1], Stefanie Rinderle-Ma[1,2]

[1]Research Group Workflow Systems and Technology, Faculty of Computer Science,
[2]Data Science@Uni Vienna, University of Vienna, Vienna, Austria
{firstname.lastname}@univie.ac.at

In order to facilitate (re-)processing, the uploaded documents are retained in a database. Do not upload data that is subject to privacy.

RegMiner is a web service for mining and visualizing constraints from regulatory documents and is accessible at

`http://regminer.wst.univie.ac.at/`

A screencast is available at

`http://gruppe.wst.univie.ac.at/~gallm6/RegMiner/Video/`

It employs NLP and data mining techniques in order to automatically extract compliance constraints group them based on three options and visualize the results in a graph-based representation. RegMiner enables a separation of relevant and non-relevant document parts and provides insights into, e.g., duties of stakeholders. A single page application (cf. Fig. 1) based on HTML markup and JavaScript components serves as user interface. In order to start the process a user needs to provide the following information.

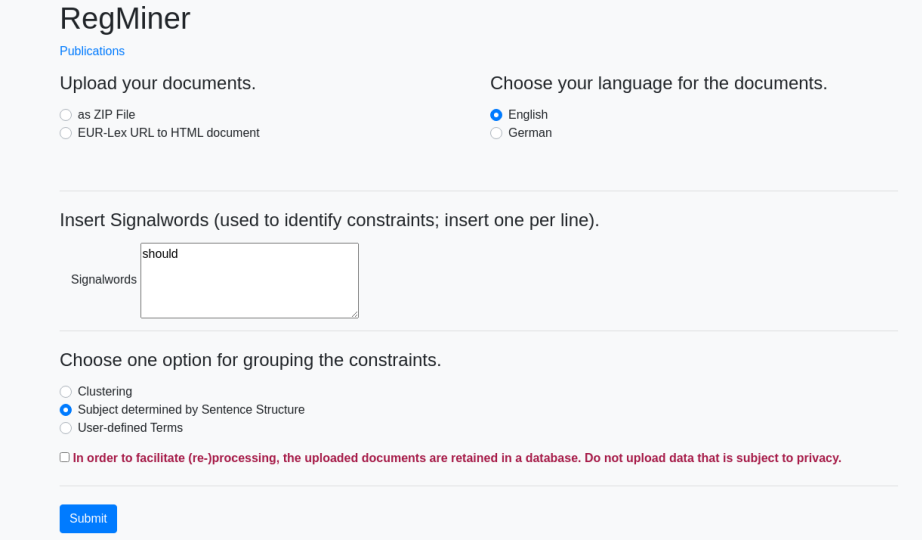

Fig. 1: Screenshot of Initial Configuration

Document. A document can be provided via two options. First, it can be uploaded as ZIP file. In this case, the document should already be partitioned into several .txt files each containing one part of the document, called paragraphs in our context. This must be ensured by the user and can be, e.g., based on the document structure by splitting the document into its sections (cf. [1,2]). Though this is not mandatory, it facilitates the visual inspection of results afterwards. In addition, a restriction to a selection of document parts can be enforced by the user. Further examples can be downloaded via

`http://gruppe.wst.univie.ac.at/projects/RegMiner/index.php?t=prototypes`

As second option, we integrated support for the EUR-Lex platform[1] which contains an extensive collection of legal documents such as the General Data Protection Legislation affecting stakeholder of various domains. The user can provide an URL referencing the HTML markup of the desired EUR-Lex document (cf. Fig. 2). The document is downloaded and automatically split into sections based on HTML tags and attributes. Please bear in mind that such large documents need more time for processing.

In this tutorial we use the *macro-financial assistance* document[2]. Copy the link and enter it into *EUR-Lex URL* field.

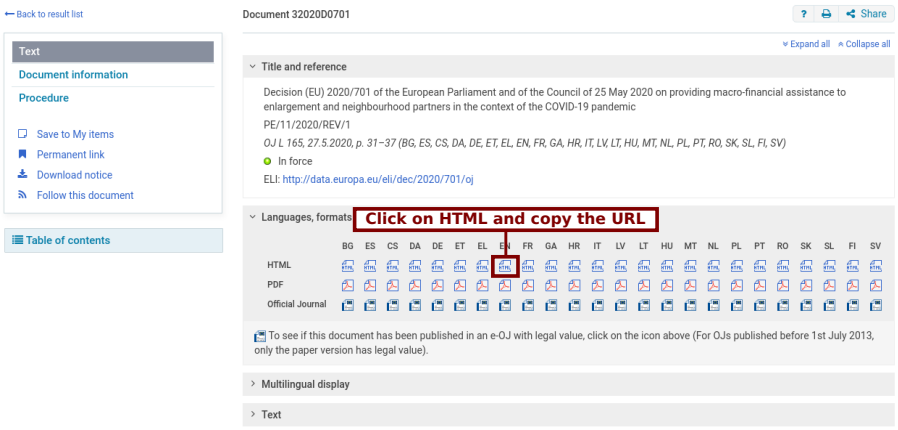

Fig. 2: Retrieving a suitable URL from EUR-Lex

Language. Currently, English and German is supported as document language. For the tutorial an English document is used.

Signal Words. Specify at least one signal word that is used to identify constraints, e.g. “shall”, “should” or “must” (cf. [1]). In our tutorial we use “shall”, “should” and “must” as signal words.

[1] `https://eur-lex.europa.eu/homepage.html?locale=en`

[2] `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32020D0701&qid=1591613404669&from=DE`

Grouping Option. Three options for grouping constraints are available

Clustering Constraints are clustered based on term frequency using k-means++; in this case the user needs to specify the number of clusters.

Subject determined by Sentence Structure The subject of each constraint is identified based on the NLP tags determined by a NLP parser. Per subject one group is created.[3] This option is fully automatic and no further user input is required. For the sake of a simple tutorial this option is chosen.

User-defined Terms Constraints are grouped based on terms defined by the user which need to be uploaded as .txt file. Each term defines one group, e.g., if "authority" and "user" are contained in the .txt file, all constraints containing "authority" are assigned to one group, all constraints containing "user" are assigned to another group. If neither of them is present, the constraint is shifted to group "undefined", if both of them are present, the constraint will be contained in the "authority" as well as "user" group.

Figure 3 depicts the filled form containing the example information.

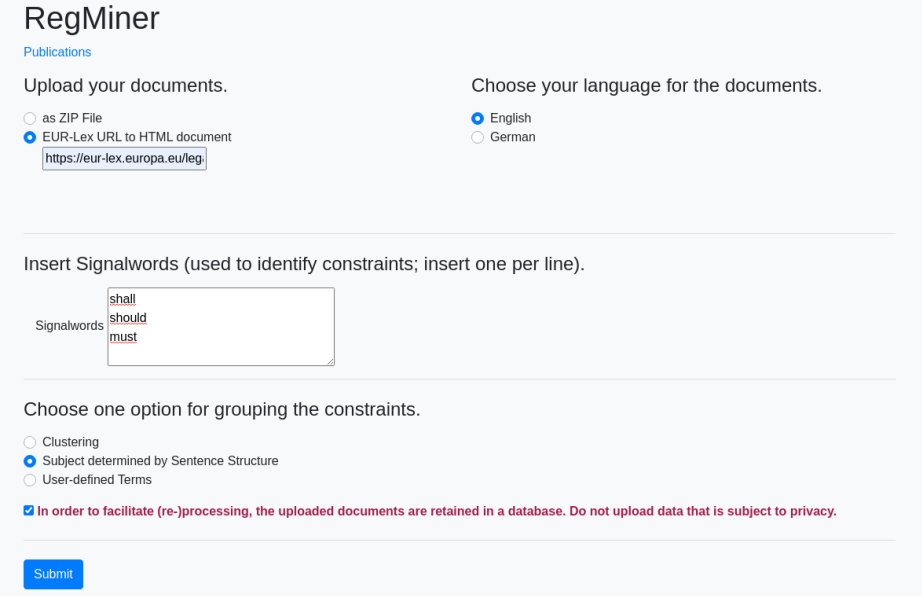

Fig. 3: Screenshot of the form configuration for the example demonstration.

In order to start the process press `Submit`.

As soon as the results are retrieved the graph consisting of several dots accumulated in clusters is displayed. Each dot represents one constraint and the color

[3] For further details see [1].

indicates the corresponding cluster. By hovering over the dots the constraint is shown. By double-clicking onto a dot the paragraph containing the constraint is displayed above the graph. The Resulting graph of the document chosen for this tutorial is depicted in Figure 4. The *Subject determined by Sentence Structure* grouping created multiple groups such as, *report*, *macro-financial assistance* and *Commission*.

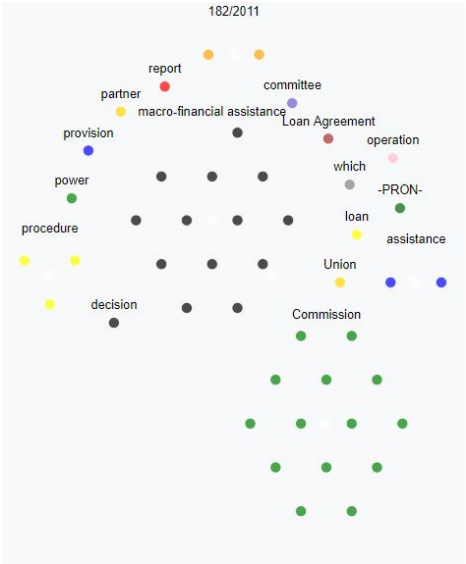

Fig. 4: Resulting graph

Acknowledgment

This work has been funded by the Vienna Science and Technology Fund (WWTF) through project NXT19-003.

References

1. Winter, K., Rinderle-Ma, S.: Detecting constraints and their relations from regulatory documents using NLP techniques. In: On the Move to Meaningful Internet Systems. pp. 261–278 (2018)
2. Winter, K., Rinderle-Ma, S., Grossmann, W., Feinerer, I., Ma, Z.: Characterizing regulatory documents and guidelines based on text mining. In: 25th International Conference on Cooperative Information Systems. pp. 3–20 (October 2017). https://doi.org/10.1007/978-3-319-69462-7_1, `http://eprints.cs.univie.ac.at/5279/`