

Article

Knowledge Acquisition from Critical Annotations

Tamás Mészáros ^{1,*}  and Margit Kiss ²

¹ Department of Measurement and Information Systems, Budapest University of Technology and Economics, H-1117, Budapest, Hungary

² Institute for Literary Studies, Hungarian Academy of Sciences, H-1118, Budapest, Hungary; kiss.margit@btk.mta.hu

* Correspondence: meszaros@mit.bme.hu; Tel.: +36-1-463-2899

Received: 30 June 2018; Accepted: 17 July 2018; Published: 20 July 2018



Abstract: Critical annotations are important knowledge sources when researching one's oeuvre. They describe literary, historical, cultural, linguistic and other kinds of information written in natural languages. Acquiring knowledge from these notes is a complex task due to the limited natural language understanding capability of computerized tools. The aim of the research was to extract knowledge from existing annotations, and to develop new authoring methods to facilitate the knowledge acquisition. After structural and semantic analysis of critical annotations, authors developed a software tool that transforms existing annotations into a structured form that encodes referral and factual knowledge. Authors also propose a new method for authoring annotations based on controlled natural languages. This method ensures that annotations are semantically processable by computer programs and the authoring process remains simple for non-technical users.

Keywords: critical annotations; knowledge acquisition; controlled natural languages

1. Introduction

Information technology (IT) tools are used in Digital Humanities (DH) projects for storing, searching, retrieving and displaying digitalized texts. They vastly improve the efficiency of various research tasks. As these systems already accumulated huge datasets in digital form, it is a natural step forward to transform these data into knowledge, and utilize knowledge-based tools to provide better IT support for researchers.

A key step in this direction is to develop knowledge acquisition tools that can process the available data and transform it into a knowledge representation. In application fields where data present in an easy-to-process structured form this might appear to be an easy task, but many DH systems store natural language texts from which extracting knowledge is far from trivial.

The ongoing DHmine project at our institutions aims to create software tools to support various digital humanities (DH) research tasks [1]. We are processing the works of Kelemen Mikes, an 18th-century author often called the “Hungarian Goethe” [2]. With its 1.5 million words of text, his oeuvre stands out among Hungarian authors’ oeuvres. The main goal of our work is to create a unified system for storing the digitalized texts of Kelemen Mikes, to attach the already existing critical annotations to them, and to create an author’s dictionary to facilitate the deeper analysis of linguistic, semantic and historical analysis of the oeuvre.

Our previous work concentrated on digitalizing and processing the corpus, and on creating the author’s dictionary [2]. We developed automated software tools that analyze the corpus and encode texts in TEI XML [1].

We also decided to take a step beyond data-centric digitalization: our aim is to introduce knowledge-based methods in the system. We have already explored the possibility of incorporating

additional knowledge to the dictionary [3]. Our research objective is to develop methods for incorporating various kinds of knowledge, and to utilize these in other research tasks later. We focus our efforts on two areas: (1) incorporating external knowledge about named entities from linked open data (LOD) sources; and (2) analyzing and acquiring knowledge stored in critical annotations created by literary researchers. In this paper, we concentrate on the second topic. We discuss what kind of knowledge presents in these research notes, how it could be retrieved from already existing annotations, and how we can facilitate knowledge acquisition by rethinking their authoring process.

There is other research work that introduces knowledge-based methods and tools in the field of Digital Humanities. For example, connecting contextual knowledge to a corpus [4], semantic corpus annotations, or adding sentiment information to lexicons [5]. The so-called “digital geographical edition” [6] uses a geographical apparatus layered over a base critical edition to tackle geographical knowledge. Safaryan et al. aimed to store a critical edition in a graph database to ensure long-term preservation and maintenance [7]. Cheema et al. [8] proposed to organize and visualize annotations such as mind maps, and to attach data from distant reading techniques to them. Processing the critical annotations using knowledge-based techniques or creating a knowledge base from these research notes is a rather unexplored field of research.

2. The Role of Critical Annotations in Literary Research

Critical annotations are a primary knowledge source when researching one’s oeuvre. They are created by literary researchers who have a deep knowledge of the author, the oeuvre and the literary, cultural, geographical and historical era when the works were created.

There are two main purpose of creating such annotations. The first documents text variations from the manuscript to the printed version, also showing alterations in the text. It is mainly used to describe the authoring process of the work. The other type of annotation is the editorial or research note that contains information regarding a text fragment of the work. We concentrated our efforts on this second category: how and what kind of information can be acquired from such annotations using computerized methods.

An annotation refers to a text fragment that is typically a word, a phrase or a sentence. These notes fulfill many different roles such as explaining a given text, analyzing its language usage, giving encyclopedic knowledge or references to related information to provide a better comprehension of the text. They describe a diverse set of knowledge including linguistic, etymological, historical, geographic or other types of factual information about the text piece. As critical editions are typically published in printed form, this knowledge is described in a very concise and abbreviated form.

2.1. Structural and Content Analysis of Critical Annotations

Our basis for analyzing the annotations was the critical edition of Mikes’ letters written from exile in Turkey. The critical edition of these letters was created by Lajos Hopp [9]. He wrote more than 5000 research annotations about various parts of the Mikes oeuvre. The following simple example shows a note about the usage of the word “Constantinapolyban” (excerpt):

“[1.]

0 Constantinapolyban—Előfordul még Constantinápoly, Constancinapoly, Constancinápoly,
Constáncinápoly [...]

The first line refers to the first letter of Kelemen Mikes, and the very beginning of the second line (“0”) specifies that the word can be found in the beginning of that letter. This line 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.

Lajos Hopp attached many kinds of annotations to the texts including literary, historical, cultural, geographical and linguistic notes, and many references to other works. Their length varies from a few lines to several pages. If we analyze their internal structure, we can observe certain similarities

regardless their length. Hopp 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. Figure 1 shows a more detailed example.

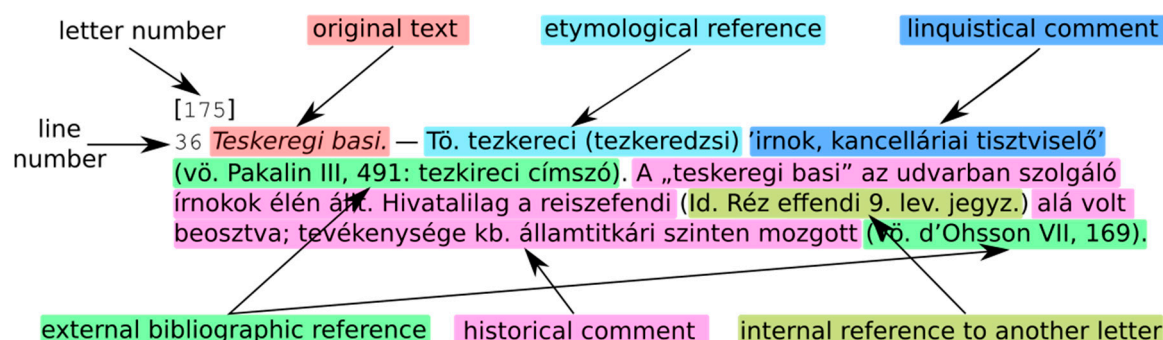


Figure 1. A critical annotation example.

We can observe the typical structure and content in the example. The first line refers to the 175th letter of Kelemen Mikes, and the very beginning of the next line specifies that the cited expression can be found in Line 36 of that letter. The example also shows the original text “Teskeregi basi” and the attached note (after a “—” sign) that describes several types of information about the expression. After the letter and line numbers, the annotation contains a citation from the original text. The next part is a concise explanatory comment that describes the information regarding the text citation. Finally, it may also contain bibliographical references to external works (typically in a very abbreviated form) and internal references to other annotations and text locations.

We also analyzed the content of these annotations. We identified two main types of knowledge: factual (e.g., Constantinapole is a city) and relational (e.g., a note is referring to a certain part of the corpus, or it contains a reference to external entities). We further divided the factual knowledge types into the following categories:

- historical (events, persons, their background information, etc.);
- social (relations and affairs);
- geographic;
- explanatory;
- grammatical, etymological, and filologica;
- literary history;
- cultural history.

As Figure 1 shows, an annotation may belong to several categories at the same time.

Our goal was to process the annotations written by Lajos Hopp, to store them in electronic form, and to extract as much knowledge as possible for further analysis.

2.2. Storing Annotations in a Computer-Processable Form

By observing the previously detailed structure, we created a software tool that automatically transforms these annotations into a structured format including citations, references and the annotations themselves. This transformation is necessary to develop tools that can identify and extract certain parts of the annotations based on their structural or semantic role.

We have chosen the XML TEI standard [10] for storing the annotations because it is very flexible in representing various kinds of data in a self-describing format and it is also easily processable by computerized tools. It is also a common choice in DH projects for various purposes. We used a subset of the standard to encode the content: TEI XML tags are marking notes, texts, corpus citations,

source references, etc. We also used this standard for other parts of the project, e.g., for encoding the dictionary and the corpus.

We developed an open source software tool that automatically performs the XML tagging process (see Supplementary Materials). It is a rule-based tool that analyses the input text according to various tagging rules and transforms the matching text into XML format. Although this tool was created for annotations written by Lajos Hopp, it can serve as a prototype for similar projects.

The following example (Figure 2) shows the result of this automatic transformation.

```
<note type="critical annotation" subtype="N,Tö,B">
  <text>Tö. tezkereci (tezkeredzsi) [...]</text>
  <cit><quote>Teskeregi basi</quote>
    <bibl unit="line" from="36">TL.175</bibl>
  </cit>
</note>
```

Figure 2. An annotation in TEI XML (simplified).

The first line is the XML frame for the entire annotation. It also encodes that the note belongs to linguistic, etymological and historical categories. The next parts store the research note marked with <text> tag and the original text including its citation information between <cit> and </cit> tags. We also ran named entity recognizers using the Mikes Dictionary to find and mark entities (e.g., geographic and person names) in the annotations [2]. These were also enclosed using the <name> tag.

3. Extracting Knowledge from Already Existing Critical Annotations

The DHmine system contains three main datasets: (1) the entire corpus (also encoded in TEI XML); (2) an author's dictionary containing headwords, word forms and various kinds of lexicographic and semantic knowledge (see [3] for more details); and (3) an RDF knowledge store of named entities and certain related concepts extracted from DBpedia [11]. The critical annotations were added to this system as a new database.

There are many connections among these datasets. For example, word forms found in the texts are connected to their dictionary entry; headwords are linked to RDF entries retrieved from DBpedia; etc. Critical annotations also contain reference type of knowledge. Citations are connected to their corpus entries; word forms found in the annotations are linked to their appropriate dictionary headwords; and the recognized named entities are also connected to the entity dataset.

This relational knowledge can be extracted and used to enrich the RDF knowledge store. The Resource Description Framework (RDF) is a very flexible system to represent knowledge. Due to its graph storage it is a natural choice for representing relational knowledge. RDF also provides a very powerful graph query language, SPARQL, that allows querying the database of these relations using complex conditions for graph matches.

The knowledge extraction can be done automatically by analyzing the encoded annotations to create RDF triplets. This is the first type of knowledge acquired from the annotations.

Extracting factual knowledge from the annotations is a much harder task to perform. Although information about named entities and the annotation categories can be retrieved, these only represent a small subset of the available knowledge. Acquiring other kinds of factual knowledge from the annotations is a far more complex task due to the limited natural language understanding capability of computerized tools. To solve this problem, we propose a new way to author critical editions.

4. Authoring Critical Annotations Using Controlled Natural Languages

There are two key requirements to develop a better method for authoring critical annotations: (1) notes should be semantically processable by computer programs; and (2) the authoring process should

not be changed significantly, i.e., it should be simple enough to be used by people not familiar with knowledge editors or other advanced information technology tools. We developed a method based on controlled natural languages that meets these requirements.

4.1. Controlled Natural Languages

To effectively use available NLP techniques, some authors suggested “restrictions” to natural languages. So-called controlled natural language (CNL) [12] resembles the ordinary languages but it has a strict (and restricted) set of language rules, vocabulary and unambiguous meaning, therefore it could be more easily processed by computers. These restrictions allow the successful processing of a controlled natural language by avoiding the problem of disambiguation and uncertain grammar rules, and by explicitly linking the language to the contextual knowledge stored in the system.

Typical application areas of CNLs [13] include writing technical documentation [14], database querying [15], knowledge engineering [16], decision support and machine translation [17]. An important application field for CNLs is knowledge acquisition and editing. For example, there are tools developed for editing ontologies (e.g., Clone [18] or GINO [19]) in a user-friendly way. CLANN (Controlled Language for Annotation) [20] is a more complete system for annotating documents in controlled language and it also includes a domain ontology in RDF.

4.2. Writing Annotations in Controlled Natural Languages

Although CNLs resemble ordinary natural languages they are not exactly the same. The user has to learn the restricted rule set and vocabulary. To speed up this process, we used interface techniques such as predictive text input and annotation templates.

The predictive text input method continuously analyzes the text the user types in; it determines the set of possible sentences based on that input; and it provides suggestions to continue the typing at the cursor position. With this solution, we can attain two goals. The text input is constructed following the language rules, and the user gets immediate help in using the controlled language. This way the user can use the interface from the very beginning (it might be slow for the first time, yet it will be usable), and with time the interface language will become more and more familiar and easy to use. Figure 3 shows the architectural overview of this text input module.

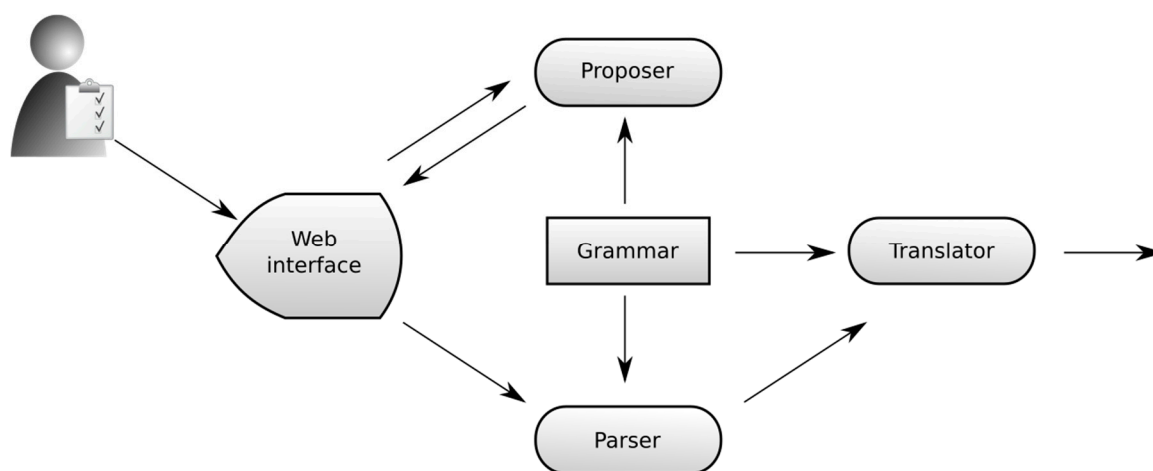


Figure 3. The architecture of the natural language text input module.

Annotation templates help in formulating common types of research notes, for example internal references, factual knowledge regarding named entities, etc. We developed these based on the annotation structure and categories detailed in Section 2.1.

4.3. Selecting the Grammar Representation and Language Parser

There are many possible grammar representations and parsers to choose from. To select an appropriate representation and parser, we analyzed the requirements they should met.

As shown in Figure 3, the language parser is used in two places: generating suggestions and parsing the user's input. The selected grammar representation should hence support both applications. The real-time nature of the user interface requires that the representation should support efficient parsing while creating suggestions.

After analyzing the most common sentences found in already existing critical annotations, we observed that, in most cases, they have a simple grammatical structure describing relations (e.g., a section refers to a place, a person, a bibliographical entity, etc.), definitions (a word refers to a concept defined elsewhere), factual knowledge (e.g., a word is a noun meaning, etc.), etc. In some cases, these are combined into longer sentences (e.g., listing all bibliographical references for a citation).

After analyzing these requirements, we selected a simple grammar type to use in our system: the context-free grammar (CFG). From our point of view, this grammar has some favorable features: its parsers are effective and easy to implement, and its expressiveness proved to be sufficient in several other applications.

We selected an efficient grammar representation, called ordered packed forest (OPF), that is able to serve for both the suggestion generation (Proposer) and the parsing phase (Parser). We implemented a simple chart parser to analyze the user's input. The details of these components are out of the scope of this paper, and can be found in our previous paper describing natural language interface agents [21].

4.4. Constructing the Controlled Natural Language

A key task of using this method is to construct the controlled language rules and the vocabulary. Fortunately, the analysis of already existing critical annotations helps greatly this process.

To simplify the development of the controlled language, we decided to create a composite grammar. We developed separate rule sets and vocabularies for each annotation type (e.g., referral, cultural, historical, linguistic, etc.) based on sample sentences found in the critical annotations written by Lajos Hopp. For example, to create language rules for reference type annotations, we analyzed internal and external references already present in the annotations, and developed a grammar model that suits them.

These sublanguages were easy to create as they were concentrating on a single aspect such as identifying named entities, describing properties of words, etc. We also developed general grammar rules and vocabularies to be used in all categories. These include common abbreviations, word references, numbers, dates including month, names, etc. Some parts of the vocabulary were generated from the problem domain. We extended the grammar with the ability to retrieve entities found in the Knowledge Store or in the Author's Dictionary. To achieve this, we introduced special terminal symbols that specify a rule how these symbols can be instantiated during parsing. For example, the grammar symbol "*Settlement*(KS:type = *settlement*)" specifies that the list of terminals can be retrieved from the Knowledge Store by querying the entities with type "settlement".

4.5. Knowledge Acquisition from Controlled Natural Language Annotations

The key benefit of using controlled natural languages is that they are easily processable by parsers and they can be translated into other representations without ambiguity problems. After constructing a parser tree from the user's input sentence, it can be translated into knowledge representations like RDF triplets or Prolog clauses.

To perform this, we extended the grammar representation with additional data that help to construct the translation. For example, grammar symbols can be extended with their Prolog counterparts and their arguments. A grammar symbol such as *Verb* can be mapped to a Prolog symbol *verb*, and its arguments can also be specified in the form of *verb*(*Word*), where *Word* is a grammar

symbol that will be instantiated during parsing. This way a sentence matching the grammar rule “Word is a Verb:verb(Word)” will be translated into a *verb(Word)* clause in Prolog where the argument (Word) will get its value from the parse tree. RDF triplets (or other knowledge representations) can be constructed in a similar way.

5. Prototype Implementations

We developed a prototype implementation for the ongoing Mikes project.

The annotator component was implemented as a Web application. It allows the user to display, add, edit and delete annotations. In addition to selecting the appropriate categories, it has a predictive text input field for writing controlled language annotations, and a free text form for adding free text notes without any restrictions.

Figures 4 and 5 show the Annotator Tool and the Web interface for editing grammar rules.

The screenshot displays the Annotator Tool interface. On the left, a table lists stored annotations with columns for identifier, language ruleset, and other fields. The 'GALLIPOLI01' annotation is selected. On the right, a detailed editing panel for 'GALLIPOLI01' is shown, including a text field for the citation, a list of language rulesets, and sections for controlled and free text annotations.

Azonosító	Nyelvtanok	Részlet
GALLIPOLI01	[izomorfia, polimorfia]	Gallipoliból
HÁLÁ01	[hivatkozás, szócikk]	hál
INDUL01	[korrektúra]	ind
VENNI01	[szócikk, hivatkozás]	ha
AERO1	[hivatkozás, idézet, szócikk]	áer
MI01	[idézet, hivatkozás]	mi
SIPADOZ01	[szócikk, hivatkozás, idézet]	sip
HOGYNE01	[szócikk, hivatkozás]	hát
HABOK01	[szócikk]	hab
HEGYEK01	[hivatkozás]	az e
ELÉG01	[korrektúra]	elé
HINTÓ01	[]	ha
EGYEPEGYE01	[szócikk, hivatkozás]	egy
RENG01	[szócikk, idézet]	ren
BÓDÍT01	[szócikk, hivatkozás]	a f
GYOMOR01	[hivatkozás]	sze
FARKAS01	[szócikk]	ugy

Kontrollált

Ugyanígy: L3, L122 .
Másutt Gallipoli (L2).

Szabad

Gelibolu régi kikötőváros a Dardanellákban az európai parton, közel az északi szoros bejáratához.

Mentés

Figure 4. The Annotation Tool [22] lists stored annotations (left), their identifier (“Azonosító”), language rulesets (“Nyelvtanok”) and other fields. Selecting an annotation displays a panel (right) that allows editing the citation (“Részlet”), the controlled language (“Kontrollált”) and the free text (“Szabad”) annotation according to the selected grammar rule sets (“Nyelvtanok”).

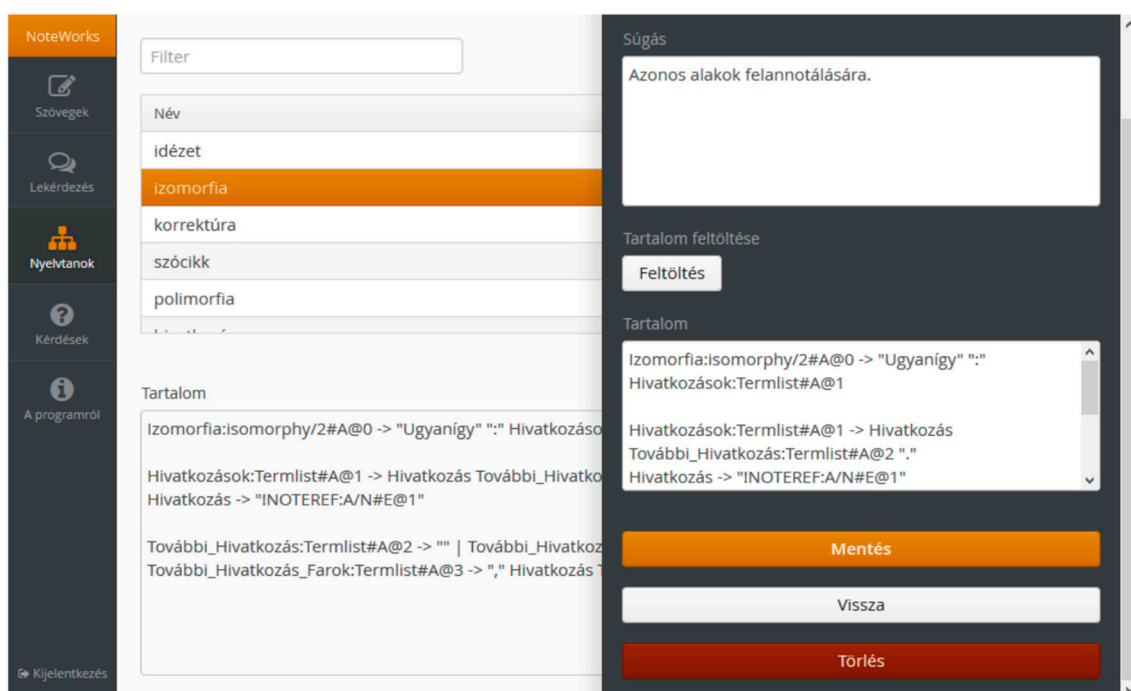


Figure 5. The Grammar Rule Editor [22] is an administrative interface to list and edit grammars.

These tools are supported by back-end services through a REST interface. The services support storing and retrieving data and knowledge pieces and parsing the controlled language sentences.

In addition to these editing and administrative interfaces, we also implemented a Web system to display the letters of Kelemen Mikes, the Mikes Dictionary and the critical annotations. Figure 6 shows a letter where colored boxes mark the locations of critical annotations.

Gallipoliból - 1717. 10. 8brus.



Figure 6. The Web interface showing a letter and annotation markers.

6. Discussion

Critical annotations contain many kinds of information about an author's work. They are written in natural languages, therefore knowledge acquisition from them is a hard problem to solve.

We developed a method and a software tool that helps in extracting knowledge from already existing annotations. It transforms the natural language text into a TEI XML document that encodes several kinds of information (mainly relational knowledge and named entities). This helps to connect the annotations to the proper corpus and dictionary entries, and it is able to retrieve knowledge about named entities found in the texts. Unfortunately, the knowledge acquisition capabilities of this method are rather limited due to the problems of computer-based natural language understanding.

To better facilitate knowledge acquisition from critical annotations, we also developed a new method to author them. This is based on the idea of using controlled natural languages to write annotations. These languages are familiar to the user and they are also easy to process and understand by computer programs. The main benefit of our method is that it does not require the user to use sophisticated knowledge acquisition tools (e.g., Prolog or RDF editors). Non-technical users can create their annotation in an easy-to-use text editor and the system can translate the user's input into the internal knowledge representation. To implement this idea, we analyzed the typical structures found in critical annotations, and selected a grammar class and parsers to implement and to process them. We have also shown the basics of how sentences written in controlled languages can be translated to other knowledge representations.

Our method has several limitations and problems. Constructing the controlled language is not easy. It requires the deep understanding of the problem domain (the annotations) and knowledge about how grammar rules can be created. The controlled language is not the same as the natural language known by the user. It has limited expressiveness; its grammar is not as flexible; and its vocabulary also contains a restricted set of words. These limitations mean that the user has to be trained (or helped) to use the controlled language interface. Although we have shown that predictive text input can help in this matter, it is not enough in many cases. This needs more investigation: What other methods can be used to increase the flexibility of the controlled language interface? How can they be tailored during run-time according to the user's needs?

We developed controlled languages for the Mikes corpus and for the annotations of Lajos Hopp. Other authors probably require other grammar rules and vocabulary. It would be worth investigating what kind of common language constructs can be identified, and how these languages can be created systematically.

Supplementary Materials: Software tools available online at www.github.com/mtwebit/dhmine/.

Author Contributions: Data Curation, M.K.; Investigation, T.M.; Methodology, T.M. and M.K.; Software, T.M.; Validation, M.K.; Writing—Original Draft, T.M. and M.K.; Writing—Review and Editing, T.M. and M.K.

Funding: The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013).

Acknowledgments: The annotation and grammar editor tools were developed by Sándor Jan Dobi as his BSc thesis project. The graphical Web interface for displaying critical annotations was developed by Béla Dienes during his BSc studies in informatics.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mészáros, T. Agent-supported knowledge acquisition for digital humanities research. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Budapest, Hungary, 9–12 October 2016.
2. Kiss, M. The Digital Mikes-Dictionary. In *Literaturtransfer und Interkulturalität im Exil*; Peter Lang Verlag: Bern, Switzerland, 2012; pp. 288–297.
3. Mészáros, T.; Kiss, M. The DHmine dictionary work-flow: Creating a knowledge-based author's dictionary. Euralex: Ljubljana, Slovenia, Unpublished work. 2018.

4. Bartalesi, V.; Meghini, C.; Andriani, P.; Tavoni, M. Towards a semantic network of Dante's works and their contextual knowledge. *Digit. Scholarsh. Hum.* **2016**, *30*, 28–35. [CrossRef]
5. Nugues, P. From Digitization to Knowledge: Resources and Methods for Semantic Processing of Digital Works/Texts. Available online: <http://dh2016.adho.org/abstracts/96> (accessed on 17 July 2018).
6. Lana, M.; Alice, B.; Fabio, C.; Timothy, T. Ontologies and the Cultural Heritage. The Case of GO! Available online: <http://ceur-ws.org/Vol-1595/paper1.pdf> (accessed on 17 July 2018).
7. Safaryan, A.; Kaufmann, S.; Andrews, T. Critical Edition as Graph: The Chronicle of Matthew of Edessa Online. In *Digital Humanities, 2016: Conference Abstracts*; Jagiellonian University: Kraków, Poland; Pedagogical University of Kraków: Kraków, Poland, 2016; pp. 879–880.
8. Cheema, M.; Jänicke, S.; Scheuermann, G. Enhancing Close Reading. In *Proceedings of the Digital Humanities Conference, Krakow, Poland, 11–16 July 2016*; pp. 758–761.
9. Hopp, L. *Mikes Kelemen Összes Művei*; Hopp, L., Ed.; Akadémiai Kiadó: Budapest, Hungary, 1966–1988.
10. Text Encoding Initiative. Available online: <http://www.tei-c.org/> (accessed on 30 June 2018).
11. DBpedia. Available online: <http://www.dbpedia.org/> (accessed on 30 June 2018).
12. Kittredge, R.I. Sublanguages and controlled languages. In *The Oxford Handbook of Computational Linguistics*; Mitkov, R., Ed.; Oxford University Press: New York, NY, USA, 2005.
13. Kuhn, T. A Survey and Classification of Controlled Natural Languages. *Comput. Linguist.* **2013**, *40*, 121–170. [CrossRef]
14. Kamprath, C.; Adolphson, E.; Mitamura, T.; Nyberg, E. Controlled language for multilingual document production: Experience with Caterpillar technical English. In *Proceedings of the Second International Workshop on Controlled Language Applications, Pittsburgh, PA, USA, 21–22 May 1998*.
15. Androutsopoulos, I. Natural Language Interfaces to Databases—An Introduction. *J. Nat. Lang. Eng.* **1995**, *1*, 29–81.
16. Kuhn, T. Controlled English for Knowledge Representation. Available online: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.9617&rep=rep1&type=pdf> (accessed on 17 July 2018).
17. Mitamura, T. Controlled language for multilingual machine translation. *Mach. Transl. Summit* **1999**, *VII*, 46–52.
18. Funk, A.; Tablan, V.; Bontcheva, K.; Cunningham, H.; Davis, B.; Handschuh, S. Clone: Controlled language for ontology editing. In *Proceedings of the International the Semantic Web and, Asian Conference on Asian Semantic Web Conference, Busan, Korea, 11–15 November 2007*; pp. 142–155.
19. Bernstein, A.; Kaufmann, E. GINO—A guided input natural language ontology editor. In *International Semantic Web Conference*; Springer: Berlin, Heidelberg, 2006; pp. 144–157.
20. Davis, B.; Dantuluri, P.; Dragan, L.; Handschuh, S.; Cunningham, H. On designing controlled natural languages for semantic annotation. In *Proceedings of the International Workshop on Controlled Natural Language, Marettimo Island, Italy, 8–10 June 2009*; pp. 187–205.
21. Mészáros, T.; Dobrowiecki, T. Controlled natural languages for interface agents. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems, Budapest, Hungary, 10–15 May 2009*; pp. 1173–1174.
22. Dobi, J.S. Kontrollált Természetes Nyelvű Szövegannotáló-Rendszer. Bachelor's Thesis, Budapest University of Technology and Economics, Budapest, Hungary, 2015.

