# The DHmine Dictionary Work-flow: Creating a Knowledge-based Author's Dictionary

***Tamás Mészáros[1], Margit Kiss[2]***
*[1]Budapest University of Technology and Economics, [2]Institute for Literary Studies, Hungarian Academy of Sciences*
*E-mail: meszaros@mit.bme.hu, kiss.margit@btk.mta.hu*

**Abstract**

Digitalized author's dictionaries could play an important role in humanities research. Not only could they provide better ways to study an individual author's vocabulary, but they could also act as a knowledge source for other computer-based methods. We present the process of making an author's dictionary of headwords, writing variations, word forms and corpus citations extended with part-of-speech, linguistic, literary and semantic information. We also describe how this extended dictionary incorporates knowledge from linked open data sources and from critical annotations and builds an RDF knowledge base attached to the dictionary. The result is a vast knowledge source about an author's oeuvre that can be studied and used to enhance corpus analysis. We demonstrate our method on processing a large text corpora of 1.5 million words from the 18th century and on creating the digital author's dictionary of Kelemen Mikes.

**Keywords**: author's dictionaries, knowledge-based systems, corpus analysis, linked open data

## 1 Aim of the Research

The ongoing DHmine project at the Budapest University of Technology and Economics aims to create a software tools to support various digital humanities (DH) research tasks (Mészáros 2016). In cooperation with the Institute for Literary Studies of Hungarian Academy of Sciences, we processed the works of Kelemen Mikes, an 18th-century author often called the "Hungarian Goethe" (Kiss 2012).

Our main goal was to create an author's dictionary of Kelemen Mikes. This was a groundbreaking work since this era is rather underrepresented in computerized corpus building, and no complete digital author's dictionary had been created in Hungarian language before. Our aim was thus to establish a work-flow for creating such dictionaries and also to demonstrate the possible benefits of information technology in this field.

We concentrated our work on two main aspects: increasing the efficiency of the dictionary-making process by utilizing various software tools, and taking a step beyond data-centric digitalization and introducing knowledge-based methods in creating and using the dictionary. Our research aim was to develop methods for incorporating various kinds of knowledge in digital author's dictionaries and then utilize them in corpus analysis.

## 2 Previous Work

Computerized tools play an increasingly important role in humanities research. They provide efficient tools for storing, searching, retrieving and displaying digitalized texts, they also vastly improve the

efficiency of various research tasks, including making concordance lists and performing many different kinds analyses on the collected data.

Creating computerized concordance lists has been a focus of humanities researchers for many decades now. Roberto Busa (Busa 1980) was a pioneer in this field, creating a list of concordance called *Index Thomisticus* in 1951. Since the 1950s research and development in computerized lexicography has yielded significant results by processing the works of Kant, Shakespeare, Goethe, Dante and many other authors. In the Hungarian language the work of Ferenc Papp in the 1970s is also notable. He was processing Ady's oeuvre and emphasized the importance of computer-created concordance as a raw material of author's dictionaries (Mártonfi 2014).

Although a concordance list itself is a valuable data source for lexicography research, it can be augmented with many different kinds of information to form a dictionary of an author's oeuvre. Completing it with grammatical, semantical, stylistic, historical or cultural information yields a vast knowledge store for research (see Mattaush 1990: 1552-1553; Mártonfi 2014). The resulting author's dictionary is "a type of reference work which provides information on the vocabulary of a specific author." (Hartmann & James 2001: 10). Many author's dictionaries have been created in various languages (Goethe, Schiller, Thomas Mann, Bertolt Brecht, Dante, Ibsen, Shakespeare, etc.). In the Hungarian language the Petőfi dictionary is a notable example (J. Soltész et al. 1973-1987).

Paper-based dictionaries often face problems due to space limitations and the substantial manpower required to complete them. These dictionaries were typically created slowly, with a detailed entry structure and a sophisticated meaning description, stylistic qualifications, phraseological references, and so on, ad computerized tools can help to overcome the related time and space limitations. They provide a virtually endless storage capacity and also speed up certain work phases of the dictionary-making process. Our aim was thus to provide such tools for processing Mikes' oeuvre, for the dictionary making process and for storing and accessing the related corpus and vocabulary.

Recently some pioneering research work has applied artificial intelligence (AI) techniques in the field of digital humanities. AI is a vast field of research that tries to grasp human knowledge and mimic our behavior in problem solving. Knowledge-based systems are especially successful in representing and using human knowledge in computerized systems. As digital humanities projects have already accumulated a lot of data in digital form, it is a natural step forward to transform these data into knowledge and utilize them to support human researchers. This can be done by, for example, connecting contextual knowledge to a corpus (Bartalesi et al. 2015), semantic corpus annotations, knowledge extraction from text using NLP techniques, or adding sentiment information to lexicons (Nugues et al. 2016). Our research focused on how knowledge can be incorporated, represented and used in digital authors' dictionaries.

## 3 The Dictionary-making Process

We present the process of making the digital Mikes dictionary based on the works Kelemen Mikes, a famous Hungarian author from the 18th century.

### 3.1 The Mikes Oeuvre

Our basis for compiling the dictionary was the critical edition of Mikes' work created by Lajos Hopp (Hopp 1966-1988). This contains Mikes' letters written from exile on almost 6,000 pages, and it also contains the critical annotations and research notes of Lajos Hopp. The initial electronic form of the corpus was created in cooperation with the National Széchényi Library. They performed the OCR

process on the scanned documents. After manually correcting scanning and recognition errors we created an electronic version of the original text and the critical annotations. This was the initial corpus for our dictionary-making process.

### 3.2 Creating the List of Concordance with Full-text Citations

The process started with creating a full concordance list of words with attached full-text citations. This automatically generated concordance list was not only linked to the appropriate corpus locations, but entries were also extended with full-text citations. An entry in this list contains a word form (e.g. "*Constáncinapolyban*") and all its occurrences in the corpus with full-text quotes. It is important to note that digital dictionaries do not have the space limitations that their paper-based versions exist with. Thus we generated all citations in their full-length form in the concordance list, helping researchers to analyze them and create the dictionary entries later on.

We chose the XML standard for storing the concordance list and also for authoring the dictionary. This is a common choice for DH researchers, as it is very flexible in storing various kinds of data in a self-describing format, and it is also easily processable by computerized tools. We used a subset of the XML TEI standard to encode the content: tags were marking headwords, writing variations, word forms, corpus citations, source references, and so on. We also developed a simplified syntax to support researchers in using their favorite non-XML text editor during the authoring process of the dictionary.

The following two examples show excerpts with citation data for the word form "*Constáncinapolyban*".

Simplified XML syntax showing the original sentence, the word form marked and the location in the corpus (TL 250):

(1) ö parancsollya. innét csak hamar <I>Constáncinapolyban</I> megyünk, azért hogy meg lássuk (TL 250)

The above sample encoded in XML TEI after an automated transformation is as follows:

(2) <cit type="example"><quote>ö parancsollya. innét csak hamar <I>Constáncinapolyban</I> megyünk, azért hogy meg lássuk </quote><bibl>TL 250</bibl></cit>

The result of the automated concordance-making process was a huge text document (roughly 60,000 pages, 150MB). This full concordance list contained roughly 260,000 different word forms with 1.6 million citations. This formed the base data set of making the dictionary.

### 3.3 Creating Headwords

This huge concordance list was then processed manually by researchers to identify the list of headwords and to attach the word forms to them. This time-consuming process required many man-months of expert work. During this, researchers corrected the errors of the automatically recognized word forms, identified the proper headwords and attached the word forms and citations to them. They also identified the headword's modern form and common writing variations and extended the dictionary entries with this information.

For the above example the headword "*Constancinapoly*" was created. In simplified XML form (excerpt):

(3) <U>Konstantinápoly</U>
(4) <Q>Constáncinapoly</Q>
(5) <B>Constáncinapolyban</B> – 1
(6) ö parancsollya. innét csak hamar <I>Constáncinapolyban</I> megyünk, azért hogy meg lássuk (TL 250)

For this headword researchers identified 10 writing variations (Q tag), 22 word forms (B) in the corpus, they attached 99 citations (I) and also noted its "*Konstantinápoly*" (Constantinople) modern form in Hungarian (U tag).

After processing all word forms and creating their respective headwords, the core of the digital Mikes dictionary was created (Kiss 2012).

### 3.4 Extending the Core Mikes Dictionary

Our main goal was to build an extended dictionary that acts as a knowledge source for literary research and also supports corpus analysis. In order to achieve this we extended the dictionary entries in various ways.

#### *3.4.1 Incorporating Additional Lexicographic Knowledge*

We analyzed dictionary data (word forms and citations) and extended them with lexical and other attributes. For example, many Turkish words were used in the texts written by Mikes (the author spent many years in exile in Turkey), and he also created his own words that could not be found in other dictionaries. These were marked in the dictionary.

A rather complex task was to extend the citations with part-of-speech (POS) information. In order to perform this we developed a tool to automatically identify the POS roles of word forms based on the common rules of the Hungarian language grammar of that time period. After automatically extending the headwords with this information the researchers corrected these manually by reviewing all word form uses in the attached full-text citations. When a word form had multiple POS roles in the citations they were also marked manually.

#### *3.4.2 Adding Semantic Knowledge to Headwords*

The other type of knowledge we introduced in the dictionary was semantic information about headwords.

Word forms in Hungarian have changed significantly since the 18$^{th}$ century. Adding the modern word form to headwords already helps greatly in accessing more information about them.

We also marked named entities in the dictionary. Headwords of place and person names were enriched with semantic markings. In order to speed up this process we developed automated entity recognizers to detect proper names of persons and geographic locations in the corpus (Mészáros 2016). The proposed entities were then reviewed by human experts and this information was also added to the appropriate headwords. For example, the "*Konstantinápoly*" headword was marked as a place name.

It is important to note that although these extensions describe knowledge about headwords they do not alter the structure nor the usage of the dictionary, they merely store additional information about headwords.

### 3.5 Linking the Dictionary with Knowledge Sources

Our aim was to create a knowledge base that grows beyond the structure (and capabilities) of a traditional author's dictionary. In order to achieve this we took a step forward in extending the dictionary to better utilize knowledge-based methods and tools. To implement this change we decided to link the dictionary with additional data sources and store this information as a knowledge base attached to the dictionary.

The dictionary is already linked to the corpus in many ways through word forms and sentences, but there are already other data sources available that can be referred to. Firstly, the set of critical annotations is a natural place to investigate. Secondly, the semantically enriched headwords can be linked to knowledge sources related to these concepts. For example, the recognized named entities (e.g. person and place names) can be connected to Linked Open Data (LOD) like DBpedia (Auer S. et al. 2007) sources that contain more information about them. We explored these two possibilities during our research and present the results in the following sections.

### *3.5.1 Attaching Research Notes*

The first candidate to attach more knowledge to the dictionary was the set of critical annotations created by Lajos Hopp (Hopp 1966-1988). Critical annotations are a primary knowledge source when researching an oeuvre. They are traditionally text fragments attached to various parts of the corpus. In contrast to the author's texts we can observe that they usually have a more or less well-formed structure and they can be categorized based on their primary purpose: linguistic, historic, external reference, etc.

Lajos Hopp wrote more than 5,000 research notes about various parts of the Mikes oeuvre. He created his annotations in a more or less standardized form: a citation and a reference to the corpus, his researcher note, and references to external documents. The following example shows a note about the usage of " *Constantinapolyban*" (excerpt):

(7) [1.]
(8) 0 Constantinapolyban — Előfordul még Constantinápoly, Constancinapoly, Constancinápoly, Constáncinápoly [...]

We can observe the corpus reference in lines 7 and 8 of the above example. Line 7 selects the first letter of Mikes, and the very beginning of line 8 specifies that the word can be found at the beginning of that Mikes' letter. Line 8 in the example also shows the original text "*Constantinapolyban*" and the attached note (after a "—" sign) that lists other word forms and describes historical notes and Mikes' personal attachment to the city.

By observing this structure we were able to create a software tool that automatically transforms these annotations into a structured XML TEI form including citations, references and the annotations themselves. The following example shows the result of this automatic transformation.

(9) <note type="critical annotation">
(10) <text>Előfordul még Constantinápoly, Constancinapoly, Constancinápoly, Constáncinápoly [...]</text>
(11) <cit><quote>Constantinapolyban</quote><bibl unit="line" from="0">TL.1</bibl></cit>
(12) </note>

We also analyzed what kinds of annotations were created by Lajos Hopp. We categorized them into 10 subtypes (like historical, social, geographical, grammatical, and so on), and then manually labeled the annotations based on their categories to determine what type of knowledge is stored in them. For example, the above note was categorized as geographical (G) and historical (H). To store these categories line 9 in the previous example was changed to

(13) <note type="critical annotation" subtype="G,H"> ...

From the XML version of the critical annotations we created a database and automatically attached its entries to the appropriate parts of the dictionary and to the corpus by identifying word forms in the citations and by extracting references from the XML tags and attributes. These annotations and their links to other data elements were the first type of knowledge attached to the dictionary.

### *3.5.2 External Knowledge Bases*

In order to incorporate more knowledge about headwords we explored external knowledge sources. DBpedia (Auer et al. 2007) is a well-known LOD data source that contains a vast amount of knowledge. For example, it contains many kinds of information related to the headword "*Constantinople*": it is a general concept that is a subject of many other entities, it is one of the names of Istanbul, a geographic location, it was also the capital city of many empires, etc.

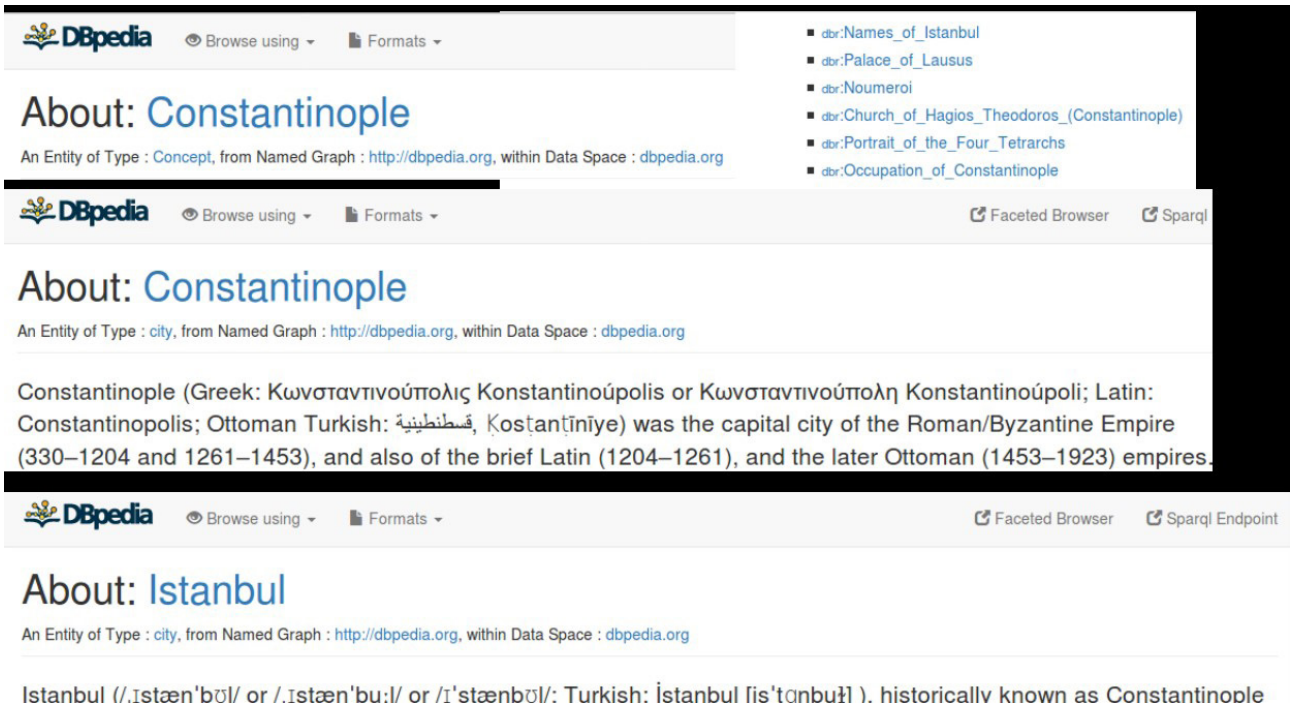


Figure 1: Excerpts of DBpedia entries related to Constantinople

Obviously, it is not suitable to incorporate all this knowledge in the dictionary. Since we focused on named entities we decided to narrow queries to these. Even this dataset was too large to include, so further selection had to be made based on the possible applications of this knowledge. We followed a restricted set of DBpedia links like geographical location, person, temporal and name alternatives to select the proper subset of information.

We developed an automated software tool to connect dictionary entries to these external sources and retrieve a restricted set of data from them. This tool builds a small-scale local mirror of selected knowledge pieces retrieved from the sources. As named entities were already marked in the dictionary, and their modern word forms were also included, it was fairly easy to perform DBpedia lookups to find the appropriate external records. The list of entries proposed by this tool was also manually reviewed by researchers before starting the knowledge transfer in order to solve disambiguation problems. The selected entries were transferred to a local graph database. The details of this will be discussed in the next chapter.

## 4 Implementing the Extended Author's Dictionary

The main goal of the dictionary is to support literary research by storing data and providing various functions to retrieve the stored information. We explored several use cases regarding authoring,

storing and accessing the information in order to develop appropriate data models, storage methods and system functions, and to select the appropriate software tools to implement them.

## 4.1 Data and Knowledge Storage

The extended dictionary contains many types of data, including dictionary entries, corpus text, critical annotations and semantic knowledge. There are also various types of connections among these entities: dictionary entries refer to the corpus through citations and their references, words in the corpus can be found in the dictionary as word forms, critical annotations are attached to the corpus but they are also linked to the relevant headwords, and pieces of knowledge from external sources are attached to headwords.

An SQL database system is a natural choice in order to store the electronic version of the dictionary. A NoSQL database was chosen for corpus texts and critical annotations, and a graph database for describing semantic and structural knowledge in a unified system.

### *4.1.1 Headword Database*

As the headwords were authored using a predefined XML schema (a detailed and structured data model) importing them to a database was a straightforward process. Following the headword structure we designed a simple database model to store them. During database modeling we also took possible data queries into consideration and created separate data tables and indexes for those pieces of data. The system was implemented using the MySQL database engine.

### *4.1.2 Corpus Storage*

As our goal was to maintain a complete list of text occurrences at each dictionary entry we stored the entire corpus in the system. Since our work concentrates on the dictionary and we store corpus only for related citations, we decided not to use complex corpus analysis tools but choose a simple solution to store the attached text fragments.

Corpus texts are encoded using XML TEI, but they are far less structured than dictionary entries. In contrast to the data-centric dictionary XML data, the corpus XML is a document-centric data set. In order to store it a schemaless (NoSQL) database system, MongoDB was chosen. NoSQL databases have the required flexibility to store and organize such documents, and they also provide scalability and performance with regard to accessing them. The corpus was stored at various levels of granularity in the database (e.g. chapters, sections, paragraphs, lines, etc.) to ensure that we can create different kinds of relations between the dictionary and corpus.

Critical annotations are traditionally text fragments attached to various parts of the corpus. Our system stores these text annotations using the same technique in the NoSQL database. It also uses the same referencing system: annotations refer to corpus entities at various granularity levels, and the word forms found in annotations can be looked up in the dictionary.

### *4.1.3 Knowledge Store*

The incorporated external knowledge and internal references between dictionary entries, corpus parts and critical annotations can all be represented as graphs. Since external knowledge bases (LOD sources) use the Resource Description Framework (RDF) to store and exchange this kind of information, we decided to follow this practice to avoid transformation during the import process.

RDF is a very flexible system to represent knowledge. Its atomic data entity is a triplet consisting of a subject, a predicate and an object. Together they represent a fact. From these building blocks a

graph can be constructed to store a knowledge base. Due to this graph storage it can be extended or modified without altering previously inserted knowledge. Its flexibility is very useful if we do not know in advance what kind of knowledge will be stored or how it will be extended later. In order to implement the knowledge store we have selected the RDF4J open source software.

In addition to the retrieved DBpedia RDF dataset, we also assigned resource identifiers for corpus and dictionary entries and stored these and their relations in the RDF database. This way we stored the knowledge about their relations and also the relevant semantic information in a unified system.

RDF also provides a very powerful graph query language, SPARQL. Inserting knowledge as RDF triplets is a straightforward process but performing a query on a graph database is a more complex task, since we may want to formulate complex conditions for graph matches. SPARQL allows such constructs. We designed many kinds of query functions in SPARQL in order to support corpus and dictionary information retrieval, as detailed in the next section.

### 4.2 Query Functions

The main use of the system is to access dictionary data. The input is typically a keyword (headword, writing variation, word form) and its attributes (e.g. part-of-speech role), and the result is a dictionary entry. It is also common to restrict the query by selecting a part of the corpus in which the keyword has to be found. These queries were implemented in SQL using the dictionary database.

For completeness, we also developed corpus search methods attached to the dictionary. In this case the input is again a keyword and its attributes, but the result is a corpus entry. This is similar to dictionary search, but it also supports query expansion to replace keywords with other data (e.g. with other word forms), while keyword and corpus normalization are used to provide better precision and recall during the search process.

Finally, knowledge retrieval is the third kind of search function that allows queries regarding the semantic information of the stored data. In this case the input is a complex query that specifies entities (e.g. places or persons, dictionary and corpus entries), their attributes and relations to other entities. These functions were implemented in SPARQL that queries the RDF knowledge store of semantic and reference information.

One interesting aspect of these query functions is the query expansion backed by the RDF knowledge store (Varga et al. 2003). The search engine is capable of recognizing concept and entity names in the user's query and it is able to replace them with related keywords when needed. For example, it is able to answer queries like

(XX) Retrieve headwords related to cities

By using knowledge from external data sources like DBpedia the following complex query can be also answered

(XX) Retrieve citations related to geographic locations within 100 km of Constancinapoly.

The system identifies that *Constancinapoly* is a writing variation of Constantinople using the dictionary. It also finds out from the attached RDF store that it is a city with a known geographic location. To answer the query it searches for other geographic locations in the knowledge base that are within the given range. After finding them it retrieves the relevant citations from their dictionary entries and presents the results back to the user.

### 4.3 Web Interface for the Extended Dictionary

We created a web-based software tool to import, store, query and display the corpus, the dictionary and critical annotations. This tool provides an administrative interface for importing XML data (the dictionary, corpus and annotations) and it also has a user interface for the previously outlined query functions and for displaying their results. The web interface was implemented using the open source ProcessWire PHP framework with additional modules developed for XML import and displaying dictionary entries and corpus annotations.


Figure 2: The Mikes dictionary web interface showing a dictionary entry (excerpt) and an original letter from Kelemen Mikes.

## 5 Summary

Creating an author's dictionary is a labor-intensive task, and information technology can greatly help researchers in this process. It provides efficient tools for making concordance lists, editors for supporting the authoring process, various methods to store available data, and a user interface for accessing the dictionary and attached corpus. We have demonstrated how the digital Mikes dictionary was created using such IT methods and tools.

Our aim was then to take a step further by introducing knowledge-based methods in creating and using the dictionary. In order to achieve this we extended the dictionary with additional knowledge and also connected it to already available knowledge sources like critical annotations and linked open data sets. We created an RDF knowledge base by retrieving information from these sources and linking them to the appropriate dictionary entries and corpus locations. This base contains semantic information imported from linked open data sources, and it also incorporates knowledge from critical annotations by linking them to dictionary and corpus entries. We also developed advanced dictionary and corpus search functions that utilize this knowledge using query expansion to acquire higher quality search results.

There are many possible benefits of developing an extended author's dictionary. For example, it enables access to much more information about an author's vocabulary than before, and provides better awareness of the related linguistic, historical and semantic information. This vast knowledge base can also be a basis for further research. We can conduct new kinds of analyses on the integrated data, text and knowledge base. Finally, the added knowledge could also improve traditional corpus analysis and search methods by supporting corpus normalization and query expansion techniques.

There are many possible ways to take this research even further. Extending and analyzing this knowledge store opens up many possibilities. We are investigating how to extend the semantic knowledge beyond the scope of named entities, and how bibliographical data found in critical annotations can also be incorporated. Another interesting research topic is how to incorporate other kinds knowledge acquisition methods into the system. For example, we are experimenting with controlled natural language interfaces to allow researchers to introduce new knowledge to the RDF store using a natural language online interface.

Most of the software tools developed in the framework of the DHmine project is open source, and they are available in the following GitHub repository: https://github.com/mtwebit/dhmine.

## References

Auer S., Bizer C., Kobilarov G., Lehmann J., Cyganiak R., Ives Z. (2007) DBpedia: A Nucleus for a Web of Open Data. *In: Aberer K. et al. (eds) The Semantic Web. Lecture Notes in Computer Science*, vol 4825. Springer, Berlin, Heidelberg

Bartalesi, V. et al. (2015). Towards a Semantic Network of Dante's Works and Their Contextual Knowledge. In *Digital Scholarship in the Humanities*, pp. 28-35.

Busa, R. (1980). The Annals of Humanities Computing: The Index Thomisticus. In *Computers and the Humanities*. 14(2), pp. 83-90.

Hartmann, R.R.K., James, G. (2001). *Dictionary of Lexicography.* London and New York: Routledge.

Hopp, L. (1966-1988). *Mikes Kelemen összes művei.* L. Hopp (eds.) Budapest: Akademiai.

Kiss, M. (2012). The Digital Mikes-Dictionary. In G. Tüskés et al. (eds.) *Literaturtransfer und Interkultralität im Exil* [...]. Bern: Peter Lang Verlag, pp. 288-297.

Mártonfi, A. (2014). Számítógép és írói szótár – különös tekintettel a készülő József Attila szótárra. In *Magyar Nyelv,* 110(1), pp. 30-46.

Mattaush, J. (1990). Das Autoren-Bedeutungswörterbuch. In Hausmann, Franz Josef et al. (Hrsg.) *Wörterbücher–Dictionaries–Dictionnaires [...], An international encyclopedia of lexicography [...] 2*. Berlin–New York: Walter de Gruyter, , pp. 1549-1562.

Mészáros, T. (2016). Agent-supported knowledge acquisition for digital humanities research. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3936-3941.

Nugues, P. et al. (2016). From Digitization to Knowledge: Resources and Methods for Semantic Processing of Digital Works/Texts. Workshop proceedings. In *Digital Humanities 2016*, July 11, 2016, Krakow, Poland.

J. Soltész, K. et al (1973-1987). *Petőfi-szótár.* Budapest: Akadémiai Kiadó.

Varga P., Mészáros T., Dezsényi C., Dobrowiecki T.P. (2003) An Ontology-Based Information Retrieval System. In: *Chung P.W.H., Hinde C., Ali M. (eds) Developments in Applied Artificial Intelligence. IEA/AIE 2003. Lecture Notes in Computer Science*, vol 2718. Springer, Berlin, Heidelberg

## Acknowledgements

This research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013). Sándor Jan Dobi developed software for the corpus and knowledge store, and he also implemented a programming interface to these components. The initial concordance list and the first version of the web-based dictionary (not shown in this paper) were created by Attila Mártonfi.