Building Roger: Technical Challenges While Developing a Bilingual Corpus Management and Query Platform

Cosmin Strilețchi[1][a], Mădălina Chitez[2][b] and Karla Csürös[2][c]

[1] *Communications Department, Technical University of Cluj-Napoca, Cluj-Napoca, Romania*

[2] *Department of Modern Languages and Literatures, West University of Timisoara, Timisoara, Romania*

Keywords: Corpus Query Platform, Academic Writing, Bilingual Corpus, Big Data, ROGER.

Abstract: This paper presents an approach to a bilingual Corpus query system. ROGER has been designed and implemented as a cross-platform distributed web application. The backend interface available to authenticated administrators provides the digital tools for managing the database stored texts and associated metadata, and also offers an extensive statistics mechanism that cover the data composition and usage (words, characters, languages, study levels, genres, domains and n-grams). The frontend capabilities are offered to the registered users allowing them to search for specific keywords and to refine the obtained results by applying a series of filters. Current platform features include search terms and phrases, n-gram distributions and statistical visualizations for performed queries. After inputting a search term / phase, the user may filter available texts by: (i) language (English, Romanian); (ii) student genre (currently 20 genres); (iii) study year (1 through 4); (iv) level (BA, MA or PhD); (v) discipline (currently 8 disciplines) and (vi) gender (male, female or unknown). A series of solutions have been implemented to improve the response times of the intensely computational procedures that manipulate big amounts of data.

1 INTRODUCTION

This paper presents a corpus search engine for the bilingual comparable corpus ROGER (Corpus of **RO**manian Academic **GE**n**R**es), compiled by the research group at CODHUS (Centre for Corpus Related Digital Humanities) (Chitez et al, 2020) from the West University of Timisoara. ROGER consists of novice academic writing genres, in Romanian (L1) and English (L2), collected from Romanian universities, with the purpose of investigating student writing practices in both their mother tongue and English as a Foreign Language.

The corpus support platform addresses learner corpus data in a multidimensional contrastive framework: genres written by students in L1 versus L2, genres written in different disciplines as well as genres written at different study levels (Bachelor's, Master's and Doctoral study programmes).

The present paper aims to introduce the ROGER corpus platform to academic researchers and experts in fields such as Corpus Linguistics (McEnery, 2019), Academic Writing, Contrastive Analysis (Johnsson, 2003), Language for Specific Purpose studies (Ackerley, 2021) and Computer-Assisted Language Learning (Chitez & Bercuci, 2019). What distinguishes the ROGER platform from other corpus support platforms are several salient characteristics (see also Section 2.1): (a) it is the first bilingual novice / learner academic writing corpus with a dedicated open source corpus query platform; (b) it is the first corpus offering information about novice / learner academic writing in the Romanian context; (c) it is the first corpus support platform offering discipline-specific information about novice / learner academic writing ; (d) it is the first open source corpus support platform that can be used for a multitude of academic writing applications and research studies: Romanian-English genre specific contrastive studies, inter-disciplinary linguistic contrastive studies, studies in English L2 disciplinary and general writing.

[a] https://orcid.org/0000-0002-5663-9159
[b] https://orcid.org/0000-0001-9005-3429
[c] https://orcid.org/0000-0002-9556-3724

Strilețchi, C., Chitez, M. and Csürös, K.
Building Roger: Technical Challenges While Developing a Bilingual Corpus Management and Query Platform.
DOI: 10.5220/0011144500003266
In *Proceedings of the 17th International Conference on Software Technologies (ICSOFT 2022)*, pages 390-398
ISBN: 978-989-758-588-3; ISSN: 2184-2833

The paper is structured as follows: In Section 2, we describe the background of the ROGER project, as well as various aspects of the compilation and annotation processes. In Section 3, we focus on our platform's implementation details mentioning the technical challenges and provided solutions. Section 4 covers the facilities offered to the ROGER administrators and regular users. The conclusions are emphasized in the final section.

2 A CORPUS SYSTEM OVERVIEW

2.1 Background and Motivation

Numerous studies on student writing have used cross-language comparisons (Donahue, 2002, 2009; Foster, 2006; Siepmann, 2006; Foster & Russell, 2002; Kaiser, 2003; Trumpp, 1998) to detect rhetorical patterns or linguistic interference. However, such studies were not based on extensive linguistic datasets, which can be employed to extract statistical results. Beginning in about 2000, a new generation of research studies emerged, using text corpora to study texts more systematically and more efficiently. In the case of academic writing studies, collecting corpora is rather challenging, considering that authentic materials (e.g. student papers) are not quite accessible, teachers / tutors might not be willing to collect student papers or text processing in corpus format is time-consuming.

Until now, several well-known research projects have resulted in reference corpora for academic-writing research: (1) The ICLE range of corpora is, certainly, the pioneering and most extensive learner corpus project, serving as a valuable resource for studies of Contrastive Interlanguage Analysis (Granger, 1996); (2) the British Academic Written English Corpus (BAWE): a large genre mapping study by Hilary Nesi (2008; Alsop & Nesi, 2009) in the UK led to a corpus of student texts in which the authors identified about 100 different genres which they group in 13 genre families (see Heuboeck, Holmes & Nesi, 2009, pp. 46-50); (3) Michigan Corpus of Upper-Level Student Papers (MICUSP), in USA (O'Donnell & Römer, 2012; Römer & O'Donnell, 2011): the corpus includes A-graded upper-level papers in 16 disciplines at 4 levels of 7 paper types with 8 textual features; (4) Corpus of Academic Learner English (CALE), in Germany, aiming at developing a corpus-driven, text-centred method based on linguistic criteria for the assessment of writing proficiency in the academic register (Callies & Zaytseva, 2013a, 2013b); (5) the Varieties of English for Specific Purposes Database (VESPA), compiled at the Université Catholique de Louvain, in Belgium, with the purpose of contrasting academic-writing ESP features from various mother tongue backgrounds (Paquot, Hasselgård & Ebeling, 2013); (6) The Romanian Corpus of Learner English (RoCLE) compiled by Chitez (2014): it contains academic writing papers (essays and literary texts) written by Romanian students; (5) Corpus & Repository of Writing (CROW), in USA, which is a learner corpus planned for research, teaching, mentoring and collaboration (Kwon et al., 2018). From all these corpora, only few have also developed their own corpus query platforms: MICUSP and CROW are online free-access platforms, while ICLE is a licence-based product. An alternative model is offered by the BAWE corpus, which can be accessed via SketchEngine (Kilgarriff et al. 2014), which has been open-access until April 2022.

In this context, the ROGER corpus (Chitez et al., 2021) and corpus support platform is unique in the following components: online free-access corpus query platform, bilingual corpus type, user-friendly interface with statistical and feature extraction options (e.g. n-grams).

2.2 Corpus Compilation

A variety of considerations were taken when designing the ROGER corpus. The main target of the ROGER project is the study of language use in contemporary native Romanian (L1) and learner English (L2) academic writing tasks by students attending nine Romanian universities. The corpus was collected over a four-year period (2018-2021) with the help of 27 collaborators who were part of the *AWICNET* (Academic Writing Collection Network) subproject (AWICNET, 2019). AWICNET members identified student contributors, explained the purpose of the corpus collection process, obtained student consent and collected the student texts, which were both uploaded to the ROGER private cloud drive.

Student informants filled in a form that gathered various metadata: demographic information, educational background and writing practices, as well as a GDPR section in which they consented to having their data collected. Between 2018 and early 2020, student gave their handwritten consent via printed forms that were stored securely; since the beginning of the COVID-19 pandemic, in observance with all health protocols, all such forms were submitted by the students in digital format via a *Google Form* which we provided through our collaborators. The previously mentioned metadata was stored separately in *.*xls* format files. Since the metadata was collected entirely in Romanian and consisted of student fill-in-the-blank input, it was later uniformized and translated into English. After the metadata was processed, separate *.*xls* files were created to remove any identifying or unessential information for the ROGER application.

2.3 Corpus Annotation

Students submitted their texts in various digital formats (primarily *.*docx* and *.*pdf*) or in paper format (handwritten or printed). The digital variants were converted to *.*txt* files via file conversion or OCR systems, checked for transformation accuracy and processed. As for the paper format texts, they were scanned and transcribed by our team faithfully, keeping all errors (e.g., grammar, spelling) made by learners and preserving all diacritical marks in Romanian texts.

The processed text files contain basic markup that aims to anonymize the files and remove any unnecessary information. Anonymization, or de-identification, is achieved by replacing any personal or identifiable information about the author of the paper, the collaborator, or the university (e.g., <CONFIDENTIAL_NAME> replaces the names of the student or the collaborator). We have decided to replace all confidential data instead of simply deleting it in order to preserve the layout of the paper. Unnecessary information, which refers to any parts of the paper that could interfere with the results of statistical linguistic analyses (such as references, tables, mathematical calculations, graphs, etc.), were also marked up (e.g., <REF> replaces any in-text citations, footnotes, and bibliographies).

Since the corpus comprises a very large amount of data and metadata, building and processing the corpus involved significant time and human resources from our team of researchers, research assistants, interns and volunteers working part-time or full-time on the ROGER corpus. Due to this, we created a digital guide for processing ROGER texts to guarantee uniformity across the corpus.

3 IMPLEMENTATION DETAILS

The ROGER Corpus platform was designed and implemented as a web distributed software application, so that it will have no special requirements (libraries, plugins, etc.) from the devices (computers, laptops, mobile phones, tablets) used for accessing it. The system was implemented on a LAMP (Linux, Apache, MySQL, PHP / Perl / Python) software stack.

3.1 Corpus Data Structure and Flow

At the core of the entire Corpus platform resides the central database. The Input/Output operations that store and retrieve data are crystallized in two web accessible interfaces. The first one contains all the facilities meant for the Corpus administrators while the other one offers the tools that can be accessed by the application's visitors.

The ROGER Corpus system's database is supposed to store entire texts written by various authors. As simple as it may sound, this involves many aspects that must be taken care of.

The data associated with any Corpus application is defined by its volume. The amount of stored data is expected to be very large, and the immediate consequence refers to increased access times. A regular database structure can manifest unreasonably large time intervals for writing and reading operations. As opposed to this, a carefully planned and structured database can significantly reduce the input and output time intervals making the entire system responsive.

3.1.1 Corpus Texts

The Corpus database is constructed around the corpus texts. This content is stored in files and arrives in the platform by being uploaded via the administration interface.

A file contains a single text written in English or Romanian containing plain text and is composed of ANSI or UTF-8 characters.

Each file contains a useful payload (the Corpus text itself) accompanied by secondary data delimited by special markups.

A typical Corpus file looks like this.

```
<INTERNAL_IDENTIFIER>
<TITLE>Paper title</TITLE>
<MARKUP_1> markup1 content
<MARKUP_2> markup2 content
Actual  text  content  with  or  without
<MARKUP_N> additional markups.
```

The only markup that is of interest while storing the files' content in the database is represented by the <INTERNAL_INDENTIFIER>. Therefore, a corpus file must pass through the following steps in order for its content to be stored in the database.

- **upload**: the text file arrives on the server

- **normalization**: the file's special characters (escape sequences like \n, \r, \t, \0) along with inappropriate spacing (newlines and multiple space characters) are filtered
- **processing**: the internal identifier gets isolated, validated, and extracted, the rest of the markups and associated data are eliminated
- **storing**: the filtered content is deposited in the database using the detected internal identifier as primary key

Since our platform stores thousands of texts, uploading them one by one would be a very time-consuming process. Instead, bulk files upload is allowed and the platform processes 30 files upload at a time. An average Corpus file's size is 100 Kbytes so a bulk upload would consist in processing approximately 3 Mbytes of data which is totally reasonable.

Passing the uploaded files through all the steps mentioned above requires between 2 and 40 seconds, depending on the total size of processed data. To keep the server connection alive and to avoid keeping the administrator waiting in front of an unresponsive page (the regular timeout for server response in 30 seconds) our application makes repeated AJAX calls for each of the uploaded files.

After fully interpreting a file, a message addressing the parsing success is displayed to the administrator. The process continues until all the uploaded files have been processed.

We did not increase the number of the uploaded files even though their volume would allow this due to the computational intensity of the operations mentioned above.

3.1.2 Corpus Metadata

Each corpus text has a series of satellite metadata information that categorizes it. This additional data refers to the content's

- genre
- discipline
- author's gender
- author's study level
- author's study year

A traditional Content Management approach would allow the administrator to fully configure each Corpus text by introducing manually all the required data. Such a process would be very slow and highly ineffective in this platform's situation due to the large number of texts. The database is meant to store thousands of records and uploading and configuring manually the corresponding metadata would be a very time-consuming activity (dozens of man-hours).

A new solution had to be invented to provide an alternative and much more efficient route for storing the texts along with the associated metadata. We came up with creating a metadata **.xls* file that contains each text's satellite information. The correspondence between this file's rows and the corresponding text is made by an internal id matching. Uploading and parsing the metadata file would result in decorating the associated Corpus texts with the corresponding categorizing information.

3.1.3 Corpus n-grams

A very important aspect related to any Corpus database refers to the n-grams extracted from the stored texts.

An n-gram's relevance is given by its appearance number in a certain context. The Roger Corpus context is defined by:

- the language in which the n-gram was found
- the corresponding metadata classifiers (genre, discipline, author's gender, study level and study year)

For computing the 2, 3, 4 and 5 n-grams, all the adjacent groups of words are counted. An n-gram is considered valid only if it occurs more than once.

Our database contains the following average n-grams counters (Table 1).

Table 1: Roger Corpus n-grams counters.

n-gram	Avg. occurrences/ text	Avg. occurrences/ 500 texts
2-grams	168.3	81,941
3-grams	129.3	64,650
4-grams	73.61	36,806
5-grams	49.5	24,798

Considering these numbers and keeping in mind that the entire database contains thousands of texts, the n-grams counters must be computed in a non-real-time process. If stored accordingly, counting, and displaying on-demand categorized n-grams becomes a feasible task.

3.1.4 Corpus Statistics

A set of predefined statistics is defined by the Roger Corpus requirements. They refer to

- texts counters for each discipline and genre, both for Romanian and English writings
- total texts counters per language
- total words number per each language

- total characters number per each language
- total number of 2, 3, 4 and 5 n-grams

The numbers obtained for our current database are reflected in Table 2. The n-grams are presented in Table 1. and are excepted from this report.

Table 2: Roger Corpus statistics counters.

Item	Avg. counters/ 500 texts
Romanian words	369,197.27
English words	432,025.79
Romanian characters	2,201,560.80
5-grams	2,578,191.17

Considering the values in the table above and the fact that the counting SQL functions work slower as the data increases in volume, computing the counters in a non-real-time process is also recommended.

3.1.5 Database Storage

The database entities that define the **metadata characterizers** are the following ones:

- corpus_genres (corpus_genre_id, name)
- corpus_disciplines (corpus_discipline_id, name)
- corpus_genders (corpus_gender_id, name)
- corpus_study_levels (corpus_study_level_id, name)

The database entity that stores the **metadata** is defined by

- corpus_metadata_id (an autoincremented value)
- internal_text_id (an external reference to the corpus texts storing entity)
- corpus_genre_id (an external reference to the corpus_genres entity)
- corpus_discipline_id (an external reference to the corpus_study_levels entity)
- corpus_gender_id (an external reference to the corpus_gender entity)
- corpus_study_level_id (an external reference to the corpus_study_level entity)

Storing only the references to the actual entities that store categorizing information was preferred as opposed to directly memorizing the corresponding name for two main reasons. Firstly, the text is detached from its numerical abstraction making it suitable for a possible future multilanguage translation. Secondly an SQL syntax that retrieves data from the metadata entity will work faster if the filtering is done on numeric values than on textual data.

The database entity used for storing the **corpus texts** is composed of the following fields:

- corpus_text_id (an autoincremented value)
- filtered_content
- original_content (only kept for reference)
- internal_text_id

Considering the structure mentioned above, a text can be fully identified and categorized by joining the corpus texts and metadata entities.

The **n-grams** are stored in an entity defined by the following fields:

- ngram_id (an autoincremented value)
- n_gramity (can have as possible values 2, 3, 4 or 5, representing the number of words in the n-gram)
- n-gram (the exact words combination)
- corpus_genre_id (an external reference to the corpus_genres entity)
- corpus_discipline_id (an external reference to the corpus_study_levels entity)
- corpus_gender_id (an external reference to the corpus_gender entity)
- corpus_study_level_id (an external reference to the corpus_study_level entity)

Considering this structure, an n-gram can be identified, categorized, and counted.

The **statistics** are memorized in a database entity with the following fields:

- statistics_id (an autoincremented value)
- romanian_texts_counter
- english_texts_counter
- romanian_words_counter
- english_words_counter
- romanian_characters_counter
- english_characters_counter
- 2, 3, 4, 5 -grams counter

Being pre-computed and stored in this format, the counters are easy to extract and to display both in the administration and the user interfaces.

3.1.6 Big Data Processing Performance

Since our system involves manoeuvring large amounts of data, the times measured for storing and retrieving the data can make the difference between a responsive system and a platform jammed by background operations.

During the development process we worked with a set of 50 texts, and we passed them through all the processing phases. Inserting and selecting data from the database had very small execution times, but once we increased the volume of processed data, some processing steps had to be rewritten.

The **corpus text files upload** depends only on the size of each file and storing them on the server is a very direct process that leaves nothing to the optimization. **Normalizing** the content, **processing**, and **storing** it in the database takes up to 2 seconds for an average of 1571 words per file.

The **metadata** file corresponding to a set of 500 texts takes under 2 seconds for being **uploaded** and under 10 seconds for **parsing and processing.** Each row from the metadata file is transformed into an insert or update in the *metadata* entity, and if necessary, in *corpus_genres*, *corpus_disciplines*, *corpus_genders* and *corpus_study_levels*. The execution times are good considering the amount of processed information.

The **n-grams** are isolated from an input text in less than 1 second/text. An average of 105.17 n-grams (pairs of 2, 3, 4 and 5 words) are detected per text file.

Inserting all the word pairs individually in the n-grams database entity takes an average time of 5 seconds / text. Cumulated with the number of texts, this leads to an average n-gram storing time of 5,000 seconds / 1000 files. This value represents a true problem, and the solution came from cumulating the insert operations into bulk inserts. All the insert syntaxes are grouped by 400 and the execution times were decreased dramatically, reaching the lowest average of less than 2 seconds.

The obtained execution times and the associated number of bulk inserts are represented in Figure 1.

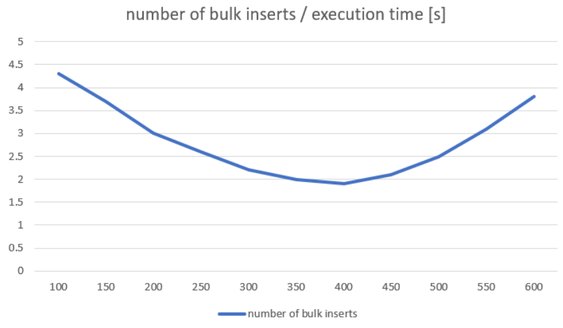

Figure 1: The number of bulk inserts / execution time.

Counting the n-grams in a table that is composed of several million records is a problematic issue. Determining the number of times a certain group of words occurs involves SQL syntaxes similar to this one.

```
SELECT count(n_gram) as occurences,
n_gram  FROM  `table_name`  WHERE
conditions group by n_gram order by
occurences desc;
```

At first, for an efficient memory usage, the *n_gram* field was declared as *varchar(50)* […]. The obtained execution times for such a routine took approximately 70 seconds for a table containing 3,500,000 records. Obviously, this time interval is totally un-practical, so we had to come up with a better solution.

The first part of the solution was to renounce the efficient memory usage in favour of the syntax rapidity. The *n_gram* field was declared as *char(50)* […] and the response times dropped immediately to an average of 25 seconds.

It's a known fact that SQL syntaxes involving SELECT and COUNT work much faster on numerical values than on arrays of characters. A very direct numerical representation of an array of characters is given by applying a hashing function.

Having those values stored in the database and rewriting the syntax as

```
SELECT count(hash) as occurences,
FROM `table_name` WHERE conditions
group by hash order by occurences
desc;
```

reduced the execution times to an outstanding 0.9 seconds for the same table with 3,500.000 records.

A small inconvenience appeared after this initial success and was due to the necessity of including the *n_gram* in the search. The following SQL syntax was used with similar results.

```
select  hash,  n_gram,  count(hash)
from   `table_name`  group  by  hash
order by count(hash) desc;

select   T1.hash   as   hash_value,
count(T1.hash)    as    occurences,
T2.n_gram  from  `table_name`  as  T1
join `table_name` T2 on T1. id=T2.id
group by T1.hash order by occurences
desc
```

Both syntaxes averaged a satisfying 4.5-5 seconds for 3,500,000 records. From these recorded times, approximately 2.5 seconds are produced by ordering the results. We tried to get the results unordered and to sort them programmatically, but the processing times increased with an average of 5 seconds so we adopted the database implicit sorting mechanism.

4 CORPUS INTERFACES

The Roger Corpus platform aims to be very close to the user and to trigger extensive background computations with simple clicks.

4.1 Administration Interface

After authentication, an administrator can have access to the following functionalities.

Texts and metadata management, where the administrator can:

- *Upload text files into the platform*; this process is followed by normalization, processing, and database storage; an extensive report is displayed to the administrator.
- *Display texts*; each text can be fully viewed, edited and deleted.
- *Set texts displaying criteria*; the filtering can be done by id, language, genre, discipline, study level, study year and author gender;
- *Upload metadata files*; each upload is followed by parsing and database storing operations in the corresponding entities.
- *View metadata*; a synthesis of genres, disciplines, study levels and genders is displayed; the metadata ids without matching texts and the texts without paired metadata is displayed.

Overall statistics, where the administrator can:

- *Generate overall statistics*: by a press of a button, the texts, texts by discipline, texts by language, characters by language, words by language and n-grams counters are computed; the date the statistics were generated is determined and displayed.
- *View overall statistics*: the values mentioned above are displayed in web format.

Detailed n-grams statistics, where the administrator can:

- *View n-grams statistics*: the n-grams can be filtered by language, genre, discipline, study level, study year, and author gender; the first 100 n-grams are displayed, and the full list of n-grams (thousands of records) can be downloaded in **.xlsx* format.

4.2 User Interface

Any visitor that accesses the Roger Corpus platform can benefit from accessing the following areas.

- *About*: a short description of the platform.
- *Corpus documentation*: a selection of downloadable documents.
- *Tutorials*: Corpus user guides.
- *Research*: an area for publishing various results related to the linguistic research performed by the academic staff that manages the platform.
- *Statistics*: an area in which the texts, words, characters and n-grams statistics are displayed.
- *Contact*
- *Terms and conditions*

After creating an account, in addition to the visitor options, a registered user can:

- *Perform Corpus searches*: by mentioning a sequence of words, the corresponding texts are displayed as a list; the user can refine the initial search by filtering the results by language, genre, and discipline.
- *Generate and download specific n-grams statistics*: by specifying the searched number of words, language, genre, discipline, study level, study year the first 100 n-grams are displayed; the results can be downloaded in **.xslx* format; the user has to manifest his agreement to the Roger Corpus platform's policy before downloading the selected n-grams;

5 CONCLUSIONS

This article presents a Corpus application that offers the tools for handling a large collection of texts written in English and Romanian. ROGER aims to expose the accumulated database and to offer to the registered users the possibility of filtering the stored information by language, genre, discipline, author's study year and learning cycle.

Both the English and the Romanian sub-corpora feature texts from eight different disciplines: (i) Humanities; (ii) Economics; (iii) Political Sciences; (iv) Engineering; (v) Computer Science; (vi) Law; (vii) Mathematics; (viii) Social Sciences. In each discipline, the students labelled the genre of their own writings as follows: (i) Essay; (ii) Scientific paper; (iii) Report; (iv) Bachelor thesis (BA); (v) Master's dissertation (MA); (vi) Case study; (vii) Summary; (viii) Literary analysis; (ix) Review; (x) Others (to be elaborated further). While samples of most genres can be found in both languages, there are genres which have only been collected either in English (e.g., CV, interview, documentation etc.) or Romanian (e.g., summary, reading notes, portfolio etc.).

The exposed data serves the purpose of investigating student writing practices in both their mother tongue and English as a Foreign Language.

The platform was designed and implemented as a cross-platform distributed web application.

The backend interface available to authenticated administrators provides the digital tools for managing the database stored texts and associated metadata. Also, it offers an extensive statistics mechanism that covers the corpus data composition, distribution, and usage. The quantified aspects target the words, characters, languages, study levels, genres, domains, and n-grams.

The frontend capabilities are offered to the registered users allowing them to search for specific keywords and to refine the obtained results by applying a series of filters.

As a technical challenge, manoeuvring high volume data was a serious problem generator. The registered processing times were sometimes bigger than a regular web application can afford. A series of optimizations were applied, including processing fragmentation, SQL syntaxes cumulation and database optimization techniques.

Overcoming the challenges brings additional value to the resulting instrument, which becomes the first European open-access corpus support platform. The platform offers user-friendly search options to a predefined original corpus (ROGER), whose compilation itself is a major contribution to the academic writing research community. The fact that the corpus is bilingual (Romanian and English), multi-disciplinary, multi-genre and multi-level makes it a valuable asset for any interested user that can access the platform for personal use, teaching, research or professional development.

ACKNOWLEDGEMENTS

The ROGER platform has been developed in the framework of the ROGER project (https://roger.projects.uvt.ro/), developed at the West University of Timisoara, Romania. ROGER was financed by the Swiss National Science Foundation, as part of the *PROMYS* program (project code IZ11Z0_166537), for a five-year period (2017-2022), through the research grant awarded to the project coordinator, Dr Madalina Chitez.

REFERENCES

Ackerley, K. (2021). 4 Exploiting a genre-specific corpus in ESP writing: students' preferences and strategies In M.Charles, & A. Frankenberg-Garcia (Eds.), *Corpora in ESP/EAP Writing Instruction: Preparation, Exploitation, Analysis* (pp. 78-99). New York: Routledge.

Alsop, S., & Nesi, H. (2009). Issues in the development of the British Academic Written English (BAWE) corpus. *Corpora*, *4*(1), 71-83.

AWICNET (2019), accessed March 1, 2022, https://roger.projects.uvt.ro/news/aprilie-2019-lansare-proiect-awicnet/

Callies, M. & Zaytseva, E. (2013a). The Corpus of Academic Learner English (CALE) – A new resource for the study and assessment of advanced language proficiency. In Granger, S., Gilquin, G., & Meunier, F. (Eds.), *Twenty Years of Learner Corpus Research: Looking back, Moving ahead* (Corpora and Language in Use - Proceedings Vol. 1). Louvain-la-Neuve: Presses universitaires de Louvain.

Callies, M. & Zaytseva, E. (2013b).The Corpus of Academic Learner English (CALE) – A new resource for the assessment of writing proficiency in the academic register. *Dutch Journal of Applied Linguistics, 2*(1), 126-132.

Chitez, M., Rogobete, R., & Foitoş, A. (2020). *Digital Humanities as an Incentive for Digitalisation Strategies in Eastern European HEIs: A Case Study of Romania*. In Curaj, A., Deca, L. & Pricopie, R. (Eds.), European Higher Education Area: Challenges for a New Decade (pp. 545-564). Springer: Cham.

Chitez, M., Bercuci, L., Dincă, A., Rogobete, R., & Csürös, K. (2021). *Corpus of Romanian Academic Genres (ROGER*). Roger-corpus.org. Retrieved [20, February, 2022], from https://roger-corpus.org/.

Chitez, M., & Bercuci, L. (2019). Data-driven learning in ESP university settings in Romania: multiple corpus consultation approaches for academic writing support. In: F. Meunier, J. Van de Vyver, L. Bradley, L. & S. Thouesny (Eds). *CALL and complexity–short papers from EUROCALL2019* (pp. 75-81), research-publishing.net.

Chitez, M. (2014). *Learner corpus profiles: the case of Romanian Learner English*. Linguistic Insights Series, Vol. 173 (Series Editor: Mau-rizio Gotti). Bern, Berlin, Bruxelles, Frankfurt am Main, New York, Oxford, Wien: Peter Lang.

Donahue, C. (2002). The lycée to university progression in French students' development as writers. In Foster, D., & Russell, D. (Eds), *Writing and Learning in Cross-national Perspective: Transitions from Secondary to Higher Education* (pp. 134-191). Urbana, IL: National Council of Teachers of English (NCTE) Press.

Donahue, C. (2009). "Internationalization" and Composition Studies: Reorienting the Discourse. *College Composition and Communication, 61* (2), 212-243. Available at: http://www.jstor.org/stable/ 40593441.

Foster, D. (2006). *Writing with authority: students' roles as writers in cross-national perspective*. Carbondale: Southern Illinois University Press.

Foster, D., & Russell, D. R. (2002). *Writing and learning in cross-national perspective: Transitions from secondary to higher education*. New Jersey: Lawrence Erlbaum.

Granger, S. (1996). From CA to CIA and back: An integrated approach to computerized bilingual and learner corpora. *Lund Studies in English*, *88*, 37–52.

Heuboeck, A., Holmes, J., & Nesi, H. (2009), *The BAWE Corpus Manual version II*. http://www.reading.ac.uk/internal/appling/bawe/BAWE.documentation.pdf (accessed 20 February 2022).

Johansson, S. (2003). Contrastive linguistics and corpora. In Granger, S., Lerot, J., & Petch-Tyson, S. (eds.), Corpus-based Approaches to Contrastive Linguistics and Translation Studies (pp. 31-44). Brill.

Kaiser, D. (2003). Nachprüfbarkeit versus Originalität. Fremdes und Eigenes in studentischen Texten aus Venezuala und Deutschland. In Ehlich, K., & Steets, A. (Eds.), *Wissenschaftlich schreiben – lehren und lernen* (pp. 305-325). Berlin: Walter de Gruyter.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., Rychlý, P. & Suchomel, V. (2014). The Sketch Engine: ten years on. Lexicography, 1(1), 7-36.

Kwon, H., Partridge, R. S., & Staples, S. (2018). Building a local learner corpus: Construction of a first- year ESL writing corpus for research, teaching, mentoring, and collaboration. *International Journal of Learner Corpus Research, 4*(1), 112-127.

McEnery, T. (2019). *Corpus linguistics*. Edinburgh University Press.

Nesi, H. (2008). BAWE: an introduction to a new resource. In A. Frankenberg- Garcia, T., Rkibi, M. Braga da Cruz, Carvalho, R., Direito, C., & Santos-Rosa, D. (Eds). *Proceedings of the Eighth Teaching and Language Corpora Conference* (pp. 239–46). Lisbon, Portugal: ISLA.

O'Donnell, M.B. & U. Römer. (2012). From student hard drive to web corpus (Part 2): the annotation and online distribution of the Michigan Corpus of Upper-level Student Papers (MICUSP). Corpora, *7*(1), 1–18.

Paquot, M., Hasselgård, H., & Oksefjell Ebeling, S. (2013). Writer/reader visibility in learner writing across genres: A comparison of the French and Norwegian components of the ICLE and VESPA learner corpora. In S. Granger, G. Gilquin, & F. Meunier, Twenty Years of Learner Corpus Research (Corpora and Language in Use - Proceedings; 1), Presses Universitaires de Louvain: Louvain-la-Neuve. Retrieved from https://www.uclouvain.be/cps/ucl/doc/cecl/documents/Paquot_al_LCR2011.pdf.

Römer, U., & O'Donnell, M. B. (2011). From student hard drive to web corpus (part 1): The design, compilation and genre classification of the Michigan Corpus of Upper-level Student Papers (MICUSP). *Corpora*, *6*(2), 159-177.

Siepmann, D. (2006). Academic writing and culture: An overview of differences between English, French and German. *Meta*, *51*, 131-150.

Trumpp, E. C. (1998). *Fachtextsorten Kontrastiv: Englisch-Deutsch-Französisch*. Gunter Narr Verlag.