D2.3 Final Text Mining Solution

Project Acronym:	PoliRural	
Project title:	Future Oriented Collaborative Policy Development for Rural Areas and People	
Grant Agreement No.	818496	
Website:	www.polirural.eu	
Contact:	info@polirural.eu	
Version:	1.4	
Date:	30 May 2021	
Responsible: Partner:	KAJO	
Contributing: Partners:	SUA, Agroinstitut, City Nitra, AREI, LLF, Perifereia, MIGAL, 21C, Innovagritech, AGFutura, S&L, HAMK, ERDN, MurgiaPiu, CONF, GGP, TRAGSA	
Reviewers:	CKA, S&L	
Dissemination Level:	Public	X
	Confidential - only consortium members and European Commission Services	
Keywords:	Text Mining; Semantic Explorer; text analysis; data extraction; Topic identification; tools for policy-making; Artificial Intelligence; Machine Learning; Natural Language Processing (NLP); rural development	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 818496

Revision History

Revision no.	Date	Author
1.1	31.03.2021	Tommaso Sabbatini (KAJO), Denis Kolokol (KAJO)
1.2	06.04.2021	Tuula Löytty (S&L)
1.3	15.04.2021	Patrick Crehan (CKA)
1.4	30.05.2021	Milos Ulman (CULS)

Table of Contents

List of Tables

List of Figures

Executive Summary

This report describes the final text mining TM tool developed within the context of the Polirural project (the Semantic Explorer) and its use by project's partners. It complements the tool that can be accessed at www.semex.io and via open API at this link. The deliverable describes the aim and then, more extensively, the technologies, functionalities and their application in some of the Polirural research tasks.

Semantic Explorer has been developed as a support tool for researchers involved in policy related activities concerning rural areas. During the development it emerged that the Semantic Explorer may produce valuable outputs for researchers involved in Foresight, System Dynamics Modelling (SDM) and policy evaluation activities. Specific functionalities have been developed to assist researchers and facilitators involved in Foresight and policy evaluation but can be pplied also to other research fields. Although it was planned to use TM in needs analysis the time schedule did not allow to have a mature Text Mining tool at the time of the needs analysis (M9). It must be also noted, that WP2 has not received any specific requirements or inputs from the needs analysis team for the development of solutions for the related task.

The organization of working groups and training sessions involving Polirural partners led to the co-development of tailored frontend applications based on the available technologies. The solutions include the possibility of processing vast amounts of text and extracting the most relevant excerpts, also evaluating its general sentiment. Moreover, it allows automatically summarizing of long texts and pulling out various Named Entities indicating to the reader whether the text relates to certain persons, organizations, geographical locations etc… Semex.io is able to analyse text in ten different languages (Flemish, Czech, Finnish, Greek, English, Italian, Latvian, Polish, Slovak, Spanish). It was not possible to develop a working system for Hebrew and Macedonian due to the unavailability of open-source semantic libraries for these languages. Semantic Explorer can be used to understand with a few clicks the general meaning of a long text and thus can help researchers by reducing the cognitive load related to tasks that are essential to policy processes.

The solutions above mentioned have been tested by ten out of the twelve Pilots on various case studies including policy evaluation and Foresight activities. From the policy evaluation case study some of the most recurrent topics related to rural areas are: rural areas, biodiversity, services, development, tourism, agriculture, landscape and others. If some Pilots have found the insights from TM "interesting" and "giving another perspective" others have underlined the potential of text mining but also the inefficiencies of the system developed. Indeed, some shortcomings about text mining applied to policy have emerged. The semantic libraries used in semex.io, those that are available online, present some gaps when analysing policy related texts. The jargon used in both policies and political analysis reports is very specific and TM does not always perceive the language subtle nuances. Moreover, its efficacy is very much linked with the sophistication of available semantic libraries which greatly differ from one language to the other. The results obtained for English, Spanish, Italian and Flemish texts are much more accurate than the ones for other 'less used' (and with a more complex grammar structure) languages such as Latvian, Finnish, Czech and Slovak. Also, these 'less used' languages present a reduced volume of messages in social media, shrinking consistently the potential of text mining. Developing more consistent custom models for policy related text could be an

interesting idea for future developments but limiting this work to more used languages (in particular English) should be considered. In this case however, the involvement of linguists may result necessary.

Introduction

Text Mining (TM) uses complex algorithms and cutting-edge technologies such as Artificial Intelligence and Machine Learning to detect patterns in vast amounts of text. Natural Language Processing (NLP) transforms text into numbers which then can be processed by the computer to gain insights out of the data. Thus, it is possible to analyse the immense amount of information available online, such as scientific papers, large collections of libraries and archives, social media, etc. TM can for instance evaluate the sentiments of discussions in social media regarding a certain policy; summarize long texts; find patterns and dependencies in vast amounts of text; categorize text by locations and/or by topic, etc. The current tool is based on NLP functions including Topic Extraction, Named Entity Recognition and Sentiment Analysis. Through these NLP processes users can access analysed text and visualize the result through effective graphical representations. The technology behind the tool developed is described in Chapters 2 and 3 of this deliverable.

Figure 1 Example of visualizations in Semantic Explorer

One of the objectives of the Polirural project is to bring solutions to policy-makers in order to support European rural areas in responding to contemporary challenges. In particular, this deliverable describes and accompanies Semantic Explorer (semex.io and open API), the text mining tool developed to provide support to researchers and facilitators involved in policy processes in rural areas. Semantic Explorer is being tested in Polirural by researchers preparing inputs for use in policy evaluation, Foresight exercises and other use cases (Chapter 5 describes the use cases and results). Testing the system has revealed to be a good

opportunity for also evaluating the usefulness of text mining in research activities related to policy processes. What emerged is that Semantic Explorer can bring some insights, for instance, by indicating the most recurrent topics related to a certain issue. Semantic Explorer provides the possibility of processing a vast amount of text with a few clicks, extracting the most relevant text excerpts and evaluating its general sentiment. It can also automatically summarize long texts and extract various Named Entities indicating whether the text relates to certain persons, organizations, geographical locations etc… These solutions can help researchers by reducing the cognitive load related to their research tasks that are essential to policy processes.

However, TM in policy related activities entailed various challenges. Firstly, it must be noted that language is a key element of a semantic projects and the type of languages involved in the project may determine the results. In this project we have used the open-source semantic libraries available on the Internet that are usually developed for marketing analysis purposes. The **jargon** used in policies is extremely specific and even the scientific papers, technical reports and social media messages related to policy making tend to have a very specific language where the potentials of text mining cannot be used at their full extent. In our case, Semantic Explorer tends to analyse most of the sources uploaded by Pilots as slightly positive, with very few negative paragraphs.

Another variable linked to language is constituted by the differences in which Semantic Explorer can recognize nuances in **various languages** and by the amount of social media messages per each language. From our experiment it emerged that for larger languages such as English, Spanish and Italian the system is fairly efficient, while for other less used languages such as Czech, Finnish, Latvian, Slovak the results of text mining present many errors. This is probably because for some languages, English and Spanish being the best cases in our experiment, the related semantic models are much more sophisticated than for the other languages. English is the most used language on the Internet, including in social media, and it is in general considered the lingua franca of computing. It is also the language of science making it the perfect language for a text mining project.

Finally, an important characteristic that emerged as determinant for the results of a text mining project is the human factor. Text mining is based on advanced computing and statistical practices and its use requires an aptitude in informatics as well as a basic knowledge of data analysis. Moreover, using text mining for research activities is an innovative methodology and like all the innovations it requires a certain time before it is accepted and included in the list of recognized tools. Stepping out of one's comfort zones can be easier for some but a limit for others. In this project we have tried to surmount this variable by providing extensive training to partners and regional Pilots, giving them practical tasks and creating opportunities for feedback. This has proven to be a fundamental step for the recognition of text mining as a useful technology for desk research activities, with the acknowledgement of its limits. Chapters 4 and in part 5.1 describe the training path including the practical tasks

that were proposed to partners and regional Pilots to become proficient in its use and eventual ambassadors in their regions. The most obvious positive result, in our opinion, is the fact that some of the partners have started to use Semex autonomously for their research activities as explained in Chapter 5. Moreover, in Chapter 6 WP2 researchers have summarized the critical issues that emerged from this experience that could also be considered for future similar activities.

Semex is also the repository for a collection of more than 5000 sources about rural development created by Polirural partners. The sources include scientific papers, technical reports, policies as well as news related to various topics, such as rural areas' needs and policy evaluation. Geographically the sources cover the regions of the 12 Pilots. Moreover, the system automatically streams Twitter for messages related to European rural policy related issues. Semex offers public access to all the sources as well as to the streamed Tweets with various technical options in terms of visualisation and search possibilities. The library is growing, thanks to the continuous contribution of partners, and may become an interesting source for rural policy related information. More information about the Regional Library is provided in Chapter 7 and in deliverables 4.1 and 4.3.

In order to manage such a complex system and to allow for various updates also responding to partners' feedbacks the developers decided to implement a microservice architecture that is described in chapter 8.

Finally, the conclusion attempts in summarising the main achievements and highlight some general conclusions.

2 Semantic Explorer -

Semantic Explorer can be accessed through its frontend application at www.semex.io or via open API at this link. The tool analyses data text creating structured information from unstructured material (Objective 2 of Polirural). The tool's design and specifications are the result of intense communication with Work Packages 1, 4 and 5 and with some of the partners involved in the innovative aspects of the project (CKA, AVINET, 22SISTEMA, CZU, S&L, VITO etc.). Its role in Polirural is embedded in the innovative concept which aims at "creating a reusable framework that draws upon participatory principles, stakeholder knowledge, big data, original research and advanced analytics to deliver more accurate foresight for rural regions, contributing to new and enhanced policy interventions." (Polirural Project Proposal) as illustrated in Figure 2.

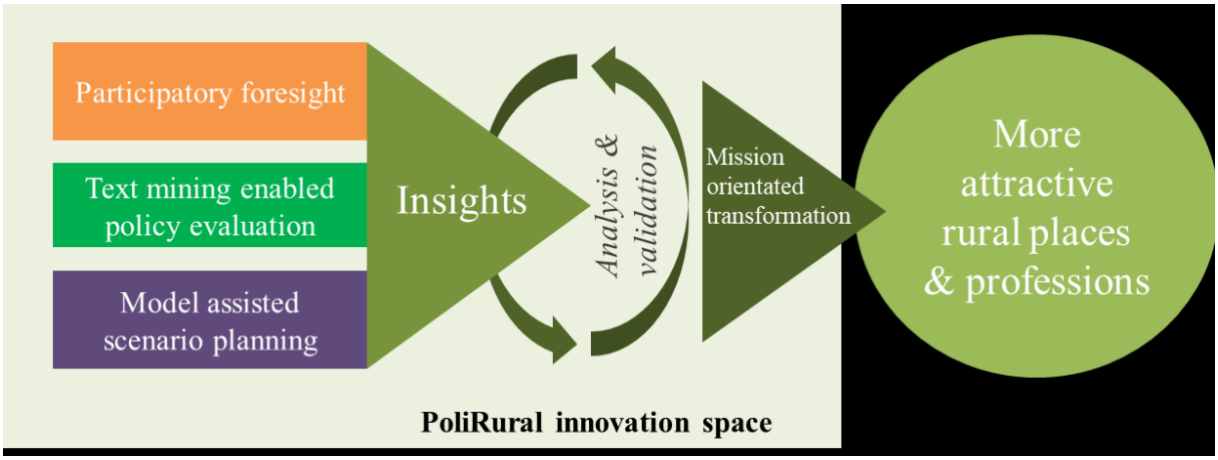

Figure 2 Polirural concept (Extracted from Polirural Project Proposal)

Semex.io reflects this concept and has been designed with the following scopes in mind: To be of support to researchers and facilitators preparing inputs for use in a Foresight exercise; to bring valuable inputs in support to researchers involved in SDM activities and to help researchers involved in policy evaluation. As illustrated in Chapter 5 other use cases have emerged during the Polirural project.

Moreover, Semantic Explorer allows text analysis from scientific literature and reports as well as from Tweets collected by own developed crawlers. The tool highlights the most relevant and emerging topics (Topic Manager); it captures well-known or recognised needs mentioned in policy documents for Needs Analysis; it can localize the impact of relevant trends at regional level for Deep Dives (Profiler); it extracts the location, giving some hints about the issues to be addressed and dynamics in trends and behaviours.

2.1. Scenarios

In its core Semantic Explorer is a set of tools that can be connected together in Pipelines to support processes of solving real-life tasks. Since the tool is primarily aimed at Foresight Exercise, System Dynamics and Policy Evaluation, and the main terms and concepts of those three domains differ substantially, we had to come up with our own set of concepts for a user-system interaction in order to cover the majority of tasks.

2.1.1. User-system interaction

The Interaction between the end-user and the system is based on the following concepts:

- Actions
- Pipeline
- Workspace

By Action we understand a feature of Semantic Explorer, which takes a user input, performs a single action on the data and returns output. Examples of actions:

- a call to a particular method of TextProcessor or TextAnalyzer (e.g. extraction of topics, calculating sentiment, etc.)
- data filtering (for example, filtering posts from Social Media by time frame and topics)
- aggregation of values obtained in the course of a previous action (e.g. averaging sentiments or displaying gathered data on a timeline).

Pipeline is an internal representation of the process. Each pipeline consists of a defined number of actions. A user interface for each action is a workspace.

Workspace is a particular arrangement of visual elements on a web-page, that are organized in a sensible way and together provide a way to define the whole Pipeline in a single view.

The relations between those three concepts can be represented by the following diagram.

Figure 3 Relations between Actions, Pipelines and Workspaces

One can think of Semantic Explorer as a toolbox - a set of features with each feature corresponding to a certain action. If those actions are enchained in a particular order, together they define a certain scenario of problem solving.

The examples of scenarios that follow are not hard-coded in https://semex.io/ but all of them can easily be configured upon request. In order to formulate such a request the end-user should learn what features it offers and how they can be combined. Below we give examples of the real-life scenarios that are supported by the system. In the following sections we describe in a greater detail basic data structures and relationships used in those scenarios.

2.1.2. Issues analysis

In terms of Semantic Explorer the Issues Analysis is the process of extracting topics from text with a negative sentiment. If not filtered, the top 40 topics in this list represent global trends. If filtered by geo-locations, they become local trends. They can also be filtered by semantic similarity to other topics, etc.

The example of the process of Issues Analysis:

Step 1. Extract data from Social Media
Filters:

- pick up main Topic (can be selected from the Similarity Cluster or via search by keywords)
- define geographical area - select the name of the region / place or use interactive map to define [lat, lon] and radius
- define period of time (optional)

Output:

- collection of texts extracted from Social Media.

Step 2. Aggregation by topics extracted from texts
Inputs:

- dataset obtained at Step 1.

Output:

- Social Media data grouped by topics:

```
{
topic_1: [
        tweet1, tweet2…
        ],
topic_2: [
        tweet5, tweetN…
        ]...
}
```

Step 3. Sentiment analysis of posts by topics:
Input:

- dataset obtained at Step 2.

Output:

- average values of sentiment calculated from all the posts for each topic.

Step 4. Topics filtering:
Input:

- values sentiments vs. topics obtained at Step 3.

Output:

- Top 10 topics that got the most negative sentiment values.

The output of this process can be interpreted as "the issues connected to a certain topic" (for example, a name of Policy or a Trend). From this result we can stretch further, looking for the issues of the subtopics of the current topic and using them in the next round of Issue Analysis.

For example, the Issues Analysis by the topic "Disadvantaged areas" uncovers the most negative sentiment connected to its subtopic "Underdeveloped areas". Then the end-user can:

- use "Underdeveloped areas" as a main topic for the next round of Issues Analysis (thus performing even deeper analysis of problems in a certain region)
- filter out posts from Social Media by this topic and negative sentiment, thus accessing the actual opinion of stakeholders on the various aspects of this topic.
- use it as a filter to extract documents from Regional Library that is relevant to this topic, thus creating a Curated Reading List (see the next section).

2.1.3. Curated Reading List

The following is the citation of the note from Patrick Crehan (partner CKA).

> One can imagine an interface whereby 100 articles or other sources are mined and presented to the reader, who quickly scans these, and based on their own insights, will swipe left or swipe right in order to select a sub-list containing those it deems most appropriate for the task at hand.
>
> This sub list could then be further processed, into a CRL or curated reading list, suitable for distribution without the need for further editing work by the reader. For example, a final step might result in a final summary, a summary of the summaries if you will, with a list of references, and a summary of the effort required to create the CRL, any annotations that the “reader” may have added and if needed the original abstracts for each of the retained references.
>
> This final object is the “curated reading list”...

Step 1. Selection of sources by topics
Input:

- list of topics - can either be list Topics manually selected from the database or the list of topics obtained as a result of the Issues Analysis (see 2.1.2 "Issues analysis").

Outputs:

- dataset of sources from Regional Library on the selected Topics.

Step 2. Text summarization:
Input:

- dataset obtained at Step 1.

Output:

- A collection of the short texts (summary) and links to the original sources.

Step 3. Profiler:
Input:

- collection obtained at Step 2.

Output:

- Named Entities extracted from each summary
- List of external links to the documents in the internet semantically close to the Topics (issues) defined at Step 2 (i.e. found in the provided sources).

A Reading List consists of both internal documents from the Regional Library and external resources, links to which can be found in the Regional Library. Such a list of publications can be proposed to the Partners and Stakeholders working in defined Pilot Areas.

This use case can also be useful to create periodic summaries in the form "resources vs. topics" as the Regional Library is being continuously updated.

2.1.4. Sentiment Analysis

Sentiment Analysis is one of the main processes for uncovering issues and their potential solutions in the pilot areas. Given that all texts in the Library are automatically labelled with topics and sentiment, it is also possible to find sources from other pilot areas on the same topics (even if in other languages) with positive sentiment, thus trying to find information about possible solutions of the problem. The topic in this case can be either the name of the policy (like "CAP") or a real-world problem (like "air pollution").

Step 1. Aggregation of posts from Social Media by time
Input:

- filter by time (for example, "last year" or "last month")
- aggregate by time-chunks for grouping of SM posts (for example, "each quarter" or "each month").

Output:

- posts of Social Media gathered in each chunk.

Step 2. Topic extraction
Input:

- Collection obtained at Step 1.

Output:

- the most important topics for each chunk of time, extracted from the posts.

Step 3. Aggregation by topics
Inputs:

- Collection of data from Social Media obtained at Step 1.
- List of Topics obtained in Step 2.

Output:

- The collection of data in the form of "topics by time":

{
2 months ago:

```
        "social security": [posts...],
        "access to funds": [posts...],
    last month:
            "biofuel": ...
            "air pollution": ...
        ...
}
```

Step 4. Sentiment analysis / aggregation for each particular chunk of time:
Input:

- collection obtained at Step 3

Output:

- dataset in the same form with "sentiment" averaged for each time chunk

Step 5. Put it on the map:
Input:

- data gathered at previous step

Output:

- Data aggregated by:
  (a) chunks of time at the whole time frame[1]
  (b) region[2] (for example, GAUL[3])
  (c) average value of sentiment

This result is statistical in its nature: it gives the opportunity to see how sentiment changed in time in a particular place on the map regarding selected topics (for example, "how the perception of CAP in Flanders has been changing every month from 2012 to 2020"). It can also be visualized on the map.

[1] In terms of the Semantic Explorer system this is called "time histogram".

[2] All data from Social Media are geo-tagged, otherwise they are not allowed to be stored in the database .

[3] http://www.fao.org/geonetwork/srv/en/metadata.show%3Fid%3D12691

2.2. Basic data structures and relationships

The basic data structures and the relationships between them are represented in Figure 4

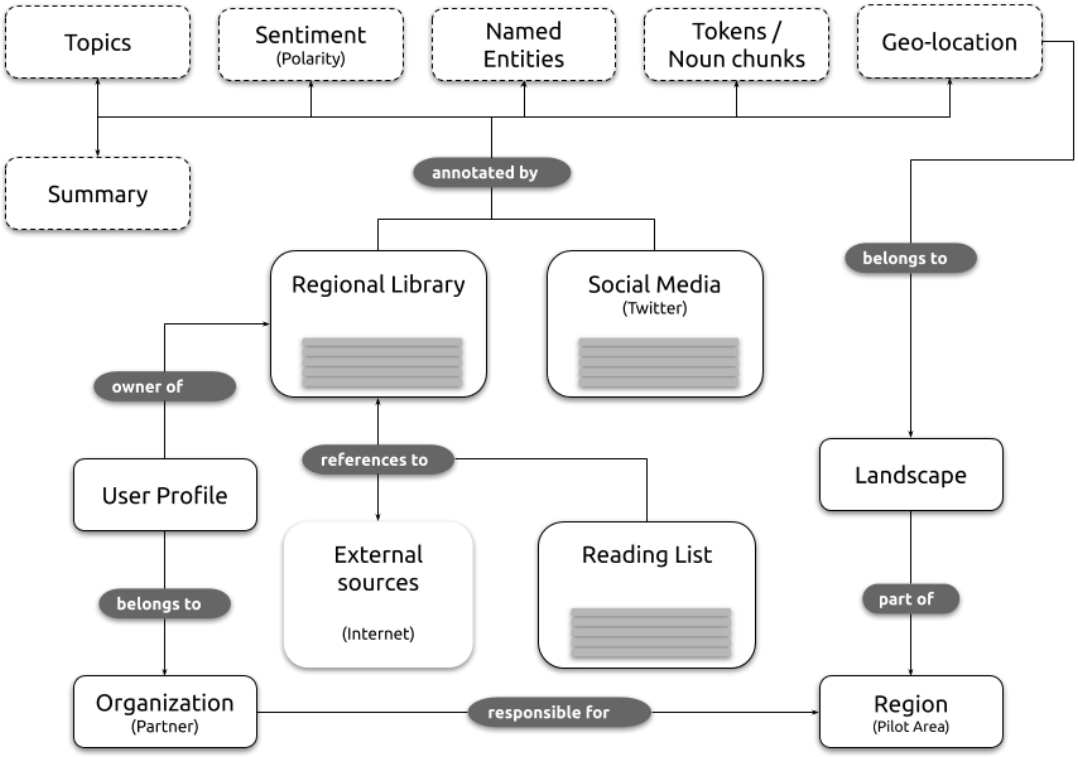

Figure 4 Basic data structures and relationships

A continuous border line represents objects that are stored in the database as separate entities. Those of them labelled with a grey list on the bottom are "transactional", i.e. they can change often (be inserted, updated or deleted), while the others are relatively static (so called "reference models").

Dashed line expresses the "embedded" character of data. For example, geo-location (which consists of place name, country, geo-coordinates, etc.) does not exist in the database as a separate object, but attached to records of Regional Library or posts from Social Media.

Both embedded and static objects are the objects that can be used for filtering and search in transactional data (for example, sources from Regional Library filtered by Topics, search by Named Entity in Library, filter Social Media by belonging to a certain Landscape through its gro-location).

The absence of line indicates that the objects are not stored in the database at all, however can be referenced (links in the internet can be a part of a Reading List).

In addition, every static and transactional resource is labelled with a date of creation and last update, which can also be used for filtering.

2.2.1. Reference models

By "model" we understand a data structure with defined fields and their types. Reference models (as opposed to transactional models), are defined structures for storing relatively static and descriptive data. In Semantic Explorer those are User Profiles, Organizations, Regional Library and Geo-Locations.

User profiles and Organizations

User profiles serve the purpose of tracing user's actions upon other reference models (such as Regional Library or Keywords). For instance, the system traces the activities from each user and keeps track of when each new source has been entered, when the existing record of Landscape has been amended and when a list of Keywords for a particular language has been changed.

Users are grouped around Organizations. The primary purpose of Organization reference is to store in the database information about the PoliRural Project's partners. In the future this purpose can be extended.

Warning: a user cannot use the system if he or she isn't a part of some organization.

Regional Library

Regional Library is the core reference in the system. Essentially this is just a set of links to the resources in the internet, that contain textual[4] information on the topics, defined by the goal of the project. Each link is accompanied by language, owner (a user who entered added the link to the library) and dates of creation and last update.

Some use cases generate a subset of links from Regional Library and other links available on the Internet. Those links can be combined in a Reading List (see detailed description of the Reading List in the Deliverable 2.1 "Technical Specification of the Text Mining Tool").

Keywords is a reference of all possible keywords that serve the purpose of tracing Social Media.

More details about the Regional Library collection is included in Chapter 6 of this deliverable.

Geo-locations

There are two main interdependent structures in the system that define geo-locations: Regions and Landscapes. Their relation to each other is "1 to (0..N)" (in one Region there can be zero, or one, or more than one Landscapes).

Landscape is described by multipolygon on the map, and can be extracted from the system in GeoJSON format.

[4] The crucial word here is "textual" - the very aim of the Text Mining is to extract meaningful information from text. Thus the effectiveness of the analysis depends on the amount of text in the resource: documents with a lot of pictures and tables produce poorer analysis and slow down the system.

The Organization profiles (i.e. partners of the Project) can be annotated with the Landscape, while each source can be associated with a certain Region.

2.2.2. Applied Semantic Analysis

There are a number of features that represent Semantic Analysis in terms of annotating free-form texts. Some of them are used for analysis in Use Cases, while others are considered as auxiliary.

Main features:

- Topics - short phrases from known thesauruses (e.g. GEMET) to categorize a chunk of text.
- Named Entities - persons, places, phone numbers, percentages, even names, etc.
- Geo-location - a place which can be geo-tagged (a second stage result from obtained after Named Entities Recognition [NER])
- Polarity - a value of Sentiment (-1..0..1)

Auxiliary:

- Text Summary - 5-6 sentences that describe the essence of the whole text
- Tokens / noun chunks - separate words and short phrases that define objects and subjects of sentences
- Word frequency - a list of most common words and phrases accompanied by the frequency of appearance in texts.

Definition: Applied[5] Semantic Analysis is the process of annotating chunks of text with the main features.

The process of Semantic Analysis is different for the two main resource types in the system.

2.2.2.1 Applied Semantic Analysis in Regional Library

Regional Library consists primarily of the big texts obtained by crawling the URLs entered in the system by the end-users.

In order to perform a detailed analysis of each source, the process of annotation consists of the following steps:

- The text is split by paragraphs
- For each paragraph lists of Topics and Named Entities are obtained
- From the list of Named Entities those are separated that can be geo-parsed
- Each paragraph is then split by chunks with exactly one found geo-location in it (chunks of text with no geo-locations are possible, too)

[5] We add the word "applied" to denote the application of SA in the current system - https://semex.io/

- Each chunk is annotated with:
  - full list of Topics of the paragraph
  - value of Polarity of the paragraph
  - individual list of Named Entities of the chunk

The described scheme is necessary for performing the following procedures:

- search by keywords
- filter by Topics and Named Entities
- aggregate Polarity
- place documents on the map

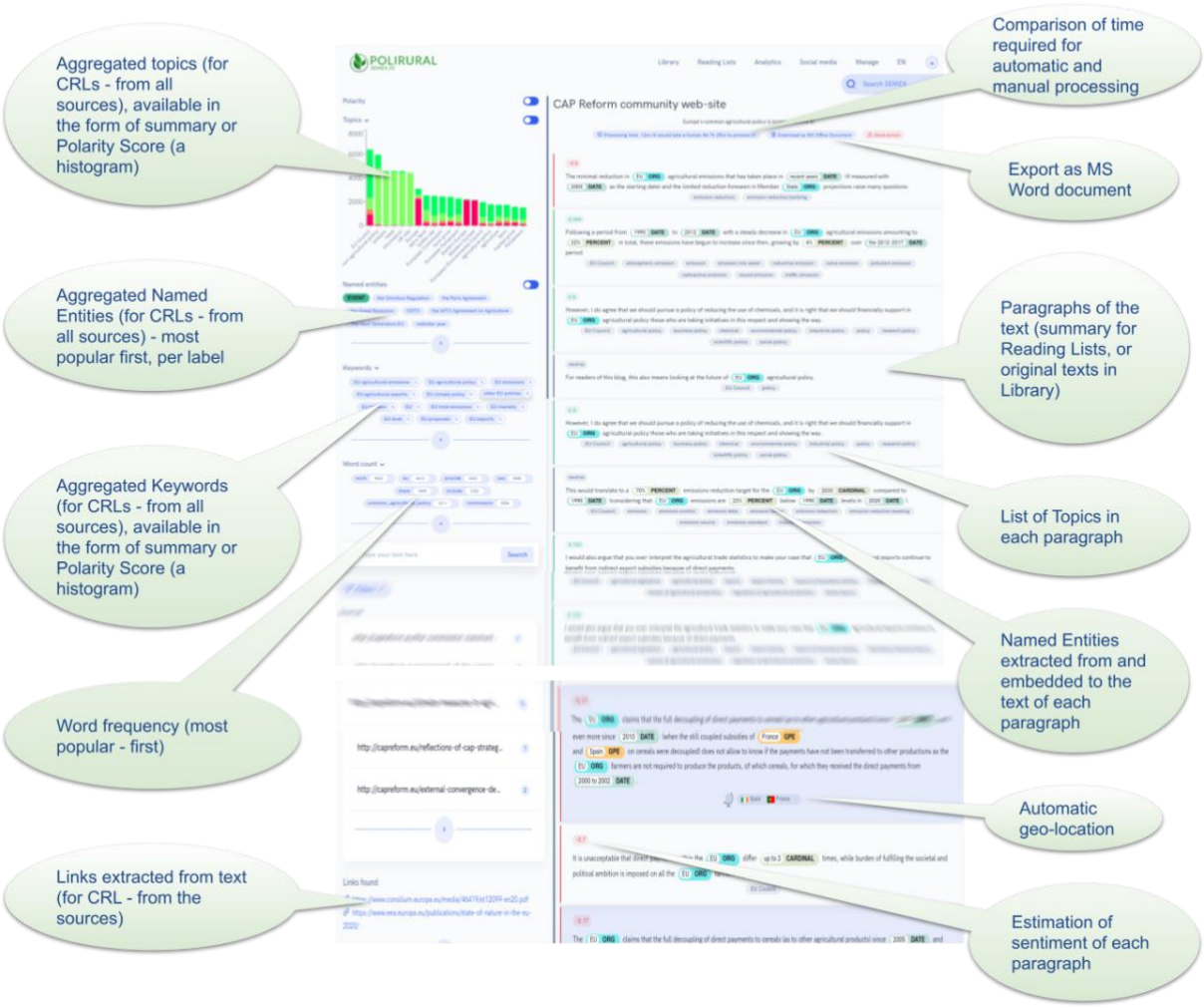

Figure 5 The example of the result of Applied Semantic Analysis on a document from the Library

2.2.2.2 Applied Semantic Analysis in Social Media

Currently only Twitter streaming is supported as a source of data from Social Media.

Each tweet is being already annotated by geo-location (currently on the tweets from the chosen list of countries are being accepted), therefore the structure of the data is the same. A tweet always has one paragraph (#0, chunk #0).

There are also two conditions for a tweet to be registered in the database of Demantic Explorer:

- a tweet must be annotated with at least one topic
- a tweet's polarity cannot be "neutral", because in this case a tweet doesn't bring any information on the sentiment of the obtained topic(s)

Figure 6 Example of positive Tweet extraction related to the European Agricultural Fund for Rural Development (EAFRD) and to the future of CAP

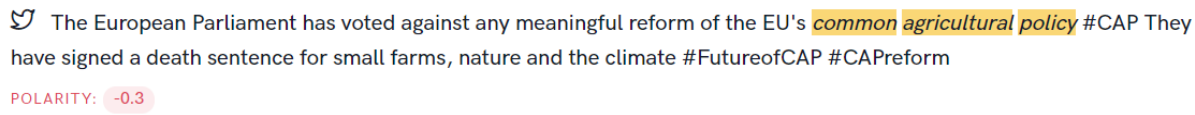

Figure 7 Example of negative Tweet extracted related to the Common Agricultural Policy

2.2.3. Semantic structures

Semantic structures are used for navigating documents in the Library that passed through the stage of Applied Semantic Analysis. There are two main Semantic structures that are used in https://semex.io/.

2.2.3.1 Similarity Cluster

Existing thesaurus explaining semantic relationships between topics in a chosen semantic space (for example, "rural attractiveness") and represented (visualized) as undirected graphs. Similarity Cluster is language dependent.

a)

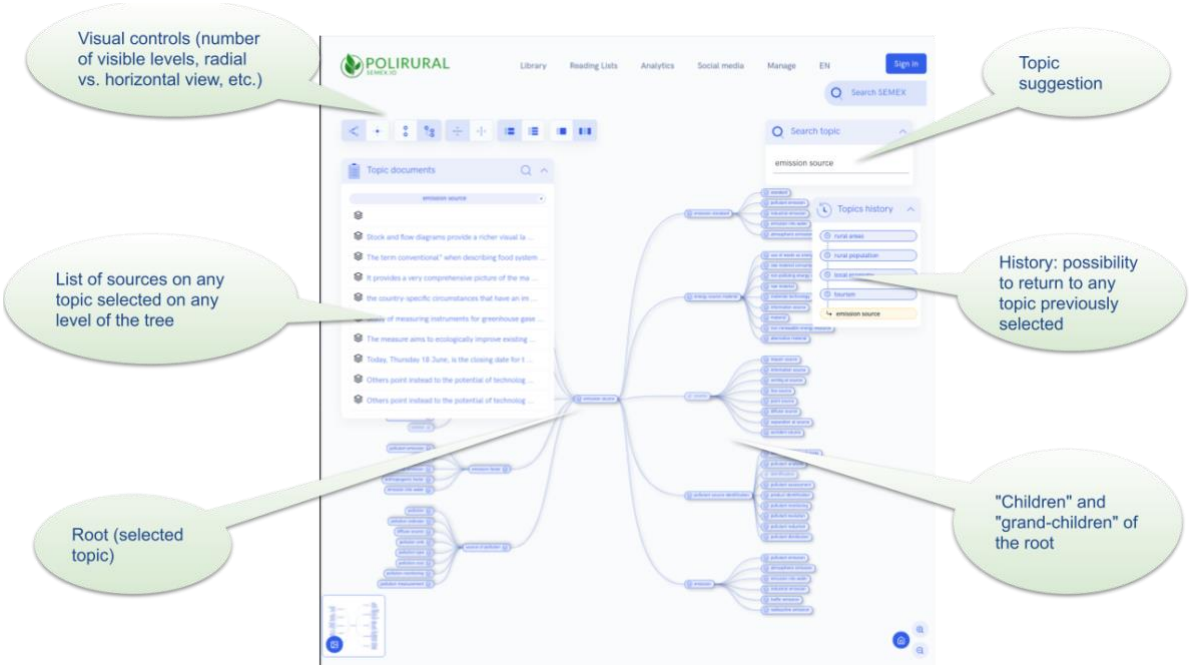

Figure 8 Visualization of Similarity Cluster in a) Radial Tree (conceptual view), b) Horizontal Tree (actual view)

3 Text Mining basic concepts

This chapter introduces the basic concepts related to text mining and how these are applied in Semantic Explorer. It also attempts at analysing the limitations and the improvements of the NLP system applied in www.semex.io.

3.1. NLP, Machine learning and AI

Research in Natural Language Processing (NLP) started in the 2nd half of the 20th century together with research on Artificial intelligence (AI). Processing of natural language was one of the goals of AI and also the well-known Turing test[6] formulated in 1950 by Alan Turing is about understanding of text and communicating in natural language[7]. Initially, the systems were based on complex sets of hand-written rules, later various machine learning algorithms and statistical methods started to be used. In the early 2000s the neural networks started to be adopted. This came together with the rise of available computing power and training data accessible from internet content. Currently all state-of-the-art systems in NLP are based on Neural Networks (deep learning). The research in this field grows rapidly and there are still open problems[8].

Generally speaking, NLP breaks down language into shorter, more basic pieces, called tokens (words, periods, etc.), and attempts to understand the relationships between tokens. The higher NLP tasks include Content Categorization, Topic Discovery and Modelling, Contextual Extraction, Sentiment Analysis, Text-to-Speech and Speech-to-Text Conversion, Document Summarization and Machine Translation[9]. The importance of text mining and NLP rises because most of the information available is not structured and in "free" natural language form, so there is a need for automated processing of this information.

In our system NLP is used for Keyword and Topic extraction and modelling, Sentiment analysis and Named Entity Recognition (NER). All these tasks require more basic NLP tasks such as Word and Sentence tokenization, Dependency parsing, Part of Speech tagging, lemmatization, semantic comparisons of words. We use state-of-the art libraries such as spaCy, udpipe, gensim or polyglot and models trained on large text corpora.

[6] "The Turing test is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human." https://en.wikipedia.org/wiki/Turing_test. Accessed 13 May. 2020.

[7] "natural language" is "A language that has developed naturally in use (as contrasted with an artificial language or computer code)" https://www.lexico.com/definition/natural_language

[8] "The 4 Biggest Open Problems in NLP - Sebastian Ruder." 15 Jan. 2019, https://ruder.io/4-biggest-open-problems-in-nlp/. Accessed 13 May. 2020.

[9] "A Brief History of Natural Language Processing" 22 May. 2019, https://www.dataversity.net/a-brief-history-of-natural-language-processing-nlp/. Accessed 13 May. 2020.

3.2. Brief description of the system workflow and core functionalities

The core of the NLP system consists of 3 main processes - topic extraction, named entity recognition and sentiment analysis as shown in Figure 9.

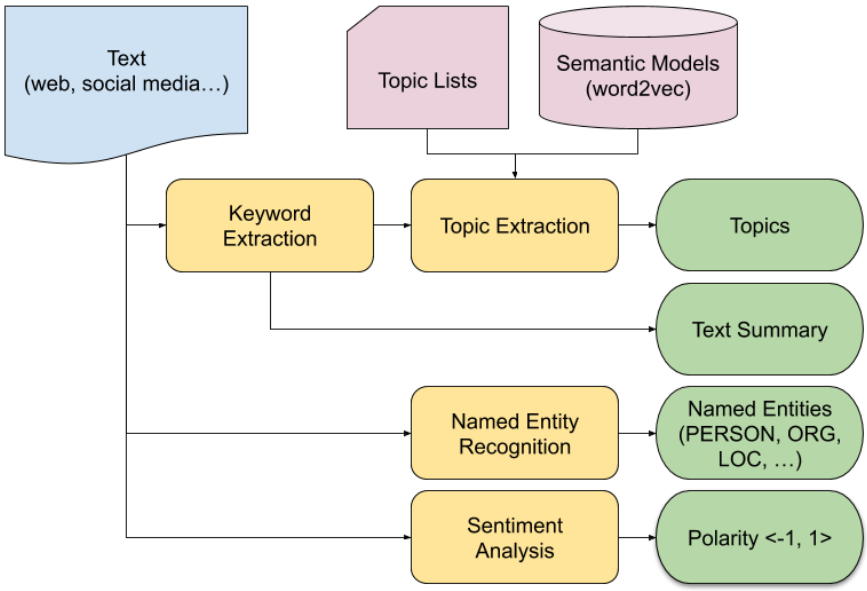

Figure 9 Main processes of the NLP system

3.2.1. Text extraction

Web page links are entered by partners. The system crawls these pages, extracts text and stores them in Elasticsearch database. We can process regular web pages, various types of text documents (pdf, .doc, …) and also process the streams from social media (e.g. Twitter). Further details about crawlers are described in the chapter dedicated to crawlers and in Polirural's Deliverables 2.1 and 2.2.

3.2.2. Topic extraction and modelling

It is the most complex NLP task in the system. It is a multi-label classification problem. We are using a restricted list of topics which can be assigned to the text. It can be broken down into smaller tasks here below described.

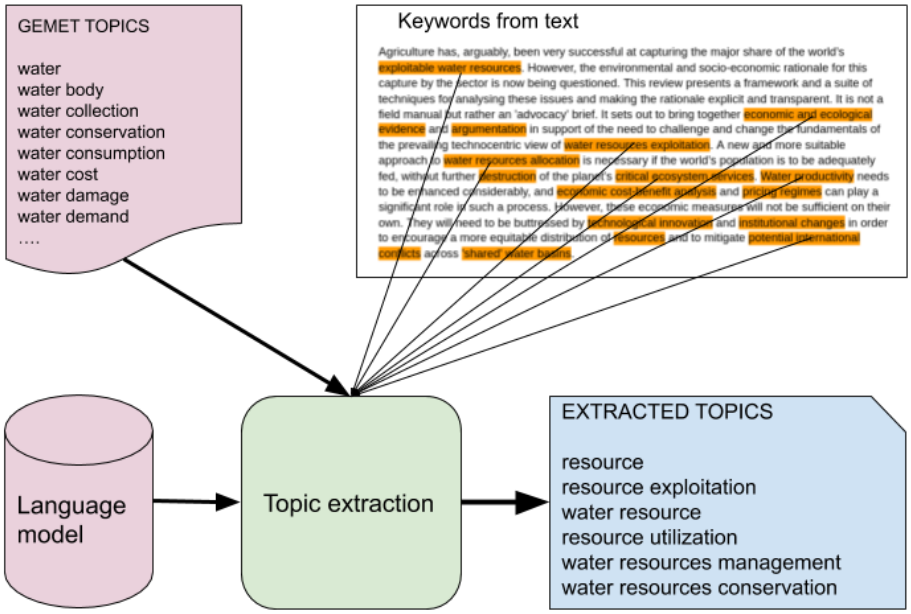

Figure 10 Topic extraction

3.2.2.1. Keyword extraction

The most important words or noun phrases are identified in the text. Graph-based TextRank algorithm[10] (based on Google's PageRank algorithm which was proposed to analyse the link structure of web pages) was selected as the best method to extract a list of keywords from the text. These keywords can also be later used for summarizing the text.

3.2.2.2. Topic similarity comparison

Extracted keywords are compared to a list of topics (see description of GEMET below) and most similar topics are selected. The comparison is done by Word Mover's Distance[11] (WMD) algorithm which is one of the most accurate algorithms available to semantically compare documents. We optimized the performance WMD algorithm itself and took advantage of multiprocessing to gain the best speed performance. The topic similarities are computed in word embedding space. We adapted available pre-trained fastText models which were trained large with corpora from Common Crawl and Wikipedia data[12]. Because of performance these models were reduced down to improve the performance and RAM usage - the models contain only 100 000 - 200 000 most used words for every language and these words are in lemmatized form to be able to deal with high inflected languages (e.g. Slovak, Czech, Polish).

[10] "TextRank: Bringing Order into Texts." https://web.eecs.umich.edu/~mihalcea/papers/mihalcea.emnlp04.pdf. Accessed 13 May. 2020.

[11] "From Word Embeddings To Document Distances." http://proceedings.mlr.press/v37/kusnerb15.pdf. Accessed 13 May. 2020.

[12] "Word vectors for 157 languages · fastText." https://fasttext.cc/docs/en/crawl-vectors.html. Accessed 13 May. 2020.

3.2.2.3. Subtopics exploration

The system allows users to explore topics which were extracted directly from the text and in addition it also gives the possibility to explore further topics which are semantically similar to those extracted. This is done using domain specific word embeddings model (trained language model where the words having the same meaning have a similar representation) which links semantically similar words and topics. Also, relations from a multilingual thesauri containing a general terminology for the environment (see section 3.3 GEMET) can be used to find related topics.

3.2.3. Named Entity Recognition (NER)

It is the identification of real world entities such as persons, organisations, locations and others (we can identify 19 different categories of entities). Accuracy of NER is highly dependent on the annotated datasets. Big human annotated datasets (e.g. OntoNotes with almost 1.5M English words[13]) exist only for larger languages (such as English, Dutch, Greek) but entities can be trained also on datasets generated from Wikipedia data[14] - in this case only 3-4 entity types can be identified (Person, Location, Organisation and Miscellaneous). Extracted entities can link the text to specific geographic location, to an organisation or concrete persons.

3.2.4. Sentiment analysis (opinion, emotion)

It is a technique used to identify or classify the polarity of text. The opinion polarity can range from negative (-1), through neutral (0) to positive (1). Correctly classifying sentiment is a challenging task if compared with human judgement. Human raters typically only agree about 80% of the time, so even when the raw accuracy of automated sentiment analysis is below perfect, statistically, it can be almost as good as human analysis[15,16]. The methods used for sentiment analysis can be based on a dictionary of polarity words or training a machine learning or deep learning model. Dictionaries can be manually collected by humans or automatically generated. The accuracy is dependent on the data used. Not much training data sets is available to train the models (mostly various reviews are used) and even the data for testing the models is scarce (this is a problem for almost all non-English languages). The best

[13] "OntoNotes Release 5.0" https://catalog.ldc.upenn.edu/LDC2013T19. Accessed 27 May. 2020.

[14] "POLYGLOT-NER: Massive Multilingual Named Entity ...." 14 Oct. 2014, https://arxiv.org/abs/1410.3791. Accessed 13 May. 2020.

[15] "How Companies Can Use Sentiment Analysis to Improve Their Business - Mashable." 19 Apr. 2010, https://mashable.com/2010/04/19/sentiment-analysis/. Accessed 20 May. 2020.

[16] "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis" http://people.cs.pitt.edu/~wiebe/pubs/papers/emnlp05polarity.pdf. Accessed 20 May. 2020.

available methods for sentiment analysis were implemented (polyglot[17], VaderSentiment[18] and deep learning with LSTM). To be noted that these libraries are not policy specifically.

3.3. GEMET[19] (GEneral Multilingual Environmental Thesaurus)

GEMET is the thesaurus used in Semantic Explorer. It has been developed as an indexing, retrieval and control tool for the European Topic Centre on Catalogue of Data Sources (ETC/CDS) and the European Environment Agency (EEA), Copenhagen. The basic idea for the development of GEMET was to use the best of the presently available excellent multilingual thesauri. It contains more than 5000 topics in 37 languages. Topics have hierarchical relationships (broader and narrower terms) and also "related terms

") relationships. Topics from GEMET were chosen because it was conceived as a "general" thesaurus, aimed to define a common general language, a core of general terminology for the **environment**, which is closely related to our main topic of interest - rural development and related policies.

3.3.1. Languages covered

The Semantic Explorer can process texts in the 10 languages listed in Table 1. It was not possible to cover Hebrew and Macedonian because of no availability of trained models and lack of linguists in this project.

Languages	Unique topics	Words in the semantic model	NER classes
Czech - cs	4961	148193	3
Greek - el	4831	188506	6
English - en	5167	200221	18
Spanish - es	4933	169586	4
Finnish - fi	4094	157796	3

[17] "Building Sentiment Lexicons for All Major Languages" 23 Jun. 2014, https://www.aclweb.org/anthology/P14-2063.pdf. Accessed 13 May. 2020.

[18] "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text" https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/download/8109/8122. Accessed 13 May. 2020.

[19] "About GEMET - Eionet - europa.eu." https://www.eionet.europa.eu/gemet/en/about/. Accessed 13 May. 2020.

Italian - it	5064	164954	4
Latvian - lv	4896	126658	3
Dutch - nl	4308	168494	18
Polish - pl	4792	122589	6
Slovak - sk	4943	148816	3

Table 1 List of languages covered by Semantic Explorer with their characteristics: number of unique topics, number of words in the semantic word embeddings model, number of NER classes in the NER model

3.4. Limitations and possible improvements of the NLP system

Semantic Explorer uses state-of-the-art NLP methods, that consists of 3 main processes - topic extraction, NER and sentiment analysis as shown in Figure 9.

However, since these methods are very innovative new techniques are emerging and space for improvement is plausible. In particular, the following points could be analysed and eventually improved.

- Short texts - Understanding of short text is still a challenge in NLP[20]. Our system deals well with longer texts (e.g. whole paragraphs in documents, web pages), but the keyword extraction (and subsequent topic extraction) can struggle with short texts (e.g. Tweets). One possibility for improvement is to add an additional classifier which could be regularly trained on already collected data in the library and even improve in time. Another option is to tune the noun_chunk extraction so that a higher number of keywords can be extracted also from short texts.
- Noun chunks (noun phrases which are used as keyword proposals) are extracted according to defined rules. For some languages (en, el, es) the rules were part of the available spaCy NLP model, but for the rest KAJO had to implement them. For languages such as Latvian and Finnish, fine tuning will come with time based on the native speakers' user feedback. For this task it will be essential that Pilots test extensively the tool and report feedback to KAJO´s developers. For this reason, a specific feedback system has been created so that users become part of the machine training process.
- Language-specific tuning of parameters - currently most of the parameters within NLP are set globally (e.g. threshold for accepting a topic), but it's planned to fine-tune them

[20] "Understanding Short Texts - ACL 2016 Tutorial." http://www.wangzhongyuan.com/tutorial/ACL2016/Understanding-Short-Texts/. Accessed 13 May. 2020.

to be language-specific. Larger dataset of texts collected from library sources and social media together with the user feedback on the topic extraction accuracy will be used for this task.

- Lemmatization[21] is crucial for languages which inflect words (Slovak, Czech, Polish, ...) to reduce the size and improve the performance of models and accuracy of semantic comparisons. Lemmatizers from spaCy and udpipe models are used, but the lemmatization is sometimes incorrect (e.g. for languages which connect words together - like Finnish). In such cases the words are not found in the model and the semantic comparison can struggle. Specialized language-specific or even dictionary based lemmatizers can be used to improve the accuracy of lemmatization.
- Sentiment analysis - the accuracy of sentiment analysis depends on the training data. Using general dictionary-based polarity classification can lead to lower accuracy when analyzing specialized text. We can improve the sentiment analysis accuracy by training neural network models on manually annotated domain-specific texts. This will require cooperation with experts in the field and native speakers.
- Named Entity Recognition was trained on general text and for some languages there are only 3 classes of entities available - more detailed classification and classification of specialized terms would be beneficial. Creating an annotated corpus specific for our field of interest with close cooperation with native speakers and experts would be needed.
- Macedonian and Hebrew: during the project development it emerged that in order to cover these two languages linguists should be involved in the project. The thesaurus used in Polirural does not provide a version for these two languages and in order to create NLP models from scratch an advanced language-specific knowledge is required. The following steps need to be considered in order to create a valuable text mining system also for these two languages:
    - translate GEMET topics to Hebrew and Macedonian - this must be done with the support of field experts, **available in Polirural.**
    - NLP model (DEP, POS) lemmatization - for Hebrew, an UDPIPE model is available while for Macedonian it must be created from scratch. Further research must be carried in order to understand what tasks will be entailed. Most probably a manual tagging of some text with universal dependencies will be needed at first - expert/native speakers with advanced linguistic knowledge should do this, **not available in Polirural**.
    - word2vec models (for semantic comparisons of topics) - **there are models for both languages**.

[21] "the process of reducing the different forms of a word to one single form" https://dictionary.cambridge.org/dictionary/english/lemmatization

- In conclusion, in order to expand Semantic Explorer to Macedonian and Hebrew additional tasks would need to be carried in cooperation with native speakers experts in the rural domain and with an advanced knowledge of their native languages (linguists).

4 Training and evaluation

The training and evaluation are fundamental parts in the development of a software. In the case of Semantic Explorer it had to take in consideration various stakeholders and several aspects. Training sessions have been done through several meetings with various groups of partners based on their needs, requirements and level of experience with the tool. Each meeting provided the possibility for partners to provide feedback and input for the system. The following sections provide more detailed information about how project's partners have been trained and how the tool has been evaluated.

4.1. Training

The researchers involved in the development of semex.io realized that text-mining and the functions involved are particularly complex. To exploit the potential of the system some basic knowledge of data analysis and advanced computing is valuable. In this perspective, a fundamental step, to make text mining more accessible to a wider range of stakeholders, is represented by training. Specific concepts linked to text mining have been introduced through the webinar on Text Mining performed by Denis Kolokol (KAJO) in December 2019. During the second Project's meeting further onsite training has been delivered to present the prototype and to instruct partners on how to feed the Regional Library autonomously. Finally, the period June - December 2020 has been dedicated to more advanced and practical trainings to coach partners in the use of the tool and to review it as much as possible.

Training has been an iterative practice based on several meetings between the working groups. The starting point has been the establishment of various working groups based on specific expertise and research fields. Foresight, policy evaluation experts and WP1 coordinator have been consulted and involved in the training phase to better define the research needs of the Pilots. It was natural for each Pilot to individuate one or more text mining referees that could participate in the working groups and try to communicate at best the research needs of its organization/Pilot. Each working group included an expert, a representative from the Pilots and a TM developer. The aim of the working groups has been to test and to provide feedback for further tweaking the Semantic Explorer toolbox and meet users' needs. Trained referees have been given enough tools so that they can provide coaching on text mining within their organizations and to their regional stakeholders, thus multiplying the training results.

As in everything, not all researchers will be good trainers, some might be good enough, and a few will be excellent. Some partners have provided very detailed feedback, uploaded a great number of texts, far in excess of what any expert or groups of experts is capable of doing and in a very short time frame. It will be important for the project to provide opportunities to these new text mining experts to share their expertise with their stakeholders and in particular with policy makers.

The results of the training were quite positive. 92% of respondents expressed a positive change in their opinions regarding text mining following the various steps of training. Moreover, a good majority of respondents were satisfied with semex.io for the activities that they carried out for training purposes.

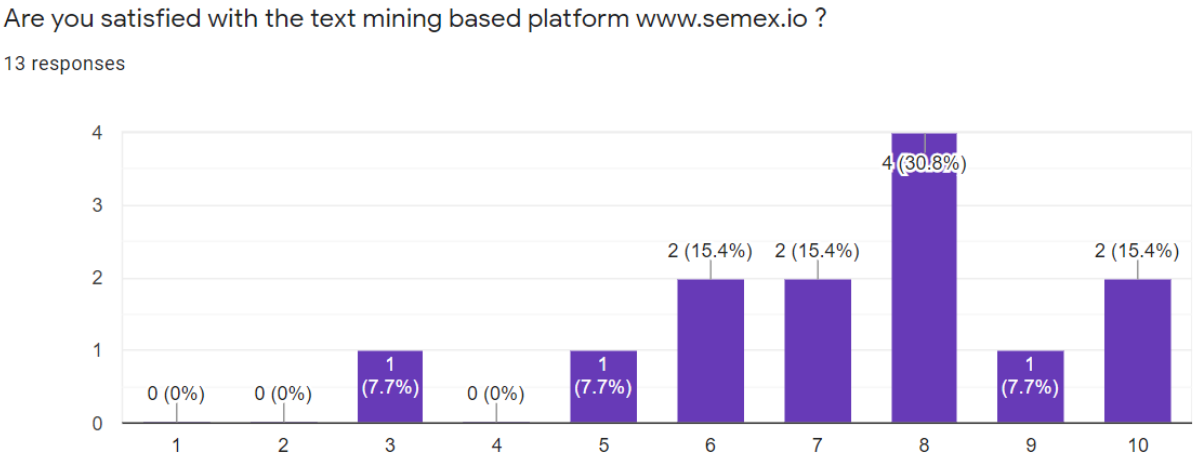

Figure 11 Pilots satisfaction with Semex.io

If some Pilots replied that it is easy to use the tool others have argued that it requires some basic data analysis knowledge and a detailed hands-on manual. Following this input, we have created a user manual accessible in Annex II.

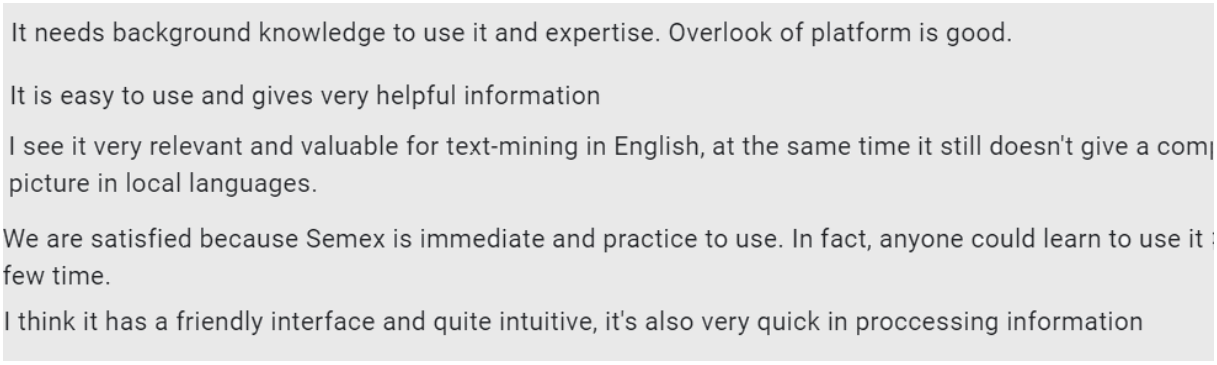

Figure 12 Pilots' review comments on semex.io

4.2. Evaluation

Training and evaluation have therefore been part of the same process. During the training, facilitators ensured the creation of a safe environment where participants could express their opinions and doubts, with space for discussions that were interactive and focused on the objectives. Brainstorming and other types of facilitation techniques have been used to ensure active participation from all the meeting participants. This led to a creative and productive environment and a good occasion for open communication between TM developers and Pilots.

Evaluation surveys have been collected at various stages of the training. There have been informal feedback sessions within the meetings, formal evaluation form filled at the initial phase and final evaluation form filled after the policy evaluation use case.

A positive insight coming from the final evaluation is that the majority of respondents (92%) commented that they would use text mining for their research but in conjunction with traditional research methods such as surveys.

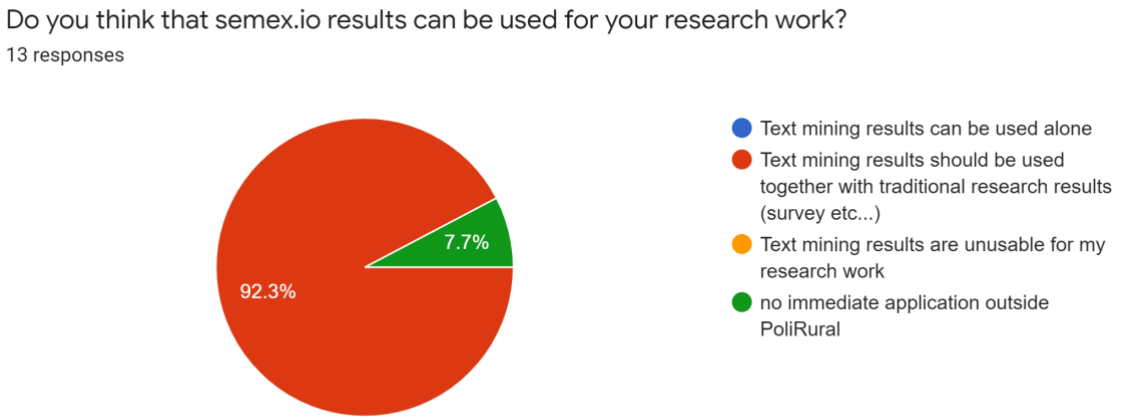

Figure 13 Is semex.io usable for research activities in Pilots?

This confirms the important point, stressed already several times is this paper, that TM should be used as a support tool to research, for example to speed up some desk activities, but cannot replace traditional research. The following 3 quotes from the evaluation questionnaire well capture this concept.

text mining gives a good overlook of the content and polarity, gives an opportunity to identify tipping points, key persons, trends etc., but as this is a system, not always all emotions can be captured completely

As we have done in WP4.5, text mining could be a part of the research. It means that we could use it after having submitted questionnaires (for example), to confirm or not the results obtained.

TM it's specially a tool to get information from desk research, but as in our case there are no many Social Media talking about our region we need surveys, interviews to get stakeholders opinion. The opinions in Social Media are not so technical and thoughtful as needed for the analysis

Figure 14 Pilots comments about the use of Semantic Explorer

5 Use Cases

Semantic Explorer has been tested in three main occasions within Polirural. First of all, the training & feedbacks sessions have been a good occasion to learn and experiment the tool, as described in the previous chapter. Following the feedback received and the fixes developed, semex.io has been used for two main activities related to the policy evaluation task in WP4 and to Foresight, supporting pilots in the Deep Dive exercise.

Moreover, the use of text mining has been extended to other ongoing activities. A research activity using Semex is being carried out by CZU to explore the possibility of adding a temporal field and attempt extracting trends from the text analysed. Finally, Polirural partner S&L is exploiting Semex in the context of another H2020 project (http://cities2030.eu/) attempting at using text mining to understand system thinking, system modelling and City/Region Food System.
This chapter illustrates these use cases and tries to draw some useful insights from them.

5.1. Policy evaluation

One of the innovative aims of Polirural is to experiment the use of text mining for policy evaluation. Together with WP1 leader we developed specific features in Semex.io and then organized a more advanced and focused training (available here) with the objective of testing the system; enhancing Pilots skills in using the tool; and extracting some useful data to be compared with results from survey. The aim was to verify the assumption behind the use of text mining in policy evaluation that it may identify additional issues linked to specific policies not picked up by the survey, and thus paint a more complete picture of a policy under investigation. The aim of the exercise was also to verify the following three points:

- The identification of additional issues/benefits linked to a specific policy (i.e. things that people talk about on the internet) that were not picked up by the survey;
- Confirm/validate survey findings by revealing broadly positive or negative sentiment toward a policy;
- Cast the same policy in a different light compared to survey, and therefore reach a more balanced conclusion about its performance.

For the exercise, Pilots have been requested to make the following actions in Semex:

- Add sources related to a specific policy(s)
- Create a Curated Reading List including all the sources related to the policy
- Analyse the results of the CRL with the following solutions provided in semex:
    - Polarity Scores (sentiment analysis)
    - Topics explorer: understand the semantic relationships and explore the related sources
    - Named Entity Recognition: study opinions of particular persons using the Named Entity PERSON

Finally, each Pilot has been requested to draft a report summarizing the main research outputs and including a discussion of whether text mining confirmed or added any new perspective on survey findings. The reports have been integrated into Deliverable D4.5 - Perceived Effectiveness of Rural Interventions summarizing the main conclusions.

5.1.1. Policy evaluation results

From this case study some of the most recurrent and interesting topics that emerged are: **rural areas, biodiversity, services, development, tourism, agriculture, landscape etc.**. More examples are included in the table here below.

Pilots country	Topic 1	Topic 2	Topic 3	Topic 4
Spain	Rural development	Health	Rural environment	Tourism
Ireland	Development	Services	New community	Tourism
Belgium	Biodiversity	Environment	Agriculture	Landscape
Italy	Agricultural company	Council of the EU	Agricultural product	Agricultural production
Finland	City center	Promotion of trade and industry	Rural habitat	Association
Greece	Rural development	Rural area	Development	LEADER
Latvia	Local development	Local municipality	Technological development	Public transport
Slovakia	Development of transport			

Table 2 some of the most recurrent topics extracted from the policy evaluation experiment

These topics are definitely very relevant to the research area and seem to picture quite correctly the current needs of rural areas and at the same time the directions for current and future development. Obviously four topics cannot precisely describe complex issues. This table, however, summarizes the main topics extracted and a more extensive list could be used to give researchers some hints for further research.

Moreover, the Named Entity Recognition feature allowed the extraction of some worth noting quotes. The Irish Pilot for example, underlined the importance of the following quotes from Dr. Patrick Forrestal (Dr, Senior Research Scientist at Teagasc, Soils, Environment and Land Use Department) extracted from the system:

0.2732 The increased employment in rural areas was mainly due to unprecedented building and construction, commercial and retail services and, to a lesser degree, to manufacturing.

-0.2944 In addition to these vulnerabilities, Irish rural areas are particularly exposed to the price of fossil fuels and constraints on gaseous emissions.

0.9169 Encourage and support migrants with the abilities and skills to create employment to Increased numbers of migrants engaged in the services of LEO.

0.8268 It is important to note that Ireland has a long history of engagement with rural development support frameworks *and* In this context the idea of a more flexible approach to administrative boundaries for the purpose of rural economic planning emerged as a theme from the consultation process.

0.7506 Higher Education Institutes, the region will be able to generate greater levels of human capital, *and* increase regional economic growth.

Figure 15 Named Entities Extraction

A more detailed evaluation of the results extracted by Semantic Explorer for the policy evaluation exercise is available in D4.5, but here we would like to concentrate on the specific application of text mining to research activities related to policy processes.

What emerged is that for some Pilots (Ireland, Spain, Greece, Belgium, Italy) the topics, keywords and Named Entities extracted confirmed the survey findings but also brought some different perspectives. The Greek Pilot, underlined that in order to have better results it is preferable to have a big data set in English. The Italian Pilot confirmed the interesting results that were, however, not in line with their survey outputs. These experiences confirm valuable insights regarding text mining. First, that for some languages, such as English, text mining is working better. Secondly, as concluded by the Greek Pilot, that in order to have good results, there is a need to have a big amount of text. For instance, WP2 NLP experts have calculated that on average, in order to train a sensible language model, it is necessary to feed the system with ca. 800 relevant sources. Although we live in an era of data abundance, smaller research activities with smaller storage facilities require users to manually select the most relevant sources. In our experience, automatic crawlers created too much data and noise and therefore could not replace the work of researchers that remains key in creating a quality data set with

selected sources. As the Massachusetts Institute of Technology puts it this is what makes text mining computationally very costly and 'very different from human learning'[22]. The most recent technological advances in text mining are exploring the possibility of learning from smaller data sets, the so called 'Less-than one shot' learning or LO-shot learning (Sucholutsky, Schonlau, 2020)[23]. Although it is still early to judge the results the direction seems to be very promising.

On the more negative side of the feedback received, other Pilots (Finland, Latvia, Slovakia, Czech Republic) underlined the fuzziness of some of the topics and keywords extracted for their languages and, in general, of the system developed. Indeed, some shortcomings about text mining applied to policy have emerged. The semantic models used in semex.io, those that are available on an open-source base, present some gaps when analysing policy related texts. The jargon used in both policies and political analysis reports is very specific and TM does not always perceive the language subtle nuances. Moreover, its efficacy is very much linked to the sophistication of available semantic libraries which greatly differ from one language to the other. The results obtained for English, Dutch, Spanish, Greek and Italian texts are much better than the ones for other 'less used' (and with a more complex grammar structure) languages such as Latvian, Finnish, Czech and Slovak. In reality, the critical issue here is not if the language is more or less used but whether it has human annotated datasets or not. Also, these 'less used' languages present a reduced volume of messages in social media, shrinking consistently the potential of text mining. In order to have more efficient results it would be necessary to create specific semantic custom models with annotated data, but this would require a big amount of text and of human work as well as the support of linguists. Developing more consistent custom models for policy related text could be an interesting idea for future developments, but limiting this work to fewer languages should be considered. The more the model available is sophisticated the less work will be necessary to tweak it for policy related topics. In this case involving more than one language (i.e. English, Spanish, Dutch, Italian) is plausible. On the contrary, if the available language model is not very accurate a lot of resources will be needed to retrain it and focusing on one language only could result as the best way to go.

As illustrated in the below pie chart it can be concluded that text mining can provide useful insights for policy evaluation, but with some limitations as discussed above. Definitely, the system will require more testing to reach a more precise conclusion even though it looks very realistic to say that text mining is becoming an important tool for research, even in the policy field.

[22] https://www.technologyreview.com/2020/10/16/1010566/ai-machine-learning-with-tiny-data/

[23] https://www.technologyreview.com/2020/10/16/1010566/ai-machine-learning-with-tiny-data/

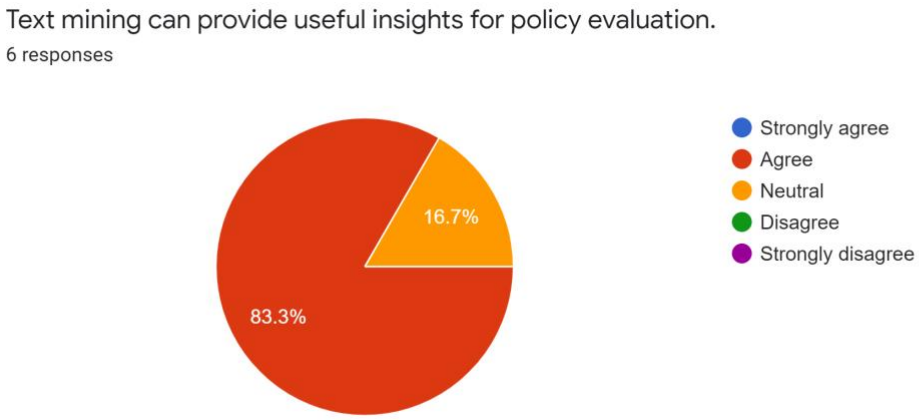

Figure 16 Did text mining provide useful hints to policy evaluation?

5.2. Foresight

Foresight's team leader has expressed a great interest in text mining from the very beginning of the Polirural project. A simple but innovative idea emerged from the many meetings and discussions that the Foresight and Text Mining teams had. The idea is to create a space in semex.io where researchers could store interesting articles about a certain topic and the NLP would extract a series of aggregated results such as **summary**, Topics, Named Entities and Keywords. The technical details of the solution are explained in section 2.1.3. After its development and discussion also with Polirural partners it became obvious that Curated Reading List could be used for many different research activities, including the policy evaluation task.

In Foresight, the Curated Reading List solution has been tested as a support tool for the creation of the document "A STEEPV - Inventory of Drivers of Change". Semantic Explorer should help researchers in Foresight by reducing "the effort required to scan and process a large number of documents on a large number of issues" and by "shortening the work of regional Foresight teams, in the creation of documents such as these"[24]. The solution has been first experimented directly by Patrick CREHAN with his collection of articles related to COVID. The system created a summary of all the sources, a summary for each source and additionally extracted for each source Topics, Keywords and most recurrent words (Wordcount). The results are available here. The Foresight leader evaluated the results of the text mining "very promising" and encouraged its use for developing the preparatory documents and briefing memos for the Pilots' own deep dive exercise.

For the Foresight team a further solution not fully developed is the use of text mining to extrapolate trends in time. The solution is not hard coded in www.semex.io but interested

[24]https://polirural.eu/wp-content/uploads/2021/03/A-STEEPV-Inventory-of-Drivers-of-Change-v4_RP29012021.pdf

developers could eventually use the open API to extract data. Particular interest for such a solution has been expressed by Polirural's coordinating partner CZU who is examining the possibility of a research paper on this topic. The description of the idea is included in section 5.4. Following the reallocation of 3 PMs from WP4 to WP2, KAJO is planning to develop a hard coded solution giving users the possibility of visualizing trends as explained in section 2.1.4.

5.3. TRAGSA use of semex.io

The Spanish partner TRAGSA has been particularly active in the use of Semex.io for their research activities. The description of their use of text mining can well summarize the use of text mining by one of the most active Pilot in this field. This example can also be of inspiration for other organizations involved in policy related activities and may represent an embryonic methodology for the use of text mining in policy processes.

" The use of the Semantic Explorer tool, by the TRAGSA Team, is divided in different phases.

Firstly, we have proceeded to test the tool, adding sources to the library and creating our own CRLs, to see the possibilities offered by the semantic analysis in terms of document summary and polarity analysis, in relation to the opinion expressed in those documents.

Next, the tool has been used to complement the evaluation assessment on the LEADER policy implemented in the territory of Segóbriga (Cuenca, Spain). In this sense, it has been possible to compare and contrast the results of the Semantic Explorer with those obtained using other evaluation techniques. An analysis of the main topics and keywords has been extracted, seeing in each case the differences between the changes in the polarity intervals.

The tool is currently being used to analyze interesting documents for Foresight activities, especially with regard to Deep Dive analysis, to explore the impacts that various issues may have on the future evolution of the territory. Our aim is to extract in a fast and simple way, a synthesis of the information contained in the documents, as well as an assessment of the polarity in the opinions reflected in said documents.

Taking into account that scientific documents are usually written in a generally positive language, we are trying to explore other types of sources such as press articles or social network publications, where opinions are more diverse and can help us to have a more plural vision of the drivers of change of our territory.

5.4. Application of semex.io in system modelling and exploitation of PoliRural results in CITIES2030 project

Polirural partner S&L has been from the beginning of this project very interested in the text mining technology and very active in the TM activities giving many useful inputs to the developers and feeding the Regional Library with hundreds of relevant sources. Recently, S&L

has started to use semex.io attempting to understand system thinking, system modelling and City/Region Food System in the context of another H2020 project, CITIES2030 project **(**http://cities2030.eu/, https://cordis.europa.eu/project/id/101000640**)**. In this subchapter S&L describes the planned activities.

CITIES2030 implements system thinking and system modelling on City/Region Food System transformation. The aim of this activity is "to discuss a methodological framework for CRFS system thinking and scenarios interventions".

The first step in the task is to carry out and document semi-structured discussions on system thinking including ca. 50 people divided into 6 teams. **The second step** is to conduct a literature review, but not in a traditional manner, but using a text mining application. The results of the text mining complement the results of the discussions. It's clear that no text mining application can substitute a human researcher's work. Because CITIES2030 is an innovation action, it is pertinent to seek and pilot new models for the research work.

Text mining purpose is to mine and extract system thinking, system modelling and CRFS knowledge from selected resources i.e., Internet sources and social media. The specific goal of text mining is to identify city/region food system elements and their interconnections. The results help to develop the CRFS Logic Framework Approach (CLFA).

Semex.io, the text mining tool developed in PoliRural, is the perfect candidate for this task. However, some tweaks must be made to the model which is currently focused on rural development. The first thing is indeed to teach Semex that besides rural development it ought to also understand the food system and system thinking issues. The training of Semex model and algorithm requires that at least 800 new resources are added to Semex source library. Adding the sources is done manually. The new sources must encompass quality and relevant texts focusing on CITIES2030 topics e.g., system thinking, food system, urban food system, food value chain, system modelling, problem structuring, causal loop modelling, dynamic modelling, and scenario planning. The 800 resources are added by S&L. Next, the model needs to be trained and fine-tuned, requiring PoliRural's IT expert's contribution.

The results of this research task will be included in CITIES2030 deliverables as well as in Polirural's exploitation deliverable in WP7.

5.5. CZU research proposal - Temporal Adaptability of Text Mining System using semex.io

CZU has recently proposed to look more into details into semex.io and in particular into the temporal variable and exploring the possibility of extracting trends. Such a solution could be also applied to Foresight, as also recently recommended by DG JRC Foresight experts during the recent Rural Vision Week. What follows is a brief description of the activities foreseen.

"CZU proposal is to focus on the temporal nature of text mining. Many existing papers that deal with the topic of text mining are singular endeavours - gather data, prepare solution, analyze text, get results/conclusions and that's it. The advantage of Semex is that it is a system that is being maintained over time. This means that there is an option to capture evolution of change within the input data over time.

One of the first steps of text mining is to build a database of keywords / topics / subtopics and the relationships between them (shared term spaces etc.). However, for the Semex solution there is evolution in time, some terms will become obsolete and replaced by new, more "trendy" keywords. The nature of discussion around rural areas / agriculture might shift into unknown new directions. The research can then focus on how the Semex system is prepared for such changes in time. Describe the updating procedures that help maintain the keyword/topic database up to date with current trends. Evaluate the adaptability of the system for such temporal changes.

One of the key features that might provide an interesting point of view is the incorporation of social media feed (the Twitter module). As far as we understand the part of the Semex system that analyzes data from social media is within the overall text mining general loop (it is not a completely separately build solution) – therefore the paper can also describe the nature of this incorporation and how it can automate/speed up the adaptability of the system to new trends. Meaning that instead of having to wait for manual updates where new source texts are fed to the system and the base model is recalculated, we can describe how any new trends/keywords/topics get automatically incorporated into the database by first being observed during the social media feed analysis.

One thing that may be of immense help would be the ability to support the discussion about these Semex features by actual data. So, we were wondering if it would be possible to capture the changes in time in some way – perhaps by calculating certain metrics over time, for instance:

- Having a list of topics/keywords and calculating their "popularity" (number of tweets? Or perhaps it is possible to calculate some "importance" metrics within the database) and do this over time.
- Having actual numbers that show the evolution over time – some topics becoming less important, some becoming more important, certain new keywords emerging – will help to demonstrate why is it necessary for the system to be adaptable in time, as well as to justify the focus we put on this topic within the proposed paper and it will overall highlight the uniqueness of the Semex solution and distinguish it from other text mining projects."

This research activity has very recently started, and the outcomes shall be available in a few months. The possibility of publishing a scientific paper will be a great help for disseminating the results of this research in particular and of Semex.io in general.

6 General methodology for text mining projects related to policy processes

The previous chapter described various practical use cases for Semantic Explorer. Some of the most active Pilots have started to use the tool for their research activities and WP2 is periodically receiving new proposals for its use. In this chapter we would like to point out the most critical issues that we have encountered and propose some recommendation for future text mining projects related to policy.

Following our experience, we believe that for such a project it would be beneficial to consider carefully the following points:

6.1 Define tool's objectives and use cases with experts

Before developers start to code it is very important to discuss objectives and critical issues with field experts. In our case it was quite difficult to develop solutions for policy evaluation since we did not receive precise requirements from task leaders at this initial stage. We applied **Topics extraction**, **Named Entity Recognition** and **sentiment analysis** to highlight important parts of text but other more specific solutions could emerge from policy evaluation experts. On the other hand, the inputs received from Foresight leader were extremely useful and determinant in the development of the tool. CKA clearly defined what was needed from text mining to support Foresight activities:

a. **Text summaries** to help researchers with literature review
b. The availability of a source repository where researchers can collect interesting sources in a **Curated Reading List**
c. The possibility of extracting **trends**. In Polirural unfortunately, for this option more research is required.

6.2 Curated dataset

Ask field experts and regional stakeholders to **collect interesting sources in a Regional Library** to create a **good dataset**: At this stage it is important that experts curate the contents of the library. In the case where there are various experts from different regions it would be advantageous to ensure a harmonised approach to content inputs from all the partner involved.

6.3 Language selection

From our experience it emerged that it is easier to work on text mining in some languages. Open-source human annotated datasets are available only for certain languages and they seem to be more sophisticated for **languages that are more used or** for those **that already possess a human annotated language model**. In our case English and Spanish are the

languages that worked best and it might not be a coincidence that these two languages are within the most used languages in the world with English being the most used for informatics and research. Moreover, messages in social media are more numerous for some languages than for others. However, if the research must include '**less used languages'** then it is necessary to create a semantic model for the specific language which requires the contribution of specialised linguists or the inputs from thousands of selected volunteers. This can obviously entail a much more important stress on resources. Language selection should be thought very carefully and mixing languages with different levels of sophistication in their models can be risky. Each language needs a different approach entailing distinctive types of expertise and consequently varied types of resources. In case of not very accurate language models it might be necessary to focus the project on one language.

6.4 Technological selection

For this project we selected **Topic extraction**, **Named Entities recognition** and **sentiment analysis**. We realized however that sentiment analysis does not work very well with scientific and technical reports that are written with a neutral tone. A custom semantic model could be created to better understand policy jargon but as in point 3 this would require more resources. Regarding technology a text mining project that aims at a wide public, such as Pilots and eventually policy-makers, should consider the necessity of developing a **frontend application** in order to make the system and its results more accessible.

6.5 Involve final users in a feedback loop from the beginning of the development stage

During the development it is very important to get continuous users' feedback to ensure that developers solutions meets users' needs. Methods like Agile can be very useful as much as identifying the right users with good computing aptitude, basic data analysis knowledge and ability to give constructive feedback.

6.6 Train, test and fix

Learning is a constant for every new informatic tool created. Training is essential to make sure that users understand how the tool works and makes the best use out of it. In our case training has been an iterative practice based on several meetings with various smaller working groups. A reduced number of participants per session created a safer environment where everyone could express its doubts. Thus, training became a good opportunity for feedback collection which was then used to better tailor the system.

6.7 Work on use cases

Having well defined use cases is very important for users. It also ensures that that the information extracted can be useful for research outputs.

- In Policy evaluation Pilots compared results from surveys with results from text mining to find similarities and differences.
- In Foresight Pilots used automatically generated summaries as a support to literature review. Curated Reading List added the possibility of gathering interesting sources and extracting aggregated results. The option of exploring trends extracted through text mining is also envisaged but will need more research.
- Using text mining as an input for system dynamics has been explored, but there was not enough time to create a specific use case in Polirural. S&L is however using semex.io for **literature review** in a task related to **system modelling** in CITIES2030 innovation project.

6.8 Plan resources so that text mining development can be parallel to other related activities

The seven above points illustrate the fact that a text mining project is quite complex, and its development requires an important amount of time and resources. In our opinion text mining development could last the entire period of a project such Polirural to ensure that project partners can make use of it in many different occasions.

7 Regional Library

The Regional Library is the main repository for Polirural documents. The following sub-chapter briefly describes how it has been built and the challenges and eventual improvements that can be made.

7.1. Needs library

In M1-M4 period KAJO created the framework for the Regional Library and collected more than 1400 new inputs from the twelve Polirural´s Pilots. Initially, the inputs included Related Websites, links to Social Media channels, Regional Landscapes, Key Search Words, Relevant Words and Crucial Needs totalizing more than 1400 new entries in the library (see Deliverable 4.1 for more details). These preliminary inputs have been used to create the basic architecture of the Regional Library.

7.2. Evaluation library

The second batch of sources include links to Websites and Social Media channels more focused on policy evaluation. By the time of this report the Regional Library contains more than 5000 links to websites, comprising HTML, .pdf and .doc files in the 12 languages of the project. This number is constantly growing and the KPI target of 1800 sources has been therefore fully achieved. The collection of documents in the Regional Library might be further extended with additional sources associated with activities in WP5 and WP4 related to Foresight and to policy evaluation. For this reason, the Regional Library will remain accessible by Polirural´s partners that will be able to add constantly new relevant resources.

7.3. Crawlers

In the Text Mining Technology, the content of the Regional Library is used automatically, receiving input for various text-mining pipelines. As those processing pipelines (and their context) will evolve over time, it is important to support the reprocessing of input in a consistent and effective way. Semex.io includes its own crawlers, but in a very limited version: it is capable of handling only well-written SGML resources (such as HTML or XML) and PDF files. Moreover, it can only obtain text from the provided page, not "looking" inside the structure of the website (so, links to other parts of the website are omitted). More advanced crawling solutions, extracting links from documents, have been tested but created too much stress on the system’s architecture and on the overall speed of the tool. Developers in fact realized that for every document an average of 20-40 new links would be added to the Regional Library, expanding excessively the size of the repository, and creating complications in the management of space and memory. Therefore, for the time being, it has been decided that it is more efficient to use the data included in the curated library provided by the experts

from the Pilots. Further research could be envisaged to develop a more advanced crawling system.

7.4 Social Media

Semex streams continuously messages from Twitter, extracting Tweets related to certain keywords and geographical regions. A long list of Keywords has been provided by the regional Pilots in the first months of the project and has gradually been adapted by WP2 researchers to obtain meaningful results. The keywords include a certain number of hashtags and Twitter users that have been considered relevant for Polirural research field. Partners and Regional Pilots, supported by WP2 developers, can add keywords if they need to stream some channel on Twitter. At the moment, the Tweets repository contains a big amount of text and shall continue to stream until the end of Polirural, if the system will allow it and also based on interest.

8 Infrastructure and deployment

As mentioned in Deliverable 2.1 the final set of services was about to be deployed on the Digitalocean platform.

However, during the testing stage of the TM process it became evident that the variety of solutions for storage of TM results is much wider than planned (taking into consideration the number of sources and their multilingual character). Therefore, the infrastructure was deployed on a completely different cloud platform that provides a more affordable solution for storing large corpus of data.

To increase the flexibility in development, testing, and deployment, we decided to implement a microservice architecture[25]. The idea of microservices is to split the architecture of a big service into multiple loosely coupled services that can be independently deployable. This enables the team to work on different parts of the project while simplifying the process of integrating the services into a larger infrastructure. This also enabled us to easily scale the services and assign them to specific server instances with low overhead as compared to a monolithic or manual installation approach.

There are three common distributed platforms when handling microservices, namely Kubernetes[26] , Apache Mesos[27], and Docker Swarm[28]. Kubernetes and Apache Mesos both require increased overhead for managing and configuring the cluster, whereas Docker Swarm is a much simpler variant. Docker Swarm was therefore our choice that offers management of the microservices with increased flexibility while being low in maintenance overhead. On single node clusters, it is also possible to use Docker Compose[29], which can be used for Docker Swarm by simply extending the configuration.

As for the time of preparing this document the main servers are operational and already able to get requests from then potential users, namely:

- main application server (user management, Library management, API to the core of Semantic Explorer features):
    - web-framework: Django 2.1 - Django is a high-level open-source Web Framework built on Python which we use as a backend for our web interface. Django is used in many well-known sites, such as Mozilla, The Washington Times and Instagram among others. It follows the model-template-view (MTV)

[25] Newman, Sam (2015-02-20). Building Microservices. O'Reilly Media. ISBN 978-1491950357
[26] [2] https://kubernetes.io/
[27] [3] http://mesos.apache.org/
[28] https://docs.docker.com/swarm/overview/
[29] https://docs.docker.com/compose/

architectural pattern, which enables faster development while maintaining a well-organized structure.

o web-server: Nginx (latest) - Nginx is an open-source and commonly used Web Server that also offers a reverse proxy, load balancer and HTTP cache among others. Nginx is used in this project to handle traffic to our Django web framework.

o core functionality written in: Python 3.7

o API engine: a Python library Tastypie (latest) - Tastypie is a web service API built for the Django web framework which we use for the REST-API endpoints.

o data streaming: MongoDB 4.2

o message queue: Celery (latest) with MongoDB 4.2 as backend - Celery is an open-source asynchronous and distributed task queue that we employ to schedule our NLP tasks.

o NLP & ML tasks: spaCy, Gensim, polyglot, scikit-learn

o front-end / visualization platform: EL Stack kibana, d3 of JavaScript, React - Our frontend consists of Kibana, D3js, and React. Kibana is a data visualization interface that is used to visualize and navigate through data in Elasticsearch. D3js is an open-source JavaScript library which is used for building advanced and interactive visualizations. Last but not least we employ the React, which is a commonly-used open-source web framework that simplifies and modularizes user interfaces on the web.

· database server for application data (users data, documents metadata, etc.):

o DBMS: MongoDB 4 - MongoDB is an open-source document-oriented NoSQL database. It allows easy storage and retrieval of JSON-like data and is also well-equipped to be scaled across multiple machines and handling large loads and throughput.

· cluster for indexed data for semantic analysis:

o database for indexed data: Elasticsearch 7.5 - Elasticsearch is an open-source search engine built on top of the Apache Lucene library. It offers high-speed full-text search. It is well integrated into the Elastic Stack with other tools like Kibana, Logstash, and the Beat framework among others.

All the aforementioned software is installed on the appropriate servers and the integration of the packages is tested. The deployed solution is ready to test the main features.

The current infrastructure model allows for quick re-organisation of services, adding new nodes in case of necessity and re-assigning already existing services to new virtual machines () VMs. See figure 34 for the diagram of the current infrastructure.

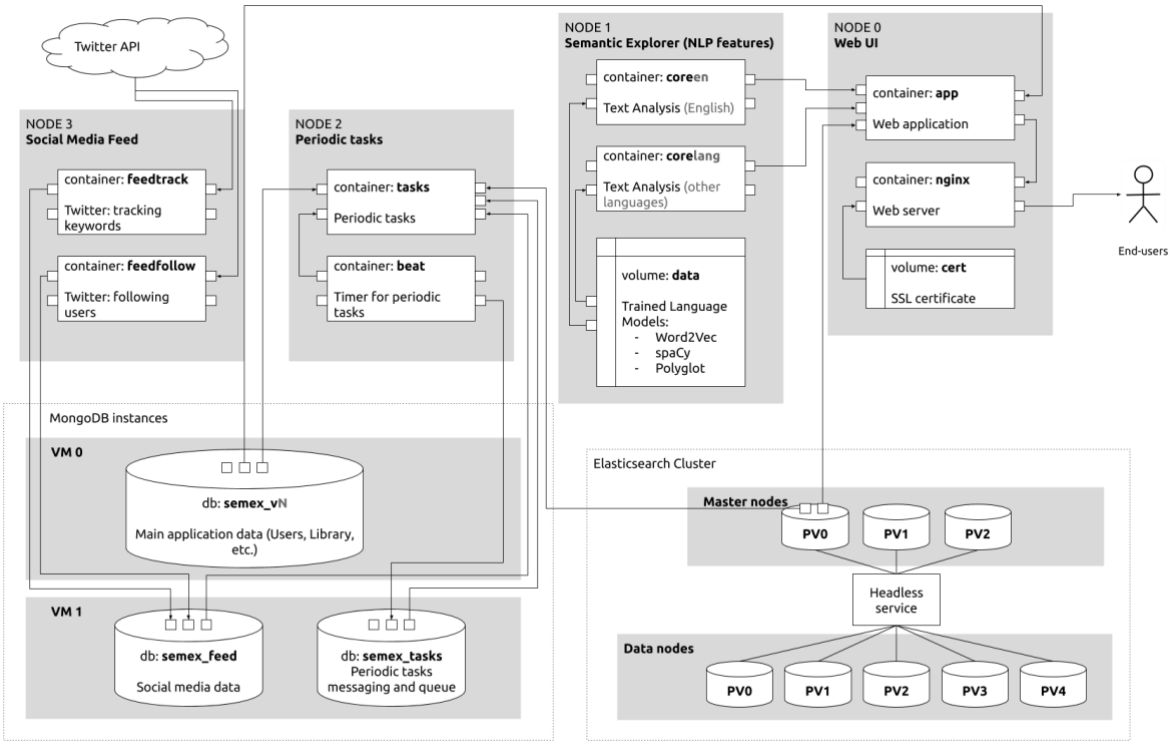

Figure 17 the current infrastructure

Conclusions

Achievements:

The development of semex.io has been a very productive research project that contributed to underlining the potential and the challenges of a text mining tool dedicated to activities related to policy-making, in particular in the field of rural development. The product created is a very powerful tool able to extract knowledge from unstructured data and communicate the results through effective visualisations. Semex.io supports researchers involved in policy evaluation, Foresight and other activities by reducing the cognitive load related to desk research tasks. This may contribute to empower policy-makers who will be able to access information faster and based on bigger data sets. Semex.io also fits in Polirural aim of creating an innovative and reusable concept "that draws upon participatory principles, stakeholder knowledge, big data, original research and advanced analytics to deliver more accurate foresight for rural regions, contributing to new and enhanced policy intervention" (Polirural Project Proposal). Plans already exist to reuse the synergies created in Polirural between Text Mining, Foresight and System Dynamics Modelling in other fields of research. At the same time rural development might benefit from the methodology described in chapter 6 for future projects.

It can be argued that Semantic Explorer has reached many achievements and created many interesting conclusions for future projects involving text mining in policy processes. Semex developers have been able to create a very complex text mining system that can analyse text in 10 different languages and includes the following features:

- Text summary
- Topic extraction
- Keyword extraction
- Named Entities recognition
- Sentiment analysis
- Geo-parsing
- Eventually the results of the research on trends

These features are linked one with the other in various pipelines and create effective data visualisations included in an analytical frontend application co-developed together with Pilots. Moreover, a very innovative feature of Semex is the possibility of creating Curated Reading Lists made of interesting sources and extracting aggregated results. We are not aware of any text mining tool with such a feature. What makes this solution particularly special is the fact that it has been conceived by various project's partners (especially from WP1, WP2, WP5) and is now widely used by Pilots for their research activities. It could be argued that it represents a good example of European cooperation.

Another important achievement of Semex, which is also the result of positive teamwork, is the Regional Library that includes more than 5000 sources related to rural areas and rural development in 10 European countries, plus Israel and North Macedonia. These sources are accessible publicly and can be consulted by researchers and eventually by interested policy-makers. It will be very important to publicize the tool in various events and through the planned MOOC so that the contents can reach a wider public.

General conclusions:

These solutions have been tested by ten out of the twelve Pilots on various case studies including policy evaluation and Foresight activities. If some Pilots have found the insights from TM ”interesting” and ”giving another perspective” others have underlined the potential of text mining but also the inefficiencies of the system developed. The jargon used in the policy related issues is very specific and TM does not always perceive the language subtle nuances. Moreover, its general efficiency greatly differ from one language to the other. The results obtained for English, Spanish, Italian, Greek and Dutch texts are much better than the ones obtained for Latvian, Finnish, Czech and Slovak. Also, these languages present a reduced volume of messages in social media, shrinking consistently the potential for text mining.

Finally, although the current infrastructure model allows for quick re-organisation of services, adding new nodes in case of necessity and re-assigning already existing services to new virtual machines, some fine tuning and continued support will be necessary. It is crucial to mention that the creation of the Text Mining solution does not end with the delivery of the finalized product. It is an iterative process not only in terms of its development, but also in its usage. Information to be analysed and processed is massive in amounts and complexity. The solution will need constant monitoring and maintenance.

Annex I - Access through open API for developers

In addition to the tools available via https://semex.io/, its data and semantic features can also be accessed via Application Program Interface (API) here. Below is the description of the main API endpoints and ways to acquire data making HTTP requests to them[30].

Semantic Explorer API

Root endpoint

To see all available resources, go to: api/v1/?format=json

Resources

Each Resource is represented by two endpoints: data and schema.

Resource

data: api/v1/<<resource_name>>/?format=json Example: api/v1/library/?format=json

Resource

schema: api/v1/<<resource_name>>/schema/?format=json Example: api/v1/user/schema/?format=json

Schema endpoint describes the structure of the response:

- allowed methods
- fields (types, options, etc.)
- possible filtering
- ordering

Representation

Resource's default mode of representation is a list of objects.

Every single object has a resource_uri attribute, which leads to a detailed representation of a particular object.

In the list mode meta container displays limit and offset (see parameters below), URLs for previous and next portion of data (in case limit and offset are used) and total number of records in the output.

Parameters

Format

```
api/v1/<resource_name>/param1_name=param1_value \
  &param2_name=param2_value& \
  ... \
  &paramN_name=paramN_value
```

Parameters that serve as filters, allow for modifiers. Each modifier can be applied to a field of a certain type:

- exact - equality: *any type*
- iexact - equality, case insensitive: *strings*

[30] In this document only a short version of the manual is given. The full version is being constantly updated. It can be found here - https://github.com/KajoServices/polirural-semex-doc/tree/master/API

- exists - *any type*
- startswith - *strings*
- istartswith - case insensitive "startswith": *strings*
- endswith - *strings*
- iendswith - case insensitive "endswith": *strings*
- contains - *strings*
- icontains - case insensitive "contains": *strings*
- match - matching pattern (can include *, e.g. racoon*): *strings*
- in - inclusion: *lists*
- nin - not in: *lists*
- lt - less than: *numeric values, dates, timestamps*
- lte - less than or equal: *numeric values, dates, timestamps*
- gt - greater than: *numeric values, dates, timestamps*
- gte - greater than or equal: *numeric values, dates, timestamps*
- ne - not equal: *strings, numeric values, dates, timestamps*
- all - must include all mentioned values: *lists*
- any - must include at least one of given values: *lists*

Format: paramname__modifier=value
Examples:
...&text__startswith=Rural
...&text__icontains=transport
WARNING: modifiers all and any can only be applied for the *ListFields* (such as *topics*), and in this case they should be divided by a vertical bar, for example the following param will return records that contain "land transportation" AND "livestock farming":
topics__all=land transportation|livestock farming
The following parameter will return records that contain either "land transportation" OR "livestock farming":
topics__any=land transportation|livestock farming
, and is equivalent to the following param:
topics=land transportation?topics=livestock farming
The list of available filters and their modifiers are available in each endpoint's schema.
Common parameters

- **format** - available formats: xml, json, yaml
- **username** - not a param, but a part of authentication token (together with **api_key**, see below). NB: this can also be sent as Authorization header.
- **api_key** - a part of authentication token (together with "username"). NB: this can also be sent as Authorization header.
- **limit** - limits number of objects returned. Applicable only in case of detailed reports. Default: 36. Example: ...&limit=100
- **offset** - number of records to skip from the beginning. Together with "limit" is used

to divide data to pages (pagination). Default: 20. Example: ...&offset=40

Endpoints

Library

List of Library Sources (GET)

api/v1/library/?username=username&api_key=api_key

Sorting

By default sources are sorted by the field *created_by* descending (latest first).

For custom sorting use parameter order_by followed by the name of the field.

Examples:

- Sort by owner (owners are users who create sources - in this particular case it will be sorted by username): api/v1/library/?order_by=owner
- Sort by language descending: api/v1/library/?order_by=-lang

Sorting by multiple fields

When sorting by multiple fields is required, the parameter *order_by* should be repeated for each field: api/v1/library/?order_by=-updated_at&order_by=-lang

WARNING: Order matters. Consider the following example: Sort by source_type, and then - within a set of each source type - sort

by created_at descending: api/v1/library/?order_by=source_type&order_by=-created_at

Sorting by Nested Fields

It is also possible to sort by fields that are represented as objects (for example, owner) - in the following example the output is sorted by the name of organization, which is represented by owner.

...&order_by=-owner__profile__organization__name

NB: Normally, "name" is redundant, since it is a default field that represents an organization. Here it used for completeness of the example.

Filtering

Use names of fields for filtering in the same manner as parameters (see "Parameters" above):

api/v1/library/?source_type=text/html

Filters can be combined:

api/v1/library/
?source_type=application/pdf
&lang=it
&created_at__gte=2020-02-18
&size_src__lte=500

The resulting list of objects can be sorted:

api/v1/library/
?source_type=application/pdf
&lang=it
&created_at__gte=2020-02-18

&size_src__lte=500
&order_by=-updated_at

It is possible to use multiple filters on almost all fields of Library (except of created_at, updated_at, meta_request, meta_src).
The following requests will produce the same result:
api/v1/library/?lang=it&lang=en
api/v1/library/?lang__in=it,en

Time Range Filters

In addition to the standard modifiers (__gt, __lte, etc.) filtering by date fields (created_at and updated_at) can be performed using time ranges. Time-range is a string that consists of two dates (start and end), divided by vertical bar (|):
api/v1/library/
&created_at=2015-05-08T10:00|2015-05-09T12:15
It is possible to use both date- and time-stamps as values for ranges, and to combine them in the same query:
api/v1/library/
&created_at=2015-05-08|2015-05-09T12:15
Time range can be specified in human readable format:
api/v1/library/
&updated_at=2 hours ago|now
It is possible to use other human readable keywords, e.g. "1 day ago", "January 12, 2017", "Saturday", etc.
Examples:
api/v1/library/
&updated_at=2020 Feb|yesterday

api/v1/library/
&created_at=1st of Jul 2012|in 2 hours

api/v1/library/
&created_at=2020 Feb|2020-03-05T12:15
&updated_at=2 hours ago|now

WARNING! In all given examples *two values* are necessary: start and end date (divided by vertical bar). The following example will cause **400 Bad Request**:
api/v1/library/
&created_at=1 day ago
If the goal is to filter the records for the last day, use the following request:
api/v1/library/
&created_at=1 day ago|now

Reserved keywords for Time Range Filters

Finally, there are reserved keywords, that don't require a pair of values: today, yesterday, this week, last week, this month, last month, this year, last year.

```
api/v1/library/
  ?created_at=last month
  &updated_at=yesterday
```

Filters based on time-ranges and reserved keywords are *inclusive*, i.e. they automatically stretch filters from the beginning of the starting date (0:00, or 12am) to the end of the ending date (23:59:59 or 11:59pm). So, the following examples are equivalent:

```
api/v1/library/
  ?created_at=2019-12-31|2019-12-31

api/v1/library/
  ?created_at=2019-12-31T00:00:00|2019-12-31T23:59:59
```

Landscapes and Regions

In Semantic Explorer territorial references represented by:

- Regions - an administrative unit as defined in Global List of Administrative Units (GAUL) level 1. Access the list of regions at api/v1/region/?username=username&api_key=api_key
- Landscapes - Pilot area "composed" of several regions, i.e. one Landscape can one or more Regions. The list of Landscapes can be accessed at api/v1/landscape/?username=username&api_key=api_key

Filtering and sorting are available in the same way as for Library Sources - see Sorting and Filtering.

Connections to Library and Organizations

Landscapes are connected to the Organizations, while Regions - to Sources from the Library. This creates a possibility for users from one Organization to connect sources to different Regions, not necessarily to the Regions they operate in.

Connections to external geoshape references

Regions are connected to GAUL via the field g2008_1_id - this is a unique ID of the geoshape in the GAUL reference that represents a certain administrative unit (for example, Nitra, Slovakia or Flanders, Belgium).

An additional field geoshape_id is pointing to the region in the geo-parser maintained by the developer (KAJO). Please contact suppport to get access to the geo-parser.

Landscape is more "abstract" geo-shape - it can either be locality or a region, or a county, or the whole country. Thus it is connected to the only to geo-parser maintained by KAJO via the field geoshape_id.

Reading Lists

A Reading List is a collection of sources (URLs) on a certain subject or point of interest.

Warning: only registered users can access API endpoints for Reading Lists.

Reading List vs. Library

Internally Reading List is a collection of URLs and therefore in principle it isn't connected to Regional Library. However, every time a new Reading List is created, the links in the field sources are being added to the library for the faster processing in the future.

Reading List processing work-flow

In order to fill Reading List with data, a series if asynchronous background processes take place:

- checking the presence of each resource is in the Library
- creating those that aren't present: crawling, extracting text, semantic analysis
- obtaining summary for each text and collecting it in a one big bag-of-sentences
- performing a summary analysis on the bag-of-sentences
- indexing a Reading List's text fields and performing semantic analysis on the resulting summary (NER, geo-parsing, topic extraction, polarity estimation, etc.)

Right after the creation of the Reading List the server only returns _id of the new document and its completion status ("completed": false). It is also accompanied by the ("pessimistically") estimated time in the field proc_time (in milliseconds).

When the processing of Reading List is over, the owner obtains email with the results and a direct link to the newly created document. Note that proc_time in the processed Reading List is set to the actual value of the time required to complete the process, and therefore is different from the estimated time.

To check if processing is finished is to periodically request a Reading List's _id: api/v1/crl/crl_id/?username=username&api_key=4f23...d3c4, (crl_id is a UUID returned after the creation) and check the value of the field completed which is se to true upon completion.

List representation of Reading Lists (GET)

Available Reading Lists can be obtained via api/v1/crl/?username=username&api_key=4f23...d3c4

Filtering and sorting are available in the same way as for Library Sources - see Sorting and Filtering.

Reading List detail (GET)

A detailed information of a selected Reading List can be accessed in the following way: api/v1/crl/crl_id/?username=username&api_key=4f23...d3c4 where crl_id is a UUID returned after the creation, or can be found in the list.

New Reading List (POST)

Example of creating a new Reading List (**warning**: URLs are fictional, use your own list of links):

```
POST api/v1/crl/?username=username&api_key=4f23...d3c4 \
--header 'Content-Type: application/json' \
--data-raw '{
   "name": "Rural Demographics",
   "description": "Demographic change",
```

```
  "sources": [
    "https://ec.eu/eurostat/statistics_EU",
    "https://cor.eu/en/engage/studies/Documents/The%20impact.pdf",
    "https://espon.eu/sites/default/Regions.pdf",
    "https://www.lulla.com/2018/12/22/the-challenge-of-rural-
depopulation/#4d74a7a81295",
    "http://landmobility.gr/",
  ]
}'
```

Updating CRL (PUT, PATCH)

To update only certain fields of a CRL a PATCH request should be issued:

```
PATCH api/v1/crl/?username=username&api_key=4f23...d3c4 \
--header 'Content-Type: application/json' \
--data-raw '{
  "sources": [
    "https://ec.eu/eurostat/new_statistics_EU",
    "https://cor.eu/en/engage/studies/Documents/TheGimpact.pdf",
    "https://espon.eu/sites/default/Regions.pdf",
    "https://www.lulla.com/2018/12/22/the-challenge-of-rural-
depopulation/#4d74a7a81295",
    "http://landmobility.gr/",
  ]
}'
```

A background processing will take place only if the field sources is changed - in this case the fields summary, keywords, and urls_ext are cleared up, and then the process will be repeated in the same way and order as after the creation of a CRL. In all other cases only field update takes place.

Warning: if the list of sources should be changed, all URLs must be provided - not only those that are added to existing list.

PUT request means full update, i.e. all fields will be re-written. If values for some required fields aren't provided in PUT, it will cause an error 400 Bad Request.

Force reprocess Reading List

Normally Reading List is being re-processed automatically only when its sources is changed. In case a manual re-processing is necessary, it is enough to PATCH a List with the filed completed set to false as shown below. This will start re-processing, even if the list of sources remained unchanged.

```
PATCH api/v1/crl/?username=username&api_key=4f23...d3c4 \
--header 'Content-Type: application/json' \
--data-raw '{
  "completed": false
```

}'

Deleting Reading List (PATCH and DELETE)

No users are authorized to delete a Reading List except of admins. For Reading List to disappear from the list of available documents, simply set their status is_active to "false".

```
PATCH api/v1/crl/?username=username&api_key=4f23...d3c4 \
--header 'Content-Type: application/json' \
--data-raw '{
   "is_active": false
}'
```

To display only active documents use the filter &is_active=true.

To delete a Reading List from the DB, use DELETE method:

```
DELETE api/v1/crl/?username=username&api_key=4f23...d3c4
```

Search

Search endpoint is available at the following URL: api/v1/search/?query=policy

Searches are performed by the text (or list of items) stored in the following fields (properties): text, text.<lang>, url, entities.

If search by a single field (or by several, but not all) is required, use match modifier on a chosen field(s):

```
api/v1/library/
        &text__match=tourist*
        &loc_name__match=flanders
```

Note that in the latter example, search will be performed by all text fields. If a boosted search by a language-specific field is necessary, use __match modifier directly on that field:

```
api/v1/library/
        &text.spanish__match=productividad agrícola
```

Search with filtering

Search can be combined with filters:

```
api/v1/search/
   ?query=agricultural biodiversity
        &created_at=last month
        &loc_country=Belgium
        &order_by=-created_at
```

All the rules applied to the filtering of Regional Library is applicable in this case. All possible options for filtering can be found in schema of the resource:

api/v1/search/schema?format=json

Search with filtering by Source Type

There is one particular field that requires a special mention: source_type. Its value explains the origin of the document and in combination with source_id is used to track the original document. The value in source_type always consists of three parts, each of which refers to module, application within the module and data model within application. For example,

consider the following fragment:


```
{
   "created_at": "2020-04-12",
   "updated_at": "2020-04-23",
   "resource_uri": "api/v1/library/5e92e8ba4dcefb2097391a17",
   "source_id": "5e92e8ba4dcefb2097391a17",
   "source_type": "app:sources:LibrarySource",
   "text": [
      "Migration of youth from <em>rural</em> <em>towns</em> to bigger cities due to lack of opportunities is a common phenomenon nowadays, resulting in the ageing of <em>rural</em> areas. Youngsters should feel themselves addressed by the affairs and future of their <em>towns</em> and the EU.",
      "They should be involved in dialogues, aiming to find solutions for challenges, hence making <em>rural</em> <em>towns</em> and the EU attractive."
   ],
   "topics": [
      "new town",
      "city",
      "inner city"
   ],
   "url": "https://europa.eu/regions-and-cities/programme/sessions/582_en"
}
```


In this fragment the value of the field source_type equals to "app:sources:LibrarySource", which can be read as follows:

- module: app
- application: sources
- data model: LibrarySource

The value of the field source_id points to the document within this scheme.

NB: This information is necessary only if you want to filter by source types (i.e. &source_type=feed:twitter:tweet). If you only want to reach the original document from the search results, the field resource_uri serves this purpose.

Possible values:

- app:sources:LibrarySource
- app:sources:Keyword
- app:sources:ReadingList
- app:accounts:Organization
- app:regions:Landscape
- app:regions:Region
- feed:twitter:tweet

Warning: This list is extendable!

Search with filtering by Regions

It is possible to filter search results by a certain region. If it is necessary to filter by administrative region (see Landscapes and Regions), the field loc_admin_region refers to the name of administrative region:

api/v1/search/?query=rural areas
&loc_admin_region=Vidzeme

Search by specific fields

It is possible to use specified fields for search query. If more than one field specified for a search phrase, for each of them param match should be added to the request:

api/v1/search/?query=agricultural biodiversity
&match=text
&match=description

The fields available for selection:

- text
- summary
- description
- url
- domain
- author

NB: If parameter match is specified, the search query will be applied to all fields mentioned above.

Wildcard search

For wildcard search use the star symbol (*):

api/v1/search/?query=agri*

api/v1/search/?query=agri*ral

Warning: wildcards can only be used for single words. If applied to phrases, the result of the query will be empty:

api/v1/search/?query=agri*ral policy

Aggregations (GET)

In the /search/ endpoint documents can be aggregated by any field. Aggregation must be defined as a dictionary in the format of Elasticsearch query for Bucket Aggregations without quotes.

The definition of aggregation differs from the other parameters by adding a prefix: agg__FOO. Instead of FOO there can be any string of alphanumeric symbols and underscore (_). This name is used to find the results of aggregation in the content of the response.

Aggregations can be combined with filtering and search keywords:

api/v1/search/
?lang=en
&query=land mobility

```
  &agg__topics={
    terms: {
      field: topics,
      size: 20
    } \
  }
```

NB: the indentation here and below serves the purpose of readability. The aforementioned request can be written like this:

api/v1/search/?lang=en&query=land mobility&agg__topics={terms:{field:topics,size:20}}

Aggregations can also be nested:

```
api/v1/search/
  ?lang=en
  &debug__query
  &query=land mobility
  &agg__topics={
    terms: {
      field: topics,
      size: 20
    },
    aggs: {
      agg__polarity_scores: {
        histogram: {
          field: polarity,
          interval: 0.5
        }
      }
    }
  }
```

It is possible to define more than one aggregation in a single query - however it isn't recommended, because aggregation is a time-consuming operation and therefore can either slow down the responses to other queries or simply result in a long response time.

```
api/v1/search/
  ?lang=en
  &query=land mobility
  &agg__topic_groups={
    terms: {
      field: topics,
      size: 20
    },
    aggs: {
```

```
        agg__polarities: {
          histogram: {
            field: polarity,
            interval: 0.5
          }
        }
      }
    }
    &agg__polarity_scores={
      histogram: {
        field: polarity,
        interval: 0.5
      },
      aggs: {
        agg__topics: {
          terms: {
            field: topics,
            size: 20
          }
        }
      }
    }
```

If any of the aggregation parameters appear in the request, the response contains additional field “aggregations”, where a summarized number of documents (or other specified aggregations) are gathered in “buckets” and sorted accordingly.

NB: In the example the names agg__topic_groups, agg__polarities, agg__polarity_scores and agg__topics are user-defined. They should be properly named keys for a JSON format (a-z, A-Z, 0-9, underscore, no spaces). and those are the names of sections in the response with results of aggregations.

If you are interested in aggregated results only, it is possible not to include original documents entirely by setting size param to zero (see example below). In this case the field objects will still be present in the response to comply with GeoJSON format, but it will be an empty list.

```
api/v1/library/
  ?&agg__name=<...>
       &size=0
```

Date-time histogram with average sentiment

In Semantic Explorer database sentiment is stored in the field polarity. Date-time histogram with average values of polarity is being obtained on the following way:

```
api/v1/library/
```

?topics=rural population
&topics=rural attractiveness
&country=Ireland
&agg_polarity=true
&agg_polarity__interval=90m

This indicates calculation of the average sentiment for each timestamp bucket. In the example above in each bucket (1.5hrs long) in addition to doc_count will contain avg_polarity.

User feedback

In cases when results of Semantic Analysis are in some way unsatisfactory, it is possible to leave a feedback.

The comment will automatically be stamped with a date-time mark and a link to a user profile.

Warning: only registered users can leave feedbacks!

The following cases are supported:

- wrong topic in text
- wrong entity text
- incorrectly defined label for an entity
- incorrect entity annotation
- incorrect estimation of polarity

In case of wrong topic:

```
POST /api/v1/feedback/?username=username&api_key=4f23...d3c4 \
  --header 'Content-Type: application/json' \
  --data '{
     "text": "Logging messages are encoded as instances of the LogRecord class. When a logger decides to actually log an event, a LogRecord instance is created from the logging message.",
     "lang": "en",
     "feature": "topic",
     "value": {"text": "residential area"},
                "action": "delete",
          "reason": "irrelevant"
  }'
```

Feedback is a text-oriented, but it is also possible to leave a feedback for any particular paragraph in any document. In this case data in the previous example should take the following form:

```
{
  "source_type": "app:readlst:ReadingList",
  "source_id": "5ffefe21e97f91b6bc72fe6b",
  "para_order": 0,
```

"text": "Logging messages are encoded as instances of the LogRecord class. When a logger decides to actually log an event, a LogRecord instance is created from the logging message.",
"lang": "en",
"feature": "topic",
"value": {"text": "residential area"},
"action": "delete",
"reason": "irrelevant"
}

Warning: feedback is a paragraph based. It is impossible (and useless) to leave a feedback for the whole document. Therefore all three fields are
mandatory: source_type, source_id and para_order, otherwise API will return an error 400.

Reasons

reason is a free-form text limited to 255 characters. The three basic reasons are: "incorrect", "irrelevant" and "outdated".

Required parameters

Different features (topic, entity, etc.) require different set of parameters in the value field:

- topic: "text"
- entity: "text", "label", "start_char", "end_char" (and optionally - "annotation", i.e. should repeat the structure of the Entity object)
- label: the same as entity
- annotation: the same as entity
- polarity: "polarity"

Actions

There are three available actions: add, delete, and replace, of which replace requires another one field in the structure: replace, which should repeat the structure of the field value.

Example - replacing entity's text (**warning**: in such cases start_char and end_char should also be updated):

```
POST /api/v1/feedback/?username=username&api_key=4f23...d3c4 \
  --header 'Content-Type: application/json' \
  --data '{
    "text": "Kultūras ministrijai sagatavot un kultūras ministram līdz 2021. gada 1. martam iesniegt noteiktā kārtībā Ministru kabinetā ORG informatīvo ziņojumu par plāna izpildi.",
    "lang": "sk",
    "feature": "entity",
    "value": {
      "text": "kabinetā",
      "annotation": "",
      "label": "ORG",
      "start_char": 114,
```

```
      "end_char": 122
    },
    "replace": {
      "text": "Ministru kabinetā",
      "annotation": "",
      "label": "ORG",
      "start_char": 105,
      "end_char": 122
    },
    "action": "replace",
    "reason": "incorrect"
  }'
```

Semantic Analysis

All endpoints described above provide access to data that has already been analyzed by the Semantic Explorer and saved in the database or indexes in the search engine. In addition to that there is a set of endpoints that provide direct access to the semantic features on the text.

Request type and structure

Semantic features can be accessed directly only by POST requests.

General form of the request (example made with curl command):

```
curl --request POST 'api/v1/analyze/?username=<username>&api_key=<api_key>' \
--header 'Content-Type: application/json' \
--data-raw '{
  "text": "<text>",
  "pipeline": [<item1>, <item2>, ..., <itemN>],
  "lang": "<lang>"
}'
```

Only registered users can access semantic features, therefore absent or wrong username and api_key will generate 401 Unauthorized error.

Warning! It is very important to set a correct language, if it is known. If the language is not provided, the system will detect it, but this would cost an overhead. If the language is set incorrectly, Semantic Explorer will use it for analysis, which can make outcome quite surprising.

The Pipeline

The main parameter of Semantic Analysis endpoint is the pipeline. It is a list of strings or dictionaries that describe what kind(s) of analysis should be performed on the text.

Each element in the list can either be a string (name of action) or a dictionary (name of action with parameters).

Example of plain list (actions are performed with the default parameters):

```
{
```

```
  "text": "<text>",
  "lang": "<lang>",
  "pipeline": ["tokens", "ner", "topics"]
}
```

To change the default parameters of each action use the following form:

```
{
  ...
  "pipeline": [
    {
      "action": "tokens",
      "params": {
        "lemmatize": true
      }
    },
    {
      "action": "vectors",
      "params": {
        "topn": 50
      }
    }
  ]
}
```

It is possible to combine those two forms in the same call:

```
{
  "text": "<text>",
  "lang": "<lang>",
  "pipeline": [
    "tokens",
    {
      "action": "vectors",
      "params": {
        "topn": 20
      }
    },
    "topics"
  ]
}
```

Actions

Every action is a singular procedure that is performed on a given text. Actions are grouped in the chain and each of them affects results on the later stages in the chain. For example if

both tokens and vectors are requested and the action tokens was accompanied by param lemmatize=true (see below), then vectors will return word vectors of lemmas instead of vectors of original words. All dependencies and parameters are described for each action below.

Semantic Explorer supports the following actions:

tokens

Text cleaned up from stop-words, special symbols and punctuation marks.

Parameters:

- lemmatize <boolean> (default false) - if it is set to true, each token will be represented by its lemma: https://en.wikipedia.org/wiki/Lemma_(morphology)

vectors

L2 norm vector of words found in text - for details see https://spacy.io/usage/vectors-similarity.

Parameters:

- topn <int> (default 10) - top N words sorted by normalized value of the vector

ner

Named Entity Recognition https://en.wikipedia.org/wiki/Named-entity_recognition.

Parameters:

- distinct <boolean> (default false) - if true Named Entities are grouped for the whole text and only labels and text of each entity is returned, otherwise all entities with their appearance in the text (accompanied by start_char and end_char). Entity labels are described here https://spacy.io/api/annotation#named-entities

ner_rendered

HTML formatted version of ner. No parameters.

noun_chunks

Objects and subjects of all sentences of the text - the result of Parts of Speech tagging https://spacy.io/usage/linguistic-features#pos-tagging. No parameters.

polarity

Estimated value of sentiment on the scale from -1.0 to 1.0, where values around -1.0 are very negative, close to 0.0 are neutral, and close to +1.0 are very positive.

Parameters:

- level <str> of analysis. Possible values:
  - text or t (default) - get sentiment for the whole text
  - paragraph or p - split text to paragraphs and return estimation of sentiment for each paragraph
  - sentence or s - split text to sentences and return estimation of sentiment for each sentence.

summary

Text summary, i.e. the most representative sentences from the text.

Parameters: - keywords <int> (default 15) - defined how many top keywords are used to

summarize the document (the bigger the number of keywords, the broader the summary).
- sentences <int> (default 5) - how many sentences should be returned.

topics

Extraction of topics as if answering the question "what this text is about, in general?" This is highly dependent on the thesaurus - current implementation of topic extraction in semex.io is based on GEMET https://www.eionet.europa.eu/gemet/en/about/.

Parameters:

- keywords <int> (default 15) - same as in summary
- max_topics <int> (default 10) - the maximum number of topics to be returned (the response is always ordered by the relevance of the topics descending - the most relevant at the top)

Inputs

There can be three types of input:

- text - plain text of an arbitrary length.
- url - a proper URL pointing to some text in the internet. In this case the URL is first scraped (which adds overhead to the response depending on the source size)
- source_id - a unique ID of the source from the Regional Library (this is generally not necessary, since all the documents from Regional Library are being constant;y updated and analyzed).

Example of analyzing a text:

```
curl --request POST 'api/v1/analyze/?username=<username>&api_key=<api_key>' \
--header 'Content-Type: application/json' \
--data-raw '{
   "text": "The productivity of a region'\''s farms is important for many reasons. Aside from providing more food, increasing the productivity of farms affects the region'\''s prospects for growth and competitiveness on the agricultural market, income distribution and savings, and labour migration. An increase in a region'\''s agricultural productivity implies a more efficient distribution of scarce resources. As farmers adopt new techniques and differences, the more productive farmers benefit from an increase in their welfare while farmers who are not productive enough will exit the market to seek success elsewhere",
   "pipeline": [
         "polarity",
         {
         "action": "topics",
         "params": {"max_topics": 3, "keywords": 15}
         }
   ],
   "lang": "en"
}'
```

Example of analyzing URL:

```
curl --request POST 'api/v1/analyze/?username=<username>&api_key=<api_key>' \
--header 'Content-Type: application/json' \
--data-raw '{
        "url": "https://en.wikipedia.org/wiki/Agricultural_productivity",
   "pipeline": [
        {
                "action": "summary",
                "params": {
                        "sentences": 6
                }
        },
        "noun_chunks"
   ],
        "lang": "en"
}'
```

Example of analyzing a document from the Regional Library:

```
curl --location --request POST 'api/v1/analyze/?username=<username>&api_key=<api_key>'
\
--header 'Content-Type: application/json' \
--data-raw '{
   "source_id": "5df73a3d5ae4c68142634e9f",
   "pipeline": [
        "tokens",
        {
                "action": "ner_rendered"
        },
        {
                "action": "topics",
                "params": {
                        "max_topics": 5
                }
        }
   ],
   "lang": "en"
}'
```

Similarity Cluster

Similarity Cluster is a general model based on thesaurus, where topics and subtopics are represented as a directed graph with the topic as a root and its subtopics as nodes. If a node has children, it is considered to be a topic on its level, while his children are subtopics.

NB: It is only a representation up to a certain number of levels (normally 3) that can be

considered as a directed graph. In reality, the graph is non-directed because the root topic can be a subtopic of the topic on a deeper levels. Similarity cluster is a dynamic structure and is being built in real-time, which allows to add new topics at any time.
Similarity Cluster for a particular topic can be obtained by GET request:
api/v1/similarity_cluster/
?topic=zonas rurales
&lang=es
&threshold=0.6
&depth=3
&username=<username>
&api_key=<api_key>
Parameters:

- topic <str> (mandatory)
- lang <str> (optional, default **'en'**) - if the topic is in a language different than English, it SHOULD be accompanied with a correct language code!
- depth <int> (optional, default 2) - how deep to dive (**warning:** the deeper, the longer the response!):
    - 1 find only nearest neighbors
    - 2 return subtopics for each subtopics of the main topic
    - 3 one level deeper,
    - 4 etc.
- threshold <float> (optional, default 0.6) - minimal similarity to consider a subtopic (internally - a distance between normalized vectors of topic and subtopics).
- topn <int> (optional, default 10) - how many similar topics should be returned. NOTE: the first node has topn subtopics, but the nodes in further levels have topn-1 subtopics.

Topic browser
Endpoint that connects Similarity Cluster and data in Regional Library and/or Curated Reading List and/or Social Media feed. In other words, for the defined set of subtopics it returns documents from the selected resource(s) (for example, from Regional Library and Social Media feed) that are categorized by those topics.
GET request:
api/v1/topic_browser/
?root=rural area
&topic=urban area
&topic=rural population
&topic=rural environment
&lang=en
&source_type=app:sources:LibrarySource
&topn=10

&username=<username>
&api_key=<api_key>

Parameters:

- root <str> (mandatory) - root topic from Similarity Cluster
- topic <str> (multiple, at least 1 is mandatory) - can be any topic different from the root, but ideally it's subtopics.
- lang <str> (optional, default "en") - if the topic is in a language different than English, it SHOULD be accompanied with a correct language code!
- source_type <str> (multiple, optional, default "app:sources:LibrarySource") - source of documents. Options to choose from :
    - app:sources:LibrarySource
    - app:sources:Keyword
    - app:sources:ReadingList
    - app:accounts:Organization
    - app:regions:Landscape
    - app:regions:Region
    - feed:twitter:tweet
- topn <int> (optional, default 10) - how many documents from each source should be returned.

Annex II – Semex.io Users' Manual

Semex.io User's Manual is enclosed as an attachment to this deliverable.

Annex III – Responses to the monitors' comments

Comment made by monitors	Explanation
WP2 is expected to end in M18; no details on the extraction of messages from the Library (e.g. policy recommendations from the national midterm reviews of EU-cofounded policy programmes for WP1;	The deliverable now includes Chapter 5.1 on the exercise made by Pilots about policy evaluation with information related to policies (mainly LEADER). In fact it was decided in WP4 that Pilots should concentrate on one policy.
No substantial involvement of WP1	WP1 has been involved in the composition of the regional Library (Ch. 6); in the Training and Evaluation phase (Ch. 4); in the use cases (Ch. 5).
Critical reflection on usefulness and applicability of TM	This is done in various parts of the deliverable, in particular in Subchapters 5.1.1 and 5.6
Upcoming ex-ante evaluation of policies in the pilot areas is missing (problem of DoA and PM allocation)	Such an activity has not been agreed in the GA. WP2 has provided the text mining tool to WP1 and WP4 so that they could analyse what they deemed relevant. In WP4 it was decided to focus policy evaluation on one policy.
More elaboration on this (p.56): At the same time rural development might benefit from the methodology described in future projects.	A more detailed methodology is described in Ch 6
Executive summary does not refer to the problems highlighted in D1.6 with TM not available for the use in WP1 in year1); problems must have occurred;	Explanation is now included in the executive summary
Kajo has spent 20 out of 24PM, all partners have only 1PM in WP2; S&L Hub has spent more than expected (145% of PM budget explained by "poor planning"); most partners underspent in Y1 and will use TM in M13-M18 for policy evaluation but no more for the needs assessment. The training	The assumption that a text mining tool in 12 languages for activities related to policy process can be ready in 12 months, especially at the beginning of the project when there are not yet clear directions, was probably too optimistic. Partners had the possibility of using the tool for practical use cases from M12 until now (M21) and will be still use it for other activities such as Foresight or other research tasks.

and evaluation will start after the finalisation of the report.	
The value of TM for rural and regional development has not been sufficiently elaborated and remains unconvincing. There are therefore concerns about the suitability of TM in the pilots. Experiences from the application of current AI tools show that the analysis of complex documents such as policy or other regional development documents requires human reading. For that reason, the limitations of the TM for the work in the pilots needs to be highlighted.	A critical reflection about these thoughts has been added in various chapters of the deliverable, including the Executive Summary. Suitability of text mining for policy processes activity has its pros and cons as described in the deliverable. The main cons are probably linked to different languages rather than to the specificities of policy related activities. However, a general conclusion is that text mining can support researchers in their tasks but cannot replace human work.
The usefulness of further technical developments for the TM needs a critical assessment taking into account the adjustments made in the wake of the review in September 2020.	Technical development continued based on feedbacks received from Pilots and thanks to the collaboration kept with WP1 leader who was an active part of the final development phase.