Baltic J. Modern Computing, Vol. 12 (2024), No. 4, pp. 535–547
https://doi.org/10.22364/bjmc.2024.12.4.15

Developing a Digital Research Environment for Finnic Oral Poetry

Maciej JANICKI[1], Kati KALLIO[1,2], Mari VÄINA[3], Eetu MÄKELÄ[1]

[1] University of Helsinki
[2] Finnish Literature Society
[3] Estonian Literary Museum

maciej.janicki@helsinki.fi, kati.kallio@finlit.fi,
mari@haldjas.folklore.ee, eetu.makela@helsinki.fi

0000-0003-3981-8021, 0000-0002-3673-1409, 0000-0001-5309-2357, 0000-0002-8366-8414

Abstract. We present the digital environment developed within the FILTER consortium – a collaboration of computational scientists and folklorists from Finland and Estonia studying the Finnic oral tradition. We describe the technical aspects of an automatized data preprocessing pipeline which combines several archival collections of folk poetry into a single database. Two user interfaces utilizing this database are shown: the close-reading interface Runoregi and a data visualization application, as well as the search tool Octavo based on a Lucene index. We discuss the principles followed in the development of these tools and experiences gained in the process.

Keywords: digital humanities, data processing pipeline, data visualization, folk poetry, oral tradition

1 Introduction

The FILTER consortium[4] (2020–2024) was a cooperation of Finnish and Estonian folklorists and computational scientists with the aim of studying the vast collections of Finnic oral runosong tradition through novel, computer-aided means. We use three large text corpora from Finnish and Estonian archives, currently 283 571 texts, added with a smaller corpora of literary poems using similar poetic form.

In the present article we describe the complete computational architecture of our project, focusing especially on the more technical parts that have not been described in

[4] FILTER stands for 'Formulaic Intertextuality, Thematic Networks and Poetic Variation across Regional Cultures of Finnic Oral Poetry', Research Council of Finland grants No. 333138 and 333139. The work continues in the IKAKE project funded by the Kone Foundation 2024–2028. See `https://blogs.helsinki.fi/filter-project/` for more information.

earlier publications, as well as the interaction between the different components. The article serves two purposes: first, as a guide to the public code and data repositories that provide added value to the source collections and will continue to be used after the end of the funding period. Second, through describing the pipeline architecture it shows how the properties of modularity, loose coupling, extensibility and robustness to change became necessary and were implemented in practice. We describe the starting situation of the project (Sec. 2), the data processing pipeline (Sec. 3), and the different user interfaces (Sec. 4). We conclude with a discussion (Sec. 5) highlighting the principles followed when designing the system and its use in ongoing folkloristic research.

2 Context and Motivation

2.1 Collections of Finnic runo-songs

Several closely related Finnic peoples – Finnish, Karelian, Ludic, Ingrian, Votic, Estonian, and South Estonian – share a common poetic system with local variations, characterized by the use of similar poetic trochaic meter, ample but non-mandatory use of alliteration and parallelism, and similar performance practices. The same poetic idiom has been used for various genres, such as narrative and lyrical songs, wedding songs and calendrial songs, lullabies, charms, mocking songs, proverbs, riddles, and so on. Some poetic expressions (formulas) and motifs are shared wider across the Finnic area. Originally, the native terms for singing and saying applied to this style of singing, as this was the prevalent way of poetic-musical expression in wide area. In the course of modernisation and wider spread of other musical traditions, the specific terms for this tradition were needed. In Finnish research discourse, the most well known term ‘Kalevalaic poetry’ comes from the literary epic ‘Kalevala’ by Elias Lönnrot, the most significant and well known literary adaptation based on the traditional songs. In Estonian the established term for this tradition is *regilaul* (from Low German *rege-* or *rigenlied* ’row song, dance song’). To denote the common Finnic tradition, we prefer to use the term runo-song (from Fi. and Ka. *runolaulu* ’poem-song’, ’wisdom-song’, *runo* from Old Germanic), relating to the oral tradition in Finnish and Karelian language. (Kallio et al., 2017)

Despite some differences, the oral corpora (described in detail in Sec. 3.1) have the same basic metadata and text structure deriving from the long parallel developments and collaboration. The Finnish and Estonian folklore collections were recorded and organised as parallel projects relating to the national awakening of small linguistic minorities in the Russian empire during the 19th and early 20th centuries and the birth of the folklore studies as an academic discipline. Also the digitizing of the materials took place as a collaboration in the early 21st century, and the resulting XML format follows similar principles (see Klementtinen, 2006; Sarv and Oras, 2020). Yet, the corpora were kept separate until the FILTER project. Bringing them together enables and encourages wider comparative projects which have clearly been lacking in this field of study ever since the Soviet occupation of Estonia in 1940.

2.2 Requirements on a computational environment

As the corpora contain various biases and details affecting the interpretations, it is crucial for a folklorist to be able to move smoothly between distant and close reading, or computational views and individual texts. Further, accessing the metadata easily is of great importance: regions have their particular local traditions and recording histories, the recording time is essential in understanding both the cultural-historical and recording context of a particular item, each recorder had their own peculiarities and data producing issues, and the pre-existing type indices give a quick access to one possible interpretation of the larger poetic contexts, and intertextual relationships of the details across the data. In addition, visualizing computational results – or even just text and metadata queries – in maps and plots enhances greatly both the humanistic reading and conveying the results. In practice, the researchers often need to import the results from one tool or interface to another to process these further. The better the research system enables these movements, the more usable and profitable it is for the researchers.

On the other hand, studying larger-scale patterns in the corpus leads to use cases too complex and varied to build dedicated user interfaces for. Therefore, in addition to user interfaces we also need the ability to access the data directly from programming environments (e.g. ‘notebooks’). We wanted this transition to be as seamless as possible and thus designed many of the user interfaces to allow advanced functionality through supporting for example custom SQL queries. Finally, as both the corpus and the computational enrichments done to it evolved throughout the project, the platform needed to robustly handle both updates to the underlying data, as well as enable quick development and prototyping of new functionalities.

Itemized, our requirements were thus the following:

- automated re-runnability of data processing in response to upstream data changes,
- the ability to access the same underlying data from multiple different user interfaces and programming environments,
- the ability to quickly prototype new functionalities,
- as much versatility as possible in enabling both easy-to-use access for novices, as well as advanced access for power users.

3 The data processing pipeline

To answer the needs sketched in the previous section, we designed a computational environment illustrated Fig. 1. An automatized data processing pipeline combines the source collections and converts them to a unified format. They are stored in a central repository (database or index), from which they can be accessed by the UI tools. These processing stages are described in the remainder of this section.

3.1 Source collections

Our corpus combines materials from three corpora hosted by the Finnish Literature Society (FLS) and Estonian Literary Museum (ELM). The three core parts are of similar size (around 90,000–100,000 poems each):

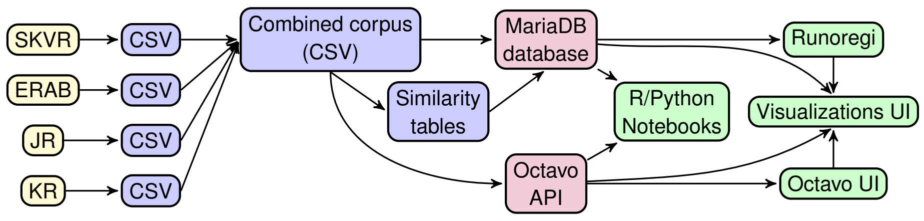


Fig. 1. The schema of the computational environment (yellow = source collection, blue = intermediate product, purple = query service, green = UI tool).

1. *Eesti Regilaulude Andmebaas* (ERAB, 'Estonian Runosongs' Database') is an online database[5] of mostly South and North Estonian traditions created at the Estonian Folklore Archives of the ELM on the basis of older machine-typed copies that have been scanned, are revised and automatically converted to XML (work is still in progress). It contains also orthographically harmonized text versions and a preliminary classification of texts (type index).
2. *Suomen Kansan Vanhat Runot* (SKVR, 'Old Poems of the Finnish [North Finnic] People') is a collection based on Finnish archival materials of Karelian, Finnish, Ingrian and Votic traditions, published as a book series 1908–1948, 1997 and made into a digital corpus at the beginning of the 2000s. The orthography is not harmonized. The texts have been systematically typologized as an effort of work group of the FLS at the end of 1990s.
3. *Julkaisemattomat Runot* (JR, 'Unpublished poems') at the FLS is an additional corpus of poetic texts compiled from the archival collections that were not included in SKVR or were collected after the publication of the regional volumes. It contains lots of other material than runosongs and is not systematized nor verified.

These three corpora are based on archival text recordings of oral tradition. In addition, during the project we started compiling a corpus of literary poems (Fi. *Kirjalliset runot*, hence KR), containing mostly Finnish literary works and edited folk song publications in runosong meter. The corpus is smaller than the former three, currently containing around 10,000 text items. It is based on public-domain literary works obtained e.g. from Project Gutenberg, Doria and Kotus (Institute for the Languages in Finland), and also includes well-known literary works like Kalevala or Kanteletar, as well as the Estonian national epic Kalevipoeg.

Two of the collections are currently publicly available by their host institutions on Github in XML and CSV format – SKVR[6] and ERAB[7], while JR and KR reside in private Github repositories with the intention of making them publicly available in near future.

[5] `https://www.folklore.ee/regilaul/andmebaas/`
[6] `https://github.com/sks190/SKVR`
[7] `https://github.com/rahvaluule/erab`

3.2 Preprocessing

The basis for our setting is combining the source collections into a single dataset. While the format of the individual collections is generally very similar, there are many differences in small details that require additional work before the collections can be combined. This is accomplished by a set of Python and shell scripts collectively referred to as the 'preprocessing pipeline'[8]. The end product of the pipeline is a set of CSV tables representing the combined dataset, which is also stored in a Github repository[9]. This is done both to facilitate external reuse of the dataset in the spirit of open data and open science, but also so that our own further processing pipeline can make use of as many standard components as possible.

The preprocessing is highly automatized, which allows recomputing the dataset at any time with minimal manual workload. Although the entire pipeline can be run by one `make` command, in practice it is split into several commands, as the resource-heavy similarity computation (see next section) is better run separately on a GPU-equipped computing cluster.

GNU make was chosen as the primary tool for managing preprocessing computations. In recipes for individual files, command-line tools (like `sed` or `jq`) and `csvkit` are preferred over own Python programs wherever possible. This allows a concise description of output files as a function of input files (a 'recipe'), often using widely known general-purpose tools. Further, `make` automatically manages dependencies and in case of updates to the source data, recomputes only the part of the dataset that is affected by the change.

The final stage of preprocessing is loading the data into both an SQL database as well as a Lucene index[10] for query access. In addition to the database import itself, the SQL generation requires some further data transformation, namely generating numeric keys and inserting them into tables. The code for producing the SQL database out of the set of CSV files is stored in a separate repository[11]. It can also be used to generate a database (and thus use the user interface tools) for other corpora, provided that they are supplied in the same CSV format as the one produced by our pipeline. Pilot projects were made with Latvian and Icelandic collections, where this part of the pipeline could be reused without changes (see Helgadóttir and Janicki 2024).

3.3 Similarity computation

Texts recorded from oral poetic tradition exhibit a lot of non-exact similarity: due to the folkloric and linguistic variation the plots, motifs, formulae and expressions acquire slightly different versions in the tradition, and they can be written down by collectors in different orthographies and exactness. The complex variation of the data makes identification of underlying similar patterns difficult. Thus, a large part of the computational

[8] `https://github.com/hsci-r/filter-pipeline`

[9] `https://github.com/hsci-r/filter-data`

[10] Lucene (`https://lucene.apache.org/`) is a popular open-source software library for text indexing and search. In order to perform searches, a text index needs to be precomputed using the functionalities provided by Lucene.

[11] `https://github.com/hsci-r/filter-db`

research in our project focused on automatically identifying similar units of text and utilizing the results of such computation both in close-reading (aiding the traditional humanistic research), as well as in quantitative analysis.

The automatized similarity computation in the pipeline consists of the following components:

1. Computing a table of verse similarities using the FAISS library for similar vector search (Johnson et al., 2017). The vectorization is based on character bigrams as described in (Janicki et al., 2023).
2. Computing a table of alignment-based poem similarities using the `matrix-align` algorithm described by Janicki (2022, 2023).
3. Computing a clustering of both verses and poems using the Chinese Whispers algorithm (Biemann, 2006) on the similarity table produced above.

The computationally most costly step is number 2., which involves computing a verse-level weighted edit distance-alignment between all pairs of poems. Due to the achieved optimizations it requires only around 4 hours on a single GPU-equipped node in a computing cluster. (Earlier publications (Janicki, 2022, 2023) reported much longer computation times, but it was later noticed that sorting poems by length greatly speeds up the execution of the `matrix-align` algorithm.)

Also the Chinese Whispers clustering is efficient, which allows us to calculate multiple variants of verse clustering with different parameters. Currently 5 different variants are computed. The `default` variant uses the threshold for cosine similarity that was determined in the evaluation as optimal (0.8) and raw frequencies of character bigrams. In order to better cope with euphonic repetitions (which do not fulfill our similarity definition), we test additional weighting methods: `sqrt` (take square roots of frequencies) and `binary` (reduce all non-zero values to one), which deemphasize repeating bigrams. Further, we include variants `tight` and `loose` with a different (higher/lower) similarity threshold than `default`.

3.4 The SQL database

The central point of out computational environment is a MariaDB database containing the combined corpus. It is the end product of the preprocessing pipeline and is used both for direct querying from e.g. notebook environments or SQL client applications, as well as for serving data to user interfaces.

SQL queries enable researchers to select the subcorpora they are interested in, either by metadata or by content queries, and to perform various counts and calculations with the large amounts of data, and finally, to download the selected data for further processing. This enables, for example, to measure and compare the prevalence of various features in the songs of different regions, singers, genres, and also to map or otherwise visualize the results. The results of verse and song similarity calculations capture the idea of folkloric type (verse type, song type) reliably enough to be used in large scale explorations (they are not always exact in minor details). Access to the well-organized database along with added similarity calculations, has substantially enhanced and widened the possibilities for exploratory analysis of the data compared to the previous state where the texts as well as metadata were stored in structured XML files.

3.5 Octavo API

Alongside the SQL database, the data is also loaded into Octavo[12], a custom REST API[13] built upon a Lucene index. Originating from earlier research projects of the Human Sciences-Computing Interaction (HSCI) research group at the University of Helsinki, the Octavo API is tuned to support efficient text search queries including complex boolean and regular expression queries, subsetting the data based on a combination of text and metadata constraints, as well as supporting functionalities such as vocabulary exploration.

4 User interfaces

4.1 Runoregi

Runoregi (Janicki et al., 2024) was developed as the primary tool for presenting the results of the similarity computation described in Sec. 3.3 with focus on close-reading. The central element of the interface is a text item with metadata. Using precomputed similarity tables available in the database, it shows similar texts, verse and poem clusters and helps to navigate between these (Fig. 2). Furthermore, it is able to compute on-the-fly passage searches (sequences of verses similar to a query sequence), side-by-side alignment views of two or more poems, and hierarchical clustering of chosen sets of poems. These features combine the computational similarity and metadata, e.g. poems of a certain indexed type can be compared using hierarchical clustering. The interface is implemented in Python using the Flask framework. It is deployed at `runoregi.fi` and the code is available on Github[14]. The name Runoregi is formed as a combination of the Karelian-Finnish and Estonian names for the tradition, but in Northern Finnic languages it can also be interpreted as ’a sledge carrying poems’, which in turn can be found as a motif in songs.

4.2 The visualizations UI

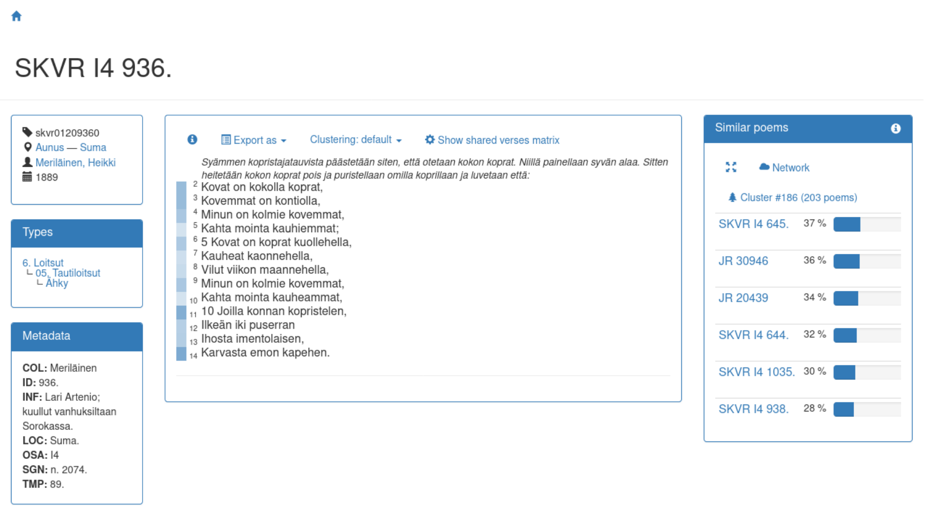
Complementary to Runoregi, we have designed a user interface for visualization of larger sets of data.[15] It contains two primary visualization types: 1) maps (Fig. 3) and 2) treemaps of genre categories and poem types from the type indices provided with SKVR and ERAB. A wide range of data queries is available: in addition to entire datasets, one can select e.g. a subset collected by a certain collector or in a certain time period, or plot a geographical distribution of a certain poem type, the typological distribution of

[12] `https://github.com/hsci-r/octavo`

[13] REST (REpresentational State Transfer) is a technique of exposing computational functionalities in the Web so that they are accessed by visiting a certain URL address – all parameters of the computation are specified in the URL. For more information on REST in general, see e.g. Pautasso et al. (2008).

[14] `https://github.com/hsci-r/runoregi`

[15] Currently deployed at `https://filter-visualizations.2.rahtiapp.fi` (though the URL might change), code available at `https://github.com/hsci-r/filter-visualizations`.



SKVR I4 936.

skvr01209360
Aunus — Suma
Meriläinen, Heikki
1889

Types

6. Loitsut
└ 05. Tautiloitsut
  └ Ähky

Metadata

COL: Meriläinen
ID: 936.
INF: Lari Artenio; kuullut vanhuksiltaan Sorokassa.
LOC: Suma.
OSA: I4
SGN: n. 2074.
TMP: 89.

Syämmen kopristajatauvista päästetään siten, että otetaan kokon koprat. Niillä painellaan syvän alaa. Sitten heitetään kokon koprat pois ja puristellaan omilla koprillaan ja luvetaan että:
2 Kovat on kokolla koprat,
3 Kovemmat on kontiolla,
4 Minun on kolmie kovemmat,
5 Kahta mointa kauhiemmat;
6 5 Kovat on koprat kuollehella,
7 Kauheat kaonnehella,
8 Vilut viikon maannehella,
9 Minun on kolmie kovemmat,
10 Kahta mointa kauheammat,
11 10 Joilla konnan kopristelen,
12 Ilkeän iki puserran
13 Ihosta imentolaisen,
14 Karvasta emon kapehen.

Similar poems

Cluster #186 (203 poems)

SKVR I4 645.	37 %
JR 30946	36 %
JR 20439	34 %
SKVR I4 644.	32 %
SKVR I4 1035.	30 %
SKVR I4 938.	28 %

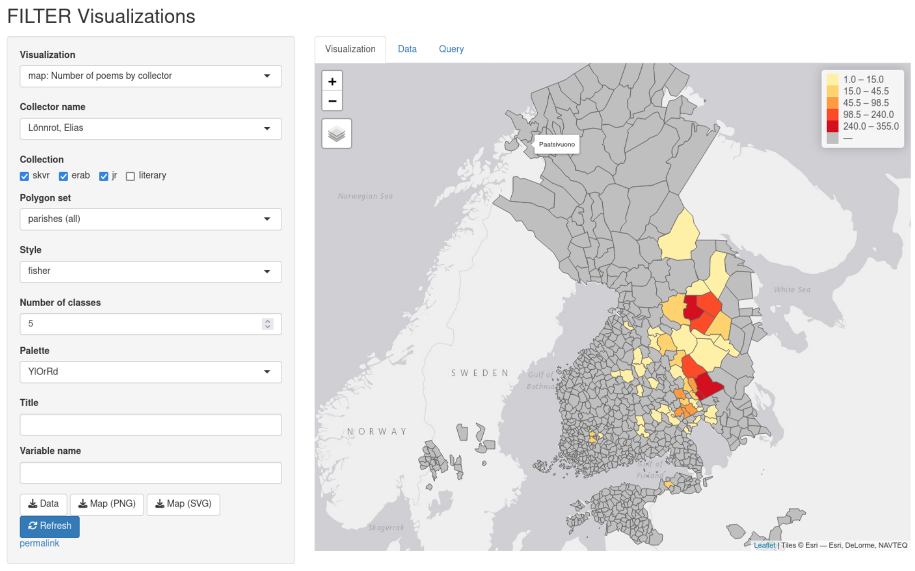
Fig. 2. Runoregi’s main view showing a poem text enriched with automatically computed similarity information (right).

collections from a certain parish or region, or create their own text and metadata query with Octavo or a direct SQL query. The application is implemented in R using the Shiny library.

Originally meant to eliminate the need to write dedicated R code for generating visualizations, the key feature of the application is its *extendability*. To add a new visualization, one only needs to specify the data query (usually an SQL query to the database) and parameters in YAML format – the application will then generate input widgets for the parameters. The final form of the query is shown to the user, which encourages experimentation and provides further flexibility, as visualizations with an entirely custom query (input in the app) and with one’s own data are also available.[16]

The application was also designed with *reproducibility* in mind. If visualizations for articles are generated with dedicated code, the code is commonly not attached to the publication and easily gets lost or stops functioning. If bugs occur, they are usually difficult to trace and not fixed after the publication, through which the visualization might show different data than intended. On the other hand, our preferred way of storing visualizations is through a *permalink* – a link to the interface which encodes the exact data query together with parameter values and visual parameters. If the query is a predefined one (e.g. ‘Number of poems by collector’), any bug fixes to the SQL code will apply also to permalinks generated before the fix (as they do not contain the SQL code, but only the name of the predefined query), so that already existing visualizations can be

[16] Enabling the user to execute arbitrary SQL code admittedly presents a security risk. So far we have mitigated it by restricting the database privileges of the app’s user to read-only. Still, it is easy to e.g. hang the database server by running a computationally heavy query. In the future, it might be necessary to restrict access to this feature.


FILTER Visualizations

Visualization: map: Number of poems by collector

Collector name: Lönnrot, Elias

Collection: ☑ skvr ☑ erab ☑ jr ☐ literary

Polygon set: parishes (all)

Style: fisher

Number of classes: 5

Palette: YlOrRd

Title:

Variable name:

🟨 1.0 – 15.0
🟨 15.0 – 45.5
🟧 45.5 – 98.5
🟥 98.5 – 240.0
🟥 240.0 – 355.0
⬜ —

Fig. 3. The Visualizations UI showing a map of collections by Elias Lönnrot.

corrected. The backwards-compatibility of the permalink format in case of changes to the data model is desirable and can be ensured most of the time.

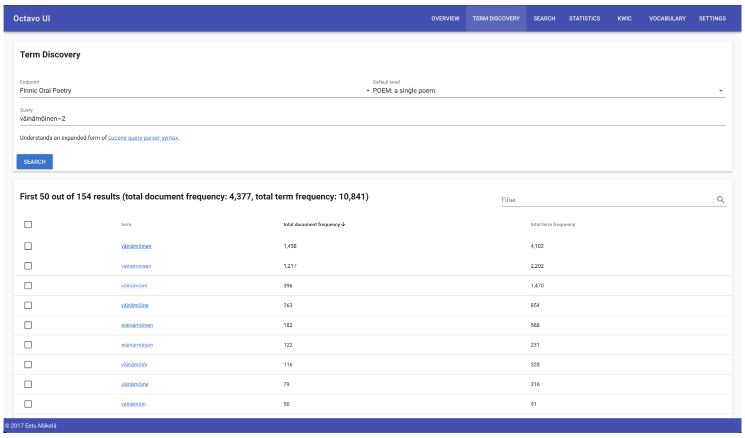
4.3 Octavo UI

The Octavo UI is a HTML/JavaScript UI built on top of the Octavo REST API, surfacing its search and vocabulary exploration functionalities for end users. Its main functionalities are allowing the user to search with complex combinations of metadata and content constraints, allowing them to see a configurable amount of context for each match in the UI, as well as a vocabulary discovery view (Fig. 4), which helps the uses to discover variant ways of expressing the same things across the poetic, dialectic and language variation pervading through the corpus. In addition, the UI offers links from the poems or the whole result set to Runoregi and the visualizations UI.

5 Discussion

5.1 The use of the system in folkloristic research

The flexible research system enables very heterogenous uses for various purposes of the tools. Some of the researchers in the project and near to it prefer using the SQL database, others Octavo or Runoregi, or a combination of these, and some like to import the results from SQL queries to process them further with other applications. For a user who knows the datasets, the setting of interlinked tools makes analysing wide sections



Term Discovery

Endpoint: Finnic Oral Poetry

Default level: POEM: a single poem

Query: väinämöinen~2

Understands an expanded form of Lucene query parser syntax.

First 50 out of 154 results (total document frequency: 4,377, total term frequency: 10,841)

term	total document frequency ↓	total term frequency
väinämöinen	1,458	4,102
väinämöisen	1,217	2,202
väinämöini	396	1,470
väinämöine	263	854
wäinämöinen	182	568
wäinämöisen	122	231
väinämöińi	116	328
väinämöińe	79	316
väinämöin	50	91

© 2017 Eetu Mäkelä

Fig. 4. Octavo UI's term discovery view showing different ways of spelling the mythic hero Väinämöinen in the corpus as found through a string similarity search.

of data in reasonable time feasible. One challenge is how to systematise or log the versatile, complex and sometimes meandering reading processes, or how to gather and hold together wider research corpora using several tools.

The tools and methods are used e.g. for understanding different patterns of folkloric variation (Sarv et al., 2024; Sarv and Järv, 2023; Helgadóttir and Janicki, 2024; Korpikallio, 2024); for thematic or content analysis (Seppä, 2020; Lintrop, 2024); and for understanding oral intertextuality (Kallio et al., 2022) or the relationships of oral tradition and literary works (Hämäläinen, 2020; Sykäri, 2020; Mäkelä et al., 2024). The Visualizations has also been used for creating similar map for rhymed folksong data not in use in the FILTER project (Sykäri, 2023). In addition to this, Runoregi and other results of similarity calculations are used in the Estonian Folklore Archives to process the typology in progress and to analyse e.g. the literary influences and the copies in the data.

Better possibilities for recognising similarity and analysing variation, visualizing these and assessing the basic characteristics and recording and curating history of the data open up new horizons for wider comparative research and understanding the variation of historical oral traditions better. In the future, our aim is to analyse in bigger scale and in more detail the patterns of oral variation and continuity, and to compare these to linguistic, ethnic and local histories.

5.2 Design principles and practices

As Barats et al. (2020) rightly point out, the problem of many Digital Humanities projects is low sustainability: tools developed in the projects are often abandoned and not usable after the end of the funding. Among the solutions proposed to this problem, modularity and interchangeability of components have been emphasised (Lee et al., 2020). This bears some similarity to the concept of agile development known in the industry, although an interdisciplinary research project presents some special challenges:

an extremely small development team (not seldom just one developer), very rapidly evolving requirements (to the point where writing tests becomes impracticable), or the need of combining stable pipeline elements with very experimental ones.

Even so, we have tried to follow the principles of agile development taught in classic works like Hunt and Thomas (2000); Fowler (2018); Martin (2018). In our case, this amounted mostly to: 1) implementing new features only when they are needed, 2) frequent refactoring, 3) design of the tools according to the single responsibility principle ('do one thing and do it well') and orthogonality principle ('changes in one component should not require changes in others'). Of these, especially refactoring needs to be highlighted, as it is often neglected in research prototypes.

The release of most of the source code was hampered by its dependence on the text corpora, which were initially not publicly available. We mitigated this by isolating computational methods to general-purpose, project-independent packages `shortsim`[17] (short string similarity toolkit) and `matrix-align`[18] (large-scale weighted sequence alignment), which were released at an early stage. Thanks to a dialogue with the institutions maintaining the corpora, the most important collections were released on Github as well, which in turn made it worthwhile to release our preprocessing code. We did not hesitate to share experimental features with outside collaborators as well, and continuously worked on improving the already implemented functionalities. Admittedly, the project could have benefitted even more if the source code was made open earlier.

Finally, we have kept in mind the principle that data is easier to understand than code ('show me your tables and I won't need your flowcharts', Brooks 1975). We have thus used plain data formats (mostly CSV) and made sure that for all code that we release, the data it works on is available as well. Already the format of the source collections is quite self-explanatory, so that we initially managed to build computational tools and user interfaces for these collections without using any documentation. We have striven for the data produced by us to be equally easy to interpret.

5.3 Conclusion

As the FILTER project is ending, the computational scientist-folklorist collaboration on Finnic oral poetry materials is expected to continue and widen. The tools will be developed further in an open source model and the user base among folklorists is growing. We have shared our thoughts and experiences on building this computational environment to help its future users and developers, as well as to provide useful insights for developers in similar interdisciplinary projects.

References

Barats, C., Schafer, V., Fickers, A. (2020). Fading away... the challenge of sustainability in digital studies, *Digital Humanities Quarterly* **14**(3).

Biemann, C. (2006). Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems, *Proceedings of TextGraphs: The First Workshop on Graph Based Methods for Natural Language Processing*.

[17] `https://github.com/hsci-r/shortsim`

[18] `https://github.com/maciejjan/matrix-align`

Brooks, F. P. (1975). *The Mythical Man-Month: Essays on Software Engineering*, Addison-Wesley.

Fowler, M. (2018). *Refactoring: Improving the Design of Existing Code*, 2nd edn, Addison-Wesley.

Helgadóttir, Y. S., Janicki, M. (2024). Text clustering of Icelandic post-medieval Þulur: Explorations using the Runoregi interface, *DHNB2024 Conference Proceedings*, Vol. 6 of *Digital Humanities in the Nordic and Baltic Countries Publications*.

Hunt, A., Thomas, D. (2000). *The Pragmatic Programmer: From Journeyman to Master*, Addison-Wesley.

Hämäläinen, N. (2020). Säe säkeeltä. Väinö Kaukosen säetutkimukset Kalevalasta. [Verse by verse: Väinö Kaukonen's verse studies of Kalevala.], *Sananjalka* **62**, 215–235.

Janicki, M. (2022). Optimizing the weighted sequence alignment algorithm for large-scale text similarity computation, *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, Association for Computational Linguistics, Taipei, Taiwan, pp. 96–100.

Janicki, M. (2023). Large-scale weighted sequence alignment for the study of intertextuality in Finnic oral folk poetry, *Journal of Data Mining and Digital Humanities* .

Janicki, M., Kallio, K., Sarv, M. (2023). Exploring Finnic oral folk poetry through string similarity, *Digital Scholarship in the Humanities* **38**(1), 180–194.

Janicki, M., Kallio, K., Sarv, M., Mäkelä, E. (2024). Runoregi: A user interface for exploring text similarity in oral poetry, *DHNB2024 Conference Proceedings*, Vol. 6 of *Digital Humanities in the Nordic and Baltic Countries Publications*.

Johnson, J., Douze, M., Jégou, H. (2017). Billion-scale similarity search with GPUs, *arXiv preprint arXiv:1702.08734* .

Kallio, K., Frog, Sarv, M. (2017). What to call the poetic form: Kalevala-meter or Kalevalaic verse, regivärss, runosong, the Finnic tetrameter, Finnic alliterative verse or something else?, *RMN newsletter* **12-13**, 139–161.

Kallio, K., Janicki, M., Mäkelä, E., Sarv, M. (2022). Recognising intertextuality in the digital corpus of Finnic oral poetry: Experiment with the Sampo cycle, *DHNB2022 Conference Proceedings*, Vol. 4 of *Digital Humanities in the Nordic and Baltic Countries Publications*, pp. 279–287.

Klementtinen, P. (ed.) (2006). *Ei se synny synnyttämättä: Selvitys digitointiprojektin vaiheista ja työprossesseista. [It will not be born without giving birth: A report of the digitization project's stages and work processes.]*, Suomen Kirjallisuuden Seura.

Korpikallio, S. (2024). Sähköistetty koeluenta vanhoihin runoihin. [Digitized reading of old [Kalevalaic] poems], *Elore* **31**(1), 90–103.

Lee, A. S., Chiarawongse, P., Guldi, J., Zsom, A. (2020). The Role of Critical Thinking in Humanities Infrastructure: The Pipeline Concept with a Study of HaToRI (Hansard Topic Relevance Identifier), *Digital Humanities Quarterly* **14**(3).

Lintrop, A. (2024). Kosmogooniline hari ja selestiline kiik. [The cosmogonic comb and the celestial swing], *Keel ja Kirjandus* **67**(3), 219–237.

Martin, R. C. (2018). *Clean Architecture: A Craftman's Guide to Software Structure and Design*, Prentice Hall.

Mäkelä, E., Kallio, K., Janicki, M. (2024). Sources and development of the Kalevala as an example for the quantitative analysis of literary editions and sources, *DHNB2024 Conference Proceedings*, Vol. 6 of *Digital Humanities in the Nordic and Baltic Countries Publications*.

Pautasso, C., Zimmermann, O., Leymann, F. (2008). RESTful Web services vs. "big" Web services: Making the right architectural decision, *Proceedings of the 17th International World Wide Web Conference*.

Sarv, M., Järv, R. (2023). Layers of folkloric variation: Computational explorations of poetic and narrative text corpora, *Folklore: Electronic Journal of Folklore* **90**, 1–32.

Sarv, M., Kallio, K., Janicki, M. (2024). Arvutuslikke vaateid läänemeresoome regilaulude varieeruvusele: "Harja otsimine" ja "Mõõk merest". [Computational views on the variation of the Finnic runosong: "searching for the comb" and "a sword from the sea"], *Keel ja Kirjandus* **67**(3), 238–259.

Sarv, M., Oras, J. (2020). From tradition to data: The case of Estonian runosong, *Arv. Nordic Yearbook of Folklore* **76**, 105–117.

Seppä, T. (2020). Katsooko metsä ihmistä? Suomen Kansan Vanhat Runot ja lajienvälinen kommunikaatio. [Does the forest look at the human? Old Poems of the Finnic People and the interspecies communication], *Kirjallisuudentutkimuksen aikakauslehti Avain* **17**(4), 22–45.

Sykäri, V. (2020). Digital humanities and how to read the Kalevala as a thematic anthology of oral poetry, *Arv. Nordic Yearbook of Folklore* pp. 29–54.

Sykäri, V. (2023). Kakssanasen sanat. Rekilaulu suullisen komposition ja variaation mikrokosmoksena. [Words of the ‘two-worded’ songs. Finnish rhymed songs (rekilaulu) as a microcosmos of oral composition and variation.], *Sanojen luonto. Kirjoituksia omaehtoisen ilmaisun poetiikasta ja ympäristöistä*, SKS, pp. 164–197.

Received November 28, 2024 , accepted December 3, 2024