DOI: 10.55612/s-5002-060-006

Lib2Life – Digital Library Services Empowered with Advanced Natural Language Processing Techniques

Melania Nitu[1], Mihai Dascalu[1,2], Maria Dorinela Dascalu[1],
Laurentiu-Marian Neagu[1], Maria-Iuliana Dascalu[1]

[1] National University of Science and Technology Politehnica Bucharest, Faculty of Automated Control and Computers,
Splaiul Independenței 313, 060042, Bucharest, Romania
[2] Academy of Romanian Scientists, Str. Ilfov, Nr.3, 050044, Bucharest, Romania
{suzana_melania.nitu, mihai.dascalu, dorinela.dascalu, laurentiu.neagu, maria.dascalu}@upb.ro

Abstract. Educational institutions are struggling to keep up with the accelerated technological advancements; hence, sustainable and supportive tools have become essential to reshape traditional models into intelligent learning systems. This paper introduces *Lib2Life*, a digital library that uses advanced Natural Language Processing techniques to facilitate the digital transformation of historical documents provided by Central University Libraries in Romania. The platform enables Central University Libraries in Romania to preserve the cultural heritage of historically valuable documents, facilitating open-source access to old, printed materials such as books, manuscripts, newspapers, or literary magazines no longer protected by copyright. *Lib2Life* offers comprehensive functionalities, allowing librarians to benefit from automated text processing and indexing workflows that facilitate digitization, ensuring a consistent representation of original documents. For readers, the platform presents a user-friendly interface with semantic search capabilities and a recommendation engine. The system employs an ontology to organize and manage documents in a unified and structured way, contributing to the evolution of intelligent education technologies. The innovative contributions of *Lib2Life* include identifying new solutions for cultural heritage preservation, promoting patrimony through modern methodologies, increasing access to documentary resources, enhancing library services, and fostering the transfer of knowledge and technology to society.

Keywords: Digital Library Service, Natural Language Processing, Semantic Search, Ontology Representation, Domain Categorization, Cultural Heritage Preservation.

1 Introduction

In an era of accelerated digital transformation, innovative technologies have become essential in bridging the gap between traditional and modern library services. The world has witnessed a remarkable shift from conventional libraries to digital repositories in search of more sustainable and accessible forms of education. Digital change is marked by the transition from static data repositories to dynamic, interactive, and intelligent platforms. This evolution has revolutionized how information is accessed and has become essential for preserving the world's cultural heritage. The digitization of physical documents is crucial to ensuring the accessibility, longevity, and sustainability of historical and educational resources. Preserving valuable manuscripts, books, and documents is an act of historical importance and a testament to a nation's commitment to its cultural legacy [1].

Romania is well-known for its vibrant culture and literary contributions and is not immune to this transformative wave. Central University Libraries across the country are home to over two million physical volumes, representing centuries of knowledge and culture. Thus, the need for intelligent technologies becomes indisputable to exploit this vast source of knowledge. In this context, we introduce an intelligent digital library platform, *Lib2Life*, powered by advanced Natural Language Processing (NLP) techniques, designed to support sustainability in education by digitizing and preserving the physical collections of Romania's Central University Libraries. Sustainability in education means ensuring that knowledge remains accessible, relevant, and up-to-date without compromising the resources that feed it. *Lib2Life* [2] is designed to have this vision as foundation while striving to make educational resources sustainable and easily accessible. With its complex suite of features, from advanced document processing [3] and semantic search capabilities to an intelligent ontology-based recommendation [4, 5] and representation [6] module, *Lib2Life* facilitates open access to legacy documents [7].

This paper aims to illustrate how the mix of technology, education, and culture can support smart education in Romania through advanced NLP techniques and digitized libraries. In terms of structure, the second section explores related systems and research in the field. The third section presents the architecture of the *Lib2Life* platform, the data sources, the *Lib2Life* ontology, the classification of documents and instantiation of the corresponding individuals in the ontology, the functionalities and workflows, the web services, and, last but not least, the web interface. Section four focuses on presenting the results, followed by conclusions and further directions.

2 Related Work

An important reference for digital libraries and literary preservation is Project Gutenberg[1] [8], founded by Michael Hart in 1971, which represents one of the first initiatives to ensure global open-source access to literary resources and preserve the world's cultural heritage. In 1971, Hart began by entering the United States Declaration of Independence into a computer system at the University of Illinois, which is often considered the inaugural eBook. This event marked the beginning of Project Gutenberg. Over time, the project expanded its scope and its commitment to digitizing and archiving literary texts for the general public, which became its foundational principle. Project Gutenberg has evolved into a vast and ever-expanding digital library that

[1] https://www.gutenberg.org/, last accessed 15/10/2023.

provides worldwide users with an extensive collection of literary works with the mission of universal access to literature and knowledge.

Europeana[2] is a platform that combines cultural heritage content from European museums, galleries, libraries, and archives. It provides access to a wide range of cultural and historical materials.

Google Books[3] is a comprehensive digital library and book digitization project initiated by Google in 2004, representing a vast repository of digitized books across a broad spectrum of fields, languages, and periods. One of its main goals is to make knowledge more accessible and discoverable for people around the world by providing users with a database of books they can search to preview, read, or buy online. The search engine allows users to search for keywords or phrases throughout the book collection. The collection includes various resources, from academic and scientific publications to historical texts, novels, and reference books. While some books are available in full text, others are accessible in a limited preview, snippet view, or simply as bibliographic information. The collection also includes magazines and journals.

The Internet Archive[4] [9] is an American digital library whose mission is to provide open access to all knowledge. Its document collection includes over 38 million printed materials, 12 million audiovisual content, 832 billion websites, and over 2 million software applications. The platform allows users to upload and download unlimited materials; however, its main collection is automatically acquired by web crawling. The Internet Archive's collection of books hosts materials from different sources, including partnerships with libraries and Google Books. Moreover, it includes the Wayback Machine[5], which allows viewing archived versions of web pages. Additionally, Open Library[6] is an initiative powered by the Internet Archive that provides access to millions of books.

HathiTrust[7] is a partnership of academic and research libraries that provides a digital library for copyrighted books, including academic and historical works. Similarly, the MUSE[8] project is a platform that offers academic books and journals in the humanities and social sciences. MUSE provides access to a wide range of academic publications. JSTOR[9] is another platform focused on academic content, including academic journals, books, and primary source materials in various disciplines. The Library of Congress[10] offers an extensive collection of digital resources, including historical documents, manuscripts, maps, and books. Their online catalog provides access to a wide range of materials.

At national level, existing solutions for accessing digitized content provide only basic functionalities. It is crucial to acknowledge that library systems currently in use rely on outdated software with restricted capabilities, highlighting the urgent need for modern technological enhancements. For instance, the management system of the

[2] https://www.europeana.eu/en, last accessed 15/10/2023.
[3] https://books.google.com/, last accessed 15/10/2023.
[4] https://archive.org/, last accessed 15/10/2023.
[5] http://web.archive.org/, last accessed 15/10/2023.
[6] https://openlibrary.org/, last accessed 15/10/2023.
[7] https://www.hathitrust.org/, last accessed 15/10/2023.
[8] https://muse.jhu.edu/, last accessed 15/10/2023.
[9] https://www.jstor.org/?typeAccessWorkflow=login, last accessed 15/10/2023.
[10] https://loc.gov/, last accessed 15/10/2023.

Central University Library in Bucharest is based on the Vubis [10] catalog system for inserting metadata and Restitutio[11], which is built on the DSpace [11] foundation.

On top of the library management systems, recommendation techniques are frequently applied to predict users' preferences and help retrieve similar documents. In addition to computing similar documents based on vectorial representations, some systems consider user profile data, such as the preferences of other users and the temporal dimension (users' preferences change over time), to provide recommendations [12].

The related work stands as a foundation for the *Lib2Life* platform, which was designed considering the requirements of modern readers and librarians, acting as a single point of access for multiple document sources. The main capabilities of the platform are document centralization, semantic search, similar document recommendation, and automatic document classification powered by Transformer-based models and knowledge graph population.

3 The *Lib2Life* Platform

Lib2Life is a web platform designed to provide access to old, printed documents that only exist in physical form, such as books, manuscripts, newspapers, or literary magazines, dating from the 18th century to the present, whose copyright has expired. *Lib2Life* initiated a partnership with four Central University Libraries in Romania to scan physical books, with an emphasis on preserving the original content and formatting. Advanced scanning technologies have been used to capture images of the pages, which are then processed using Optical Character Recognition (OCR) software to create searchable and machine-readable text. Following this step, the platform integrates advanced document processing and indexing mechanisms. Along with document management, the platform implements a semantic search engine and a semantic recommendation system that uses advanced NLP technologies. The application integrates the *Lib2Life* ontology [6] populated based on automatically classifying books in the corresponding domain using language models [4].

3.1 Architecture Overview

The *Lib2Life* platform consists of several components, and its architecture is presented in detail in Fig. 1. *Lib2Life's* modular architecture illustrates a coherent workflow that begins with the digitization of physical documents, followed by the integration of semantic models to improve the search for similar documents and the creation of a user-friendly web portal, where users can access the digitized materials. The *Lib2Life* platform is designed to meet the requirements of university libraries comprehensively and consists of several key components, namely: a) data sources for creating and storing digitized content and metadata (collection of physical documents and metadata database); b) document preprocessing, semantic models and ontology (document preprocessing pipeline and search engine, classification models and ontology

[11] http://restitutio.bcub.ro/, last accessed 15/10/2023.

representation for domain classification); c) digital services to ensure accurate and high-quality digitization (document uploading and processing, domain inference, document editing and preview, semantic search and similar document recommendations); d) a user-friendly web interface that allows users to easily interact with the system by browsing, searching, and accessing the content of the digital library.

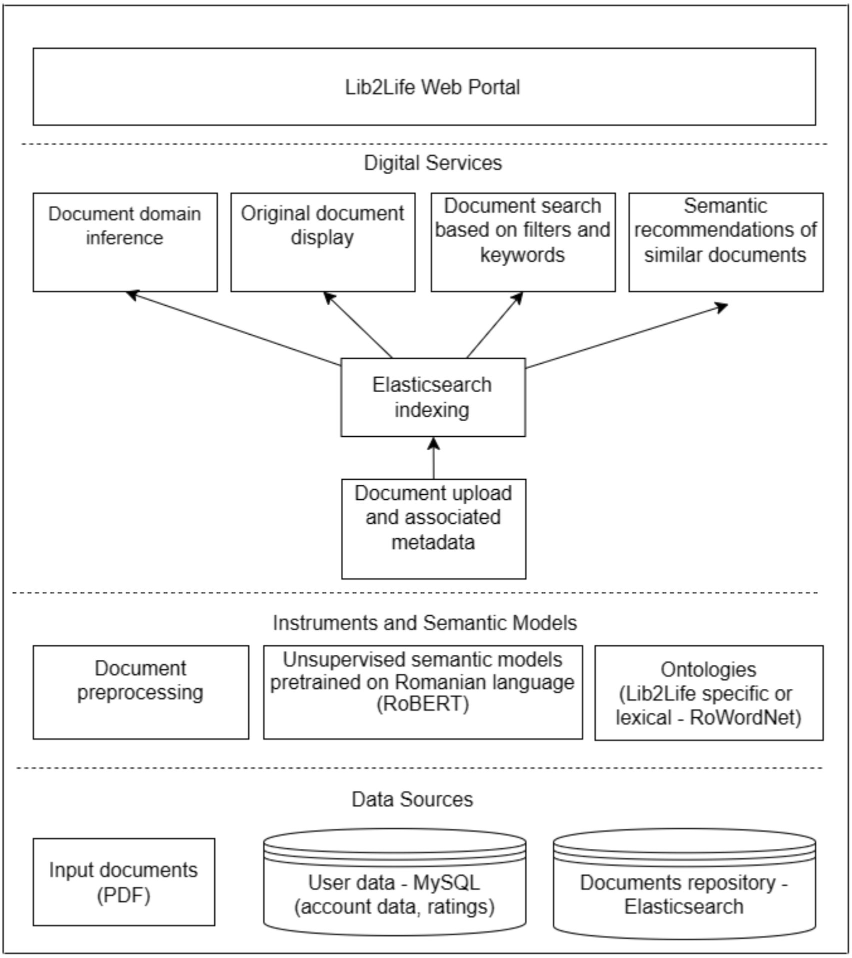
Lib2Life Web Portal
Digital Services
Document domain inference
Original document display
Document search based on filters and keywords
Semantic recommendations of similar documents
Elasticsearch indexing
Document upload and associated metadata
Instruments and Semantic Models
Document preprocessing
Unsupervised semantic models pretrained on Romanian language (RoBERT)
Ontologies (Lib2Life specific or lexical - RoWordNet)
Data Sources
Input documents (PDF)
User data - MySQL (account data, ratings)
Documents repository - Elasticsearch

Fig. 1. Lib2Life architecture.

This architecture aims to provide a comprehensive, user-centric, and technologically advanced solution for digitizing and promoting the invaluable collections of Central University Libraries in Romania. The components are further described in the following sections.

3.2 Data Sources

The input data are PDF files representing scanned images of the original printed documents provided by the Central University Libraries in Romania. The librarians manually scanned the physical documents, and occasionally, the quality and resolution of the scanned images were inadequate, leading to challenges in the subsequent processing stages. Optical Character Recognition (OCR) was applied to the documents to convert scanned images into extractable text. Based on the information held by the libraries in their internal document management systems (such as Vubis [10]), an XML file with various extracted metadata was attached to each document. The XML documents were integrated into the OCR process as an additional metadata attribute passed in JSON format. Documents were stored in Elasticsearch[12], processed, and split into paragraphs before indexing. Semantic search and recommendation engines use the average of the vectors computed at the paragraph level.

An additional data source is represented by the user data, which includes personal information (i.e., first name and last name), information used in the authentication process (i.e., e-mail, password), and the user's role within the system (i.e., reader, librarian, or administrator), as well as the ratings given to the books. This information is stored in a MySQL[13] relational database.

3.3 Document Processing Pipeline

A central component is represented by the document processing pipeline, which incorporates techniques for identifying and reconstructing paragraph boundaries, merging hyphenated words, performing grammar corrections, and leveraging algorithms for image extractions. Text processing and document categorization rely on NLP techniques. Two types of ontologies were used: a lexicalized ontology, which stores words and relationships between them (i.e., RoWordNet[14]), and a specific ontology (i.e., the *Lib2Life* ontology [6]), which holds information about the hierarchy of classes and the membership of individuals.

Documents are uploaded in PDF format. Processing is performed automatically and asynchronously by the system. Once the processing step is completed, the user can modify the metadata automatically populated from the XML file or inferred from the text, as well as the document's textual content. Inserting a new document into the system involves three stages: uploading the document, editing metadata and content, and computing the semantic representation for displaying similar documents. The document upload takes as input scanned books and documents in PDF format, on which an Optical Character Recognition (OCR) algorithm is applied.

In some cases, the OCR algorithm had to be improved to enable the correct text extraction from various types of document formatting. Issues reported include fonts of different types and sizes identified in the same paragraph or even on the same line of text, different styles of headers and footers within the same document, abrupt paragraph breaks, improper page breaks, loss of content structure, or rendering problems, as well

[12] https://www.elastic.co/elasticsearch/, last accessed 15/10/2023.
[13] https://www.mysql.com/, last accessed 15/10/2023.
[14] https://github.com/dumitrescustefan/RoWordNet, last accessed 15/10/2023.

as misspelling of certain characters or hyphenated words. The challenge is the existing text processing systems are not designed to work with OCRed PDFs [3]. Thus, the issues required a workflow to identify and match section headings with their content, recognize paragraph boundaries, merge hyphenated words, and accurately identify and extract images or tables.

The workflow for processing PDF documents is detailed in Fig. 2. Besides document scanning, OCR, uploading, and processing, the document is indexed in an Elasticsearch database to facilitate searching for relevant resources based on keywords and filters. Similar documents are suggested based on vectorial representation and cosine distance.

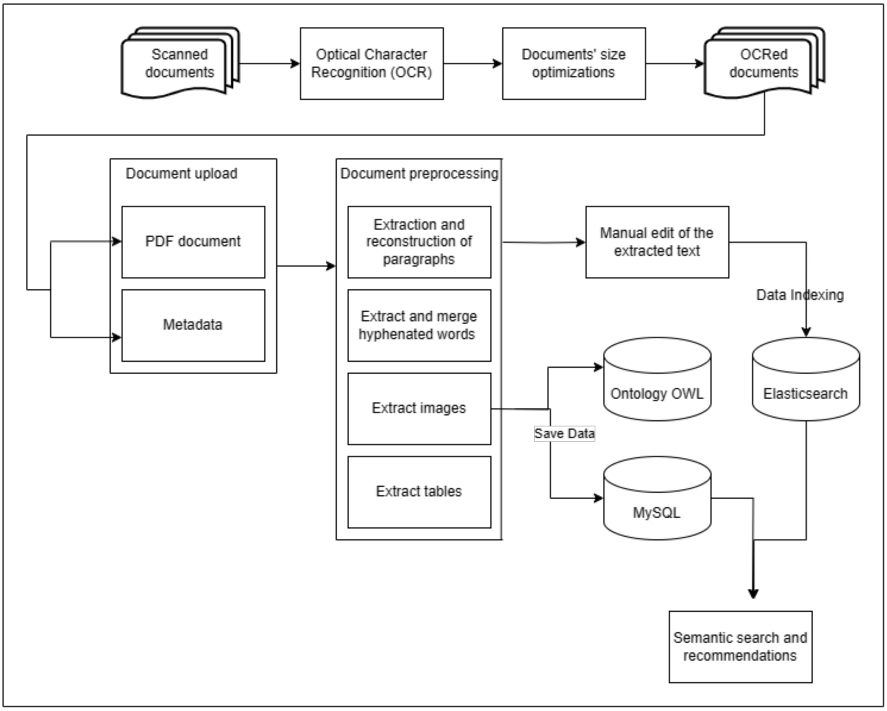
Scanned documents
Optical Character Recognition (OCR)
Documents' size optimizations
OCRed documents
Document upload
PDF document
Metadata
Document preprocessing
Extraction and reconstruction of paragraphs
Extract and merge hyphenated words
Extract images
Extract tables
Manual edit of the extracted text
Data Indexing
Ontology OWL
Elasticsearch
Save Data
MySQL
Semantic search and recommendations

Fig. 2. PDF processing pipeline.

The document preprocessing pipeline contains several steps: text extraction to reconstruct paragraph boundaries, text corrections such as merging hyphenated words, and image and table extraction.

Text extraction involves several heuristics for paragraph extraction and noise removal, which include removing blank lines of text, removing multiple spaces and other delimiting characters, merging hyphenated words, or skipping pages with less than 400 tokens. Paragraph segmentation is performed depending on several factors, namely the position of the text on the current line, the end of sentence punctuation marks, and the capital letter at the beginning of the following line. The extracted paragraphs are saved into Elasticsearch.

Image extraction is based on existing annotation in the PDF file. Images are extracted and stored into binary representation (base64 encoded) in Elasticsearch. For document reconstruction and visualization, the image is converted to JPEG format.

Table extraction is performed using the camelot-py[15] library, using two extraction methods, depending on the text alignment on the page or the lines of text in the document. The first method is preferred to identify tables with no explicit edges and may produce false positives, while the second method detects tables more accurately. The extracted tables are saved in HTML format.

After the text preprocessing, the *domain inference* step is performed if the information was not already extracted from the XML file included. Two methods were employed for this task: Transformer-based and LLM (Large Language Models)-based. The document categorization methods are described in section 3.4.

Rendering a book. Depending on the current processing status of the book, the editing page and user options are different. In the first step, while the book is being processed, the meta information corresponding to the book is displayed in a non-editable format. Once the book has been successfully processed, its contents become visible. The domain and subdomain of the book are automatically inferred based on the text. The librarian can manually edit the content of the document. The interface includes a visual editor created using TinyMCE[16]. The librarian may apply corrections to the automatically processed text in terms of both page layout and layout elements such as font type, text size, or color. This stage ensures correct delimitation between chapters, subchapters, and sections.

Computing the document's vectorial representation. The librarian can edit and save the book's metadata and content. After this step, the document processing step is marked as completed. The system starts a new process in the background that computes the vectorial representation for each book paragraph using a BERT-based model [13] pre-trained for the Romanian language, namely RoBERT-base. The vectors corresponding to a paragraph are computed as the average of the vectors on the penultimate layer for each token. These vectors are used to display semantic recommendations. The book's content can no longer be modified during this processing step. Paragraph-level vector representations are stored in Elasticsearch, where the paragraphs are also stored. The average of these vectors is used as a representation of the entire book and is saved in the relational database to efficiently calculate the semantic similarity between books in the recommendation mechanism.

Filtering and smart search engine. The filtering facility built into the smart search web portal is based on "more like this" queries in Elasticsearch. Filtering is performed according to various filters such as book name, author name, domain, or year of publication. At the same time, for the book recommendations module, the semantic similarities computed based on the vectorial representations of the books are used together with the ratings. The algorithm calculates the semantic distance by the cosine similarity of the two resulting vectors of the current book and all other books; then, these distances are ordered in ascending order. The distances obtained are weighted

[15] https://pypi.org/project/camelot-py/, last accessed 15/10/2023.

[16] https://www.tiny.cloud/docs/, last accessed 15/10/2023.

with the book rating if the number of ratings given to the book is relevant. The rating for a book is considered if the number of votes given to the book by a user is in the top 30% of voted books. Recommendation systems commonly use this concept, called quantile, which consists of dividing the dataset into adjacent subgroups of equal size. The rating of the most voted books is further integrated into the previously described semantic distance, and then the first five books are recommended to the user. In computing the book recommendation, the weight is 70% for the semantic distance between books and 30% for the general rating of the book. Equation 1 is used to compute the semantic recommendations.

$$\boldsymbol{RecommValue\left(book_i, book_j\right) = SimWeight * \left(1 - cos\ cos\ \left(book_{array_i}, book_{array_j}\right)\right) + RatingWeight * \left(MaxRating - WeightedRating\left(book_j\right)\right)} \tag{1}$$

where $book_{array_i}$ and $book_{array_j}$ are the vector representations of books i and j. The weighted rating at the book level is derived from movie recommendation algorithms integrated into various platforms (e.g., the IMDB[17]) platform as shown in Equation (2):

$$\boldsymbol{WeighthedRating\left(book_j\right) = R * \frac{v}{v+m}} \tag{2}$$

where R is the average rating of book j given by users, v is the number of votes given to the book, and m is the minimum number of votes needed to consider that book as recommended to other books (in this case, it is the quantile for the top 30% voted cards from the system). $SimWeight$ is defined as the semantic similarity weight in computing the recommendation; in this case, it is 70%. Similarly, $RatingWeight$ is the rating weight, and the value is 30%.

3.4. Documents Categorization

We integrate a mechanism for document categorization, also referred to as domain inference, where the category is automatically inferred based on the text within the document. The predictions are further utilized to autonomously populate the Lib2Life domain ontology.

3.4.1. Domain Inference

The document domain prediction method is built as a multiclass classification task. Given a document description as input, the model predicts its domain. Domains are structured as classes; a document can only belong to one of the 17 domains, corresponding to the ontology classes. We conducted a study on the capabilities of two neural network architectures, either leveraging a Transformer architecture [14, 15] or an LLM (Large Language Model) architecture [16, 17]. The dataset used for training was built from books that already had the domain and subdomain annotated. To resolve

[17] https://www.imdb.com/, last accessed 15/10/2023.

the class imbalance due to the variable number of samples per domain, the data split for train/test/validation for both models was 60/20/20.

The first model [4] employs a Romanian pretrained Bidirectional Encoder Representation from Transformers (BERT) [18] encoder-only model, namely RoBERT [13]. RoBERT was pretrained on the Romanian Wikipedia dump (50M words, 2M sentences and 0.3 GB), the RoTex corpus (240M words, 14M sentences and 1.5 GB) [19] and the Oscar collection (1.78B words, 87M sentences and 10.8 GB) [20] and followed the original BERT methodology for training. For this experiment, the RoBERT-base model (114M parameters) was leveraged, consisting of 12 Transformer layers, 12 attention heads, a hidden size of 768, and a vocabulary of 38,000 words. The architecture of our model consisted of a BERT encoder from which the "[CLS]" token embedding is passed through a dense layer, followed by a dropout layer. For the softmax activation, categorical cross-entropy was used, which outputs a probability between 0 and 1 for each class domain. The model uses a single categorical feature as the target. The categorical features are hot encoded, creating a one-hot vector from all domains, and each vector could be considered a probability distribution. The model was trained for a batch size of 8, with a sequence length of 512, during 5 epochs and fine-tuned using an Adam optimizer [21] with a learning rate of 2e-5, a weight decay of 0.01, and a dropout rate of 0.1.

The second model, namely mT0 [22], is an LLM capable of cross-lingual generalization, pretrained on multilingual Common Crawl's web crawl corpus, mc4[18] [23], and based on the T5 Transformer architecture [23]. We employed the mT0-xxl (13B parameters) model, which follows the original mT5 [24] architecture, being a multilingual encoder-decoder Transformer trained on a cross-lingual public pool of prompts (xP3)[19] representing a collection of datasets across 46 languages and 16 NLP tasks. The model's architecture consists of 2 x 12 layers, an encoder, and a decoder of BERT-base size and configuration [18]. The method leverages few-shot context learning performed via transfer learning while benefiting from the capabilities of the pre-trained LLM model. The few-shot methodology selects the top 10 labeled samples from the train set as our support set, and it combines them with the test set to create the few-shot learning dataset. The model is then fine-tuned on the few-shot dataset, after which we make predictions on the test set. Similar hyperparameter settings were leveraged to maintain a baseline for comparison: the model was trained for 10 epochs with a batch size of 8 and a sequence length of 512. We used an Adam optimizer [21] with a learning rate of 2e-5 and a weight decay of 0.01.

3.4.2. The Lib2Life Ontology

Ontologies are essential to knowledge representation in the Semantic Web [25] as they represent a set of information in a format accessible to both humans and machines (for instance, digital resources and the relationships between them). The purpose of ontologies is to enable applications to access structured information in the form of a knowledge base (in our case, an ontology implemented using a knowledge graph),

[18] https://huggingface.co/datasets/mc4, last accessed 7/11/2023.
[19] https://huggingface.co/datasets/bigscience/xP3, last accessed 7/11/2023.

which contains entities and related relationships. Storing information in this structured format enables complex search functionality using specific queries.

A dedicated ontology was developed within the *Lib2Life* project[20] based on a collection of documents provided by the "Carol I" Central University Library[21] and the other university libraries in Romania. The ontology contains 17 classes and subclasses related to literature concepts. Entities include books, authors, organizations, and literary movements, which are related using object properties (such as *writtenBy*) and described by multiple data properties (such as *publishingYear*). The ontology incorporates document attributes and specific features that support data mining through advanced queries. These queries can be used to develop advanced search engines and automatically classify new documents introduced into the system.

The development of the *Lib2Life* ontology followed the methodology proposed by Methontology [26]. The aim was to provide a solid representation of the domain and the collection of documents, a representation that meets the existing standards in the field.

Dublin Core (DC)[22] consists of metadata tags created to describe digital resources [27]. The Dublin Core scheme consists of a set of vocabulary items. The specificity of its tags, which aim to incorporate information about a resource and details about its authors, make it a perfect match for describing books and other documents. DC is widely used by librarians, archivists, scientific researchers, and software developers and has since been accepted as the standard for annotating document metadata. The *Lib2Life* ontology is aligned with this standard because it uses appropriate DC tags for some of the classes and properties needed to correctly describe the physical documents contained in central university libraries. In a digital library, metadata must be linked to the digital objects and have great potential for developing complex systems such as search engines based on filter-based search or advanced text-based search mechanisms on semantic similarity techniques [8].

FOAF[23] (Friend Of A Friend) is an ontology model describing people and organizations, their activities, and their relationships with objects. Most properties related to social networks were not included in the *Lib2Life* ontology. However, general classes such as "Person" or "Organization" were used to define authors and publishers.

The *Lib2Life* ontology integrates some of the DC and FOAF classes and properties useful for defining an ontology for a library. A new entity is created if no corresponding class or property is found in the two previous ontologies. The main class of the ontology is "Knowledge Domain," which incorporates domains organized into disjoint subclasses.

Fig. 3, generated with WebVOWL[24], shows representative properties of objects and data from the ontology. Nodes colored light blue are ontology-specific properties, while those colored dark blue are imported from other ontologies. Furthermore, we have included axioms as statements of concepts used in logical inferences. These include disjunction axioms between classes (e.g., complementary moves), *owl:differentFrom*

[20] https://lib2life.bcub.ro/en/, last accessed 15/10/2023.

[21] https://unibuc.ro/despre-ub/resurse-educationale/biblioteci/?lang=en, last accessed 15/10/2023.

[22] https://dublincore.org, last accessed 15/10/2023.

[23] http://xmlns.com/foaf/spec/, last accessed 15/10/2023.

[24] http://vowl.visualdataweb.org/webvowl.html, last accessed 15/10/2023.

axioms between individuals (e.g., linguistic entities), as well as property axioms (e.g., inverse properties like *isAuthorOf* and *writtenBy*).

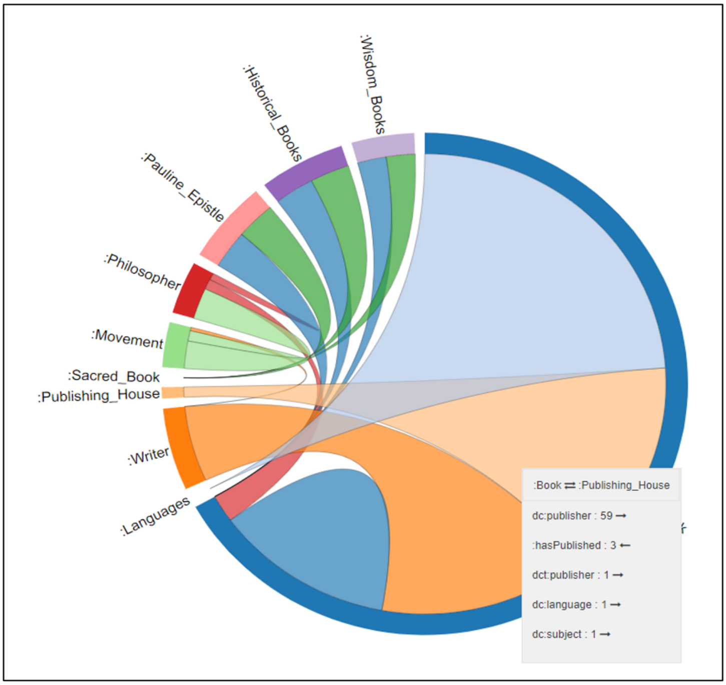

Fig. 3. Ontology object properties.

Additional information regarding objects and data properties is outlined within Table 1.

Table 1. Main object properties and data, along with descriptions and examples.

Type	Property	Description
Object property	hasFirst	Allows mention of the first volume of a series. (6 uses)
	hasNext	Allows mention of the next volume in a series. (52 uses)
	hasPrevious	Allows mentioning the previous volume of a series. (52 uses)
	hasPublished	It is used to mark that an organization has published documents. (8 uses)
	isAuthorOf	It is used to mark an author created a particular document. (90 uses)
	isRepresentedBy	It is used to mark that a literary movement is represented by one person. (10 uses)
	significantFor	It is used to mark that a person is significant to a particular movement. (1 use)

Type	Property	Description
	isTranslatedBy	It is used to mark that a document is translated by a person. (6 uses)
	dc:publisher	It is used to mark the fact that a book is published by an organization. (29 uses)
	writtenBy	It is used to mark the fact that a book is written by an author. (160 uses)
Data property	birthDate	It is used to mark that a person has a date of birth. (24 uses)
	deathDate	It is used to mark that a person has a date of death. (22 uses)
	pageNumber	It is used to mark how many pages a document has. (8 uses)
	publicationYear	It is used to mark the fact that a document is one year old. (150 uses)
	tomeNumber	It is used to mark that a document has a volume number. (8 uses)

By correlating existing data in an ontology, an API can extract connected information that might not be immediately visible to humans using an ontology query language like SPARQL [28].

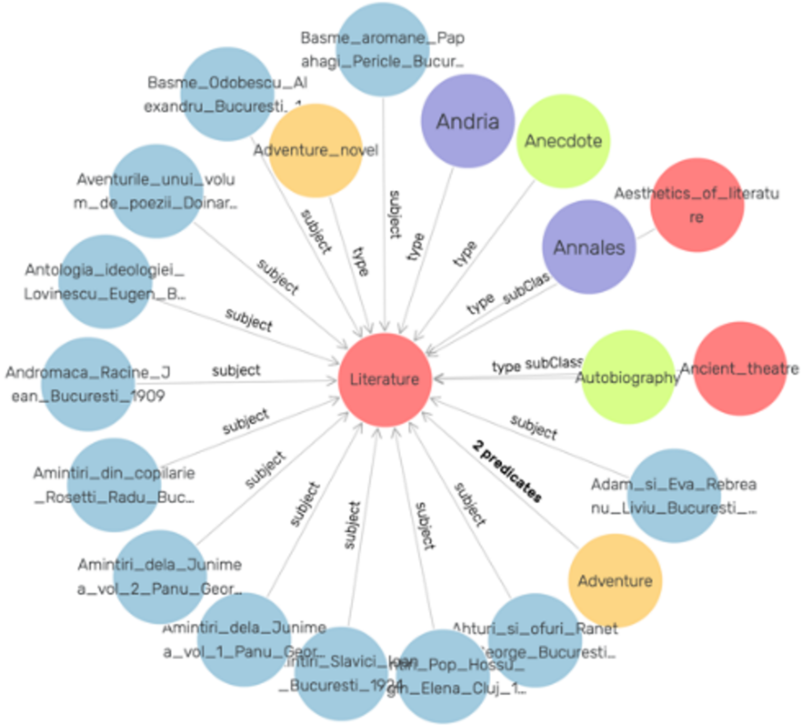
Basme_aromane_Pap ahagi_Pericle_Bucur..
Basme_Odobescu_Al exandru_Bucuresti_1..
Adventure_novel
Andria
Anecdote
Aventurile_unui_volu m_de_poezii_Doinar..
Aesthetics_of_literatu re
Annales
Antologia_ideologiei_ Lovinescu_Eugen_B..
Andromaca_Racine_J ean_Bucuresti_1909
Literature
Autobiography
Ancient_theatre
Amintiri_din_copilarie _Rosetti_Radu_Buc..
Adam_si_Eva_Rebrea nu_Liviu_Bucuresti_..
Amintiri_dela_Junime a_vol_2_Panu_Geor..
Adventure
Amintiri_dela_Junime a_vol_1_Panu_Geor..
..intiri_Slavici_Ioan _Bucuresti_1924
..an_Pop_Hossu_ ..gh_Elena_Cluj_1..
..hturi_si_ofuri_Ranet ..eorge_Bucuresti..
subject
type
subClass
2 predicates

Fig. 4. Example of individuals from the Literature domain within the ontology.

Fig. 4 shows the knowledge graph illustrating individuals within the Literature domain, arranged in lexicographic order. The central node in the graph is the overarching category label, which is linked to an array of nodes. Each of these nodes signifies an individual within the ontology, a subtype, or a subclass, collectively creating a structured network of literature-related entities.

Additionally, we leverage a system that automatically populates the ontology with individuals based on the domain predictions described in section 3.4.1. This approach is useful when metadata is not available for the newly uploaded document.

3.5 Web Services

Web services are an essential layer for operating the *Lib2life* platform. They include document processing and recommendation services based on paragraph-level semantic similarities. For document processing, the basic function is document upload (performed by a user with the librarian role), followed by the automatic classification in one of the existing *Lib2Life* ontology domains, a step integrated into the loading process. As for semantic services, they include searches and filters available in the web interface. Thus, the user can find documents whose paragraphs contain certain keywords or perform searches based on different filters: author, publisher, document publication year, and so on. Semantic recommendations are obtained based on the pre-trained semantic algorithms described previously. Semantic services are based on the text paragraphs extracted from the document and preprocessed, then saved into an Elasticsearch database.

The *Lib2Life* web interface communicates with an application server developed using the Flask[25] library in Python. Text extraction was performed via pdfminer[26], while table extraction leveraged camelot-py library. The TensorFlow[27] library was used to implement and train the text processing models, and the Transformers library provided by Hugging Face[28] was used to load the pre-trained Transformers models. The server interacts with the Elasticsearch and MySQL databases.

Access to resources within the application is based on a JSON Web Token (JWT), which is required by the application server. The token is obtained automatically by the server upon authentication request in the application. The token is valid for 30 minutes, after which it must be regenerated. The JWT token must be attached to each request, except for the authentication request and the one for registration in the platform, as an authorization header within the HTTP request. The authorization type used is Bearer Token[29]. An overview of the web services leveraged in the *Lib2Life* platform is illustrated in Table 2.

[25] https://github.com/pallets/flask, last accessed 15/10/2023.

[26] https://pdfminersix.readthedocs.io/en/latest/, last accessed 15/10/2023.

[27] https://www.tensorflow.org/resources/libraries-extensions, last accessed 15/10/2023.

[28] https://huggingface.co/, last accessed 15/10/2023.

[29] A Bearer Token is a type of token used in web applications and APIs to hold user credentials and indicate authorization for requests and access.

Table 2. *Lib2Life* web services overview.

Web service category	Description	Examples
Authentication-related	These services are valid for creating new accounts and for authentication-related processes, for instance, password recovery or account existence verification.	User authentication, account registration, verification of account existence, password change, edit user profile, JWT authentication, or forgotten password.
Document access related	These services are available for any type of user. Readers may receive results regarding the documents in the system, see recommendations for similar documents, and visualize information for specific documents. Additionally, the document rating functionality (Likert scale from 1 to 5) and displaying a document in its original format are included in this section.	Documents retrieval by ID, by filters, by domains and subdomains, by format (PDF or HTML); Retrieve similar documents or save document rating.
Document management related	They are available to users with the librarian role in charge of registering and editing documents in the system. Services provided include uploading documents, editing, and saving content.	Document upload, deletion, data editing, or computing the vectorial representation.
Statistics-related	These services are called from the statistics page available within the web application.	Display document distributions by domain, language, year, domain, author, or publishing house.
Administrator-dedicated	Administrators manage user accounts. They can edit user roles and delete their accounts.	Retrieve users list, edit user roles, delete user accounts, or search users by specific filters.

3.6 The Lib2Life Web Portal

The web portal presents the previous web services to users. Users can access the system with an accessible and easy-to-use web interface. The web portal provides specific functions for the three types of users: reader, librarian, and administrator. The web interface was built using the Angular[30] version 9 framework. The user interface is connected to two application servers, both connecting to an Elasticsearch instance for storing and retrieving indexed documents.

Depending on the assigned role, a user may or may not have access to a specific component available in the web interface.

[30] https://angular.io, last accessed 15/10/2023.

3.6.1. Regular User Functionalities

Users with reader roles may access the system to retrieve documents or search for specific content, based on the filtering and smart search engine capabilities, previously described in section 3.3.

Document display and visualization. On the home page, the user can see a list of documents with search and filter capabilities. Documents are displayed as a list, each item containing document details and a series of actions that can be performed depending on the authenticated user's role, namely view similar documents, display or edit the metadata content, display the original document in PDF format, or delete the document. A reader can only view information and have no possibility to modify documents. Document preview is represented by the cover or the document's first page. Each document has the associated users' ratings, the number of views, and the metadata such as publisher, place of publication, or year of publication.

Viewing similar documents. For each current document, the top five similar documents are computed and displayed. The semantic similarity is computed considering the vectorial representation of each document and its ratings.

Viewing the original document. This option allows the user to visualize the original document in PDF format, facilitating reading and navigation, as well as bookmarking specific pages.

General statistics. All user roles can access the general statistics page that provides document insights, such as global domain distribution by publication years, top authors by number of publications, distribution of books by language, and top publishing houses by number of publications.

Ontology visualization. The *Lib2Life* ontology can be accessed through a built-in preview tool, which allows the user to navigate through the existing classes and subclasses and study the relationships between class entities.

3.6.2. Librarian and Administrator Functionalities

Users with the librarian role are responsible for uploading documents, checking the automatically extracted text, and correcting it, if necessary, while users with the administrator role manage user accounts and modify roles.

The *document upload and processing* functionality is available only to users with librarian and administrator roles and involves uploading documents in PDF format.

Deleting a document. The user with the librarian role can delete a document from the system. To retrieve the document, the librarian can search for it on the home page either through the search functions or by navigating between pages.

Editing metadata. From the actions menu available, the user with a librarian role can edit the metadata added to a previously created document. Editing the content of a book can be done from the "Edit content" tab. TinyMCE is used to display the book's content in an intuitive and editable format.

4 Results

Table 3 describes the experimental results comparing two domain inference methods using a simple Transformer encoder-only architecture (RoBERT-base, 114M parameters) and an LLM encoder-decoder architecture (mT0-xxl, 13B parameters). The evaluation metric used is the F1-score, and the results are further contextualized with the corresponding number of test samples for each domain.

Table 3. Domain classification results (bold marks the best results).

Domain	F1-score		Test Samples
	Transformer (RoBERT)	LLM (MT0)	
Applied Sciences	0.01	**0.75**	5
Arts	0.50	**0.82**	10
Economic Sciences	**0.80**	0.70	11
Ethnography & Folklore	0.33	**0.70**	10
Exact Sciences	0.80	**0.95**	11
Generalities	0.01	**0.60**	20
History	**0.88**	0.80	70
Juridical Sciences	0.73	**0.87**	24
Linguistics & Philology	**1.00**	0.86	16
Literature	**0.91**	0.88	98
Natural Sciences	0.59	**0.87**	19
Pedagogy	0.67	**0.93**	22
Philosophy & Psychology	0.01	**0.80**	9
Politics	0.44	**0.57**	19
Public Policy	0.01	**0.67**	14
Social Sciences	0.01	**0.67**	13
Theology	0.71	**0.88**	20
Weighted average	*0.73*	***0.81***	

A clear trend emerges while analyzing the performance across domains, indicating a positive correlation between the number of samples in the test set and the achieved F1-score. Generally, domains with more samples exhibit superior performance, suggesting that the models benefit from a more extensive and diverse dataset for learning. However, despite the overall trend, the RoBERT model faces challenges in distinguishing documents from domains with low sample counts and class imbalance. Specifically, 5 out of 17 domains exhibit particularly low F1-scores, with values close to zero, namely the domains of *applied sciences*, *generalities*, *philosophy and psychology*, *social sciences,* and *public policy.*

Upon further investigation, we discovered that beyond the issue with low sample counts, the model's difficulty distinguishing certain domains can be attributed to similar vocabulary across multiple domains. For example, terms like “politic” (eng., “political”) or “juridic” (eng., “juridical”) are prevalent in both *juridical sciences* and *public policy* domains, posing challenges for the model in accurately categorizing documents from these domains.

The results highlight the benefits of a larger model in handling domain specificity and discerning intricate relationships even when dealing with limited samples and overlapping vocabulary terms. However, despite being a smaller model, RoBERT performed better than mT0 on 4 domains out of 17, emphasizing that even a smaller model with an appropriate architecture can still perform remarkably well in specific domains, showcasing the relationship between model specifications and task-specific requirements.

With some domains having significantly better F1-score, mT0 outperformed RoBERT with an average of 8%, reaching an overall weighted average F1-score of 0.81 compared to RoBERT, which only scored an overall 0.73 F1-score. RoBERT model struggles significantly in the five problematic domains, especially when faced with low sample counts and overlapping vocabulary. At the same time, mT0 demonstrates a notable improvement, showcasing the advantages of a larger model, particularly in capturing domain-specific patterns and mitigating the impact of limited samples and shared vocabulary.

The challenges faced by RoBERT in certain domains could be attributed to the limited capacity of a smaller model to grasp the confusion of vocabulary overlap and nuanced domain characteristics. In contrast, mT0’s superior performance suggests that a more extensive parameter space enhances the model’s capability to discern complex patterns.

In terms of accessibility, the initial version of our platform underwent evaluation through a survey (presented in [2]), distributed to 23 users aged 20 to 50, encompassing various educational backgrounds and professional domains. The majority of respondents were aged 20-30, with a higher representation of women. Feedback indicated satisfaction with the platform’s usability and design, although users expressed a need for enhancements such as improved document visualization and additional filtering options. System limitations were encountered, particularly with text extraction algorithms, necessitating iterative improvements and consideration for domain-specific characteristics in future developments. The insights gathered from this evaluation have played an important role in shaping the current version of the *Lib2Life* platform. More filtering options for browsing the collection of books were added, ensuring that user feedback is integrated to enhance usability. Additionally, we developed enhanced processing algorithms to address limitations and refine features for an improved user experience.

5. Conclusions

The *Lib2Life* platform represents a transformative paradigm in digitizing cultural heritage and promoting smart education in Romania. Its main objective is to promote the principles of open access, equitable knowledge distribution, and cultural preservation. This platform marks the collaborative spirit that thrives within Romania’s academic and cultural communities, as it brings libraries, scholars, and technology to redefine the boundaries of education and heritage.

The platform stores digitized versions of historical documents owned by the Central University Libraries in Romania, offering open-source access to them in an online environment. The platform includes useful filters for effective document search and retrieval, as well as the ability to read a document in PDF format. Users can additionally explore the entire domain ontology within the web application. Semantic filtering and recommendation functionalities allow users to find the most relevant documents for a query. The system architecture contains modular components, which can be separated and executed on different servers depending on the requirements. The modular architecture minimizes connections between components, and data processing can be segmented and executed separately at the component level. Moreover, the platform incorporates an automatic document domain inference based on encoder-only and encoder-decoder architectures supported by LLM capabilities that reached a weighted average F1-score of 81%.

The *Lib2Life* ontology includes specific characteristics of knowledge representation systems, such as promoting consistency in structuring and organizing information along with corresponding axioms for extended reasoning, extensibility that allows easy further development, and reusability for future implementations.

Moreover, *Lib2Life* enables librarians to build a database for their document collection, which can be further leveraged for advanced studies such as performing analyses focused on the evolution of the literature, correlation to historical events, or changes in writing styles over time.

Currently, the combined archives of the four university libraries hold roughly 2.5M physical documents awaiting digital conversion. The *Lib2Life* project successfully processed 4M scanned pages, introduced algorithms to facilitate document retrieval, indexing, and recommendation processes, created an annotation system tailored for the previous titles, and established a comprehensive database of standard heritage documents. Apart from addressing the varied requirements of users at the four central university libraries, which encompass students, faculty members, and researchers, the project aims to extend its reach to various other public sectors. This includes but is not limited to the national library network, the Romanian diaspora, as well as local and international experts specializing in the preservation and restoration of cultural heritage. Additionally, we aim to support public institutions and decision-making bodies across domains such as education, culture, and the broader informational society.

As we conclude this exploration, *Lib2Life* contributes to opening the path toward a smarter, more cultured, and sustainable future where knowledge is open and accessible to everyone. Directions for future work include exploring ways of making the *Lib2Life* system portable to other libraries or library groups. This includes assessing its scalability, adaptability, and interoperability with diverse library systems. By enhancing its portability, we aim to broaden access to cultural heritage resources across various library settings.

CRediT author statement. Melania Nitu: Conceptualization, Methodology, Software, Writing – original draft. **Mihai Dascalu**: Conceptualization, Methodology, Validation, Writing – review and editing, Supervision, Project administration, Funding acquisition. **Maria Dorinela Dascalu**: Conceptualization, Methodology, Software, Data curation, Writing – review and editing. **Laurentiu-Marian Neagu**: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – review and editing. **Maria-Iuliana Dascalu**: Validation.

Acknowledgements. This work was supported by a grant from the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0689 / „Lib2Life – Revitalizarea bibliotecilor și a patrimoniului cultural prin tehnologii avansate" within PNCDI III.

References

1. Dascalu, M., Sandric, B., Neagu, L.-M., Toma, I., Hanganu, L., Chisu, L., Trausan-Matu, S., Simion, E., Tomescu, S., Mitocaru, I., Gutu-Robu, G., Nitu, M., Cristea, A., Dinu, A., Dinu, L.P., Georgescu, S., Uban, A., Antal, E., Bota, C., Ciongradi, E., D'Annibale, E., Demetrescu, C., Dima, B., Fanini, D., Ferdani, Streinu, M., Borlean, O., Buruiana, M., Iancu, L.M., Andrei, V., Miu, C., Tufaru, M., Ghemut, F., Matei, D., Chelaru, R.-D., Paiusan-Nuica, C.: Heritage in the digital era. Cases and best practices from Romania. Pro Universitaria, Bucharest (2021)
2. Mitocaru, I., Gutu-Robu, G., Nitu, M., Dascalu, M., Trausan-Matu, S., Tomescu, S. and Florescu, G.: The Lib2Life Platform - Processing, Indexing and Semantic Search for Old Romanian Documents. Int. Conference on Human Computer Interaction (RoCHI) 11-18 (2020)
3. Nitu, M., Dascalu, M., Dascalu, M.-I., Cotet, T.-M., Tomescu, S.: Reconstructing Scanned Documents for Full-Text Indexing to Empower Digital Library Services. Emerging Technologies for Education: 4th International Symposium, SETE 2019, Held in Conjunction with ICWL 2019, pp. 183-190. Springer-Verlag, Magdeburg, Germany (2019)
4. Nitu, M., Dascalu, M., Gutu-Robu, G., Dascalu, M.-I., Tomescu, S.: Lib2Life - Domain Categorization of Books using BERT Language Models and Knowledge Graph Population. Romanian Conference on Human-Computer Interaction (2021)
5. Nitu, M., Ruseti, S., Dascalu, M., Tomescu, S.: Semantic Recommendations of Books Using Recurrent Neural Networks. 235-243 (2021)
6. Gutu-Robu, G., Ruseti, S.,Tomescu, S., Dascalu, M., Trausan-Matu, S.: Designing an Ontology for Knowledge-based Processing in Romanian University Libraries. 8th Int. Workshop on Semantic and Collaborative Technologies for the Web, in conjunction with the 16th Int. Conf. on eLearning and Software for Education (eLSE) 1, 119-126 (2020)
7. Tomescu, S., Mitocaru, I., Gutu-Robu, G., Nitu, M., Dascalu, M., Trausan-Matu, S.: Advanced Natural Language Processing Techniques for Restoring Old Romanian Documents. In: Dascalu, M., Sandric, B. (eds.) Heritage in the digital era. Cases and best practices from Romania., pp. 25-41. Pro Universitaria, Bucharest, Romania (2021)
8. Lebert, M.: Project Gutenberg (1971-2008), University of Toronto (2010)
9. Streitfeld, D.: The Dream Was Universal Access to Knowledge. The Result Was a Fiasco. The New York Times, (2023)

10. Alewaeters, G.: VUBIS: A User-Friendly Online System. Information Technology and Libraries 1, 206-221 (1982)
11. Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., Walker, J.H.: DSpace: An Open Source Dynamic Digital Repository. D-Lib Magazine 9, (2003)
12. Rana, C., Jain, S.K.: Building a book recommender system using time based content filtering. WSEAS Transactions on Computers 11, 27-33 (2012)
13. Masala, M., Ruseti, S. and Dascalu, M.: RoBERT-A Romanian BERT Model. COLING 6626-6637 (2020)
14. Lin, T., Wang, Y., Liu, X., Qiu, X.: A Survey of Transformers. AI Open 3, 111-132 (2021)
15. Turner, R.E.: An Introduction to Transformers. ArXiv abs/2304.10557, (2023)
16. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J., Wen, J.-r.: A Survey of Large Language Models. ArXiv abs/2303.18223, (2023)
17. Naveed, H., Khan, A.U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., Mian, A.S.: A Comprehensive Overview of Large Language Models. ArXiv abs/2307.06435, (2023)
18. Delvin, J., Chang, M.-W,, Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1, 4171–4186 (2019)
19. Aleris: RoTex Corpus Builder. https://github.com/aleris/ReadME-RoTex-Corpus-Builder, last accessed 07/17/2023
20. Javier Ortiz Suarez, P., Sagot, B., Romary, L. and Sagot, B.B.: Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures. 7th Workshop on the Challenges in the Management of Large Corpora (CMLC-7) (2019)
21. Loshchilov, I., and Hutter, F.: Decoupled Weight Decay Regularization. ICLR (2017)
22. Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S.R., Scao, T.L., Bari, M., Shen, S., Yong, Z., Schoelkopf, H., Tang, X., Radev, D.R., Aji, A., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., Raffel, C.: Crosslingual Generalization through Multitask Finetuning. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics 1, 15991-16111 (2023)
23. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research 21, 1-67 (2020)
24. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 483-498 (2021)
25. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities. Scientific American 284, 1-5 (2001)
26. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. (1997)
27. Weibel, S., Kunze, J., Lagoze, C., and Wolf, M.: Dublin Core Metadata for Resource Discovery. Internet Engineering Task Force RFC (1998)
28. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and Complexity of SPARQL. 30-43 (2006)