150 Years after Dillmann's Lexicon:
Perspectives and Challenges of Gəʿəz Lexicography

Supplement to Aethiopica. International Journal of Ethiopian and Eritrean Studies
5

Edited in the Asien-Afrika-Institut
Abteilung für Afrikanistik und Äthiopistik
Hiob Ludolf Zentrum für Äthiopistik
der Universität Hamburg

Series Editor: Alessandro Bausi
in cooperation with Bairu Tafla, Ulrich Braukämper,
Ludwig Gerhardt, Hilke Meyer-Bahlburg

2016
Harrassowitz Verlag · Wiesbaden

150 Years after Dillmann's Lexicon: Perspectives and Challenges of Gəʿəz Studies

Edited by
Alessandro Bausi
with assistance from
Eugenia Sokolinski

2016
Harrassowitz Verlag · Wiesbaden

The publication of this volume was supported by the European Union Seventh Framework Programme IDEAS (FP7/2007-2013) ERC grant agreement 338756 (TraCES).

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the internet at http://dnb.dnb.de.

For further information about our publishing program consult our website http://www.harrassowitz-verlag.de

Printed on permanent/durable paper.
Typesetting, copy editing, index: Eugenia Sokolinski (Hamburg).
Printing and binding: Memminger MedienCentrum AG
Printed in Germany

ISSN 2196-7180
ISBN 978-3-447-10783-9

Table of Contents

Bringing Gəʿəz into the digital era: computational tools for processing Classical Ethiopic*

CRISTINA VERTAN, Universität Hamburg

§ 1. Introduction

During the past ten years, research paradigms in traditional fields of philology have changed significantly with the increased use of methods from computer science and information technology. Regarded at the beginning in the first place as a way to preserve cultural heritage and catalogue existing objects, the Digital Humanities (DH) have since evolved into an independent field of research, which equips scholars with tools for quantitative and qualitative analysis of data, for visualisation, and interpretation of the results. The strength of DH is in the ability to process big amounts of heterogeneous, multilingual, and geographically discrete data and to provide an easy access for the users. The final interpretation remains, and should remain, in the hands of the scholar.

Computational methods range from supplying objects (or their surrogates) with descriptive metadata to linguistic annotation of text corpora, data modelling in semantic databases, critically and diplomatically editing texts, linking images to texts and GIS (geographical information system) data, application of machine learning techniques, and graphical visualizations that facilitates achieving tangible research results.

The advances of DH have not been equally used across the entire array of academic disciplines. The first field of research in the humanities to have profited from information technology was language studies when in the 1970s computational linguistics emerged at the confluence of linguistics and computer science. Forty years later, computational linguistics, with branches such as language engineering and natural language processing, can boast an impressive number of language resources and tools. The discovery of statistical corpus-based methods in the 1990s brought about a rapid growth in the subfield of corpus linguistics, large corpora for training purposes having been collected. Yet, the main beneficiaries of these developments have been the modern languages that have a large number of speakers and are widely used in international communication, and are thus politically and economically important (primarly European languages, with English in the foreground,

* The research leading to these results has received funding from the European Research Council under the EU Seventh Framework Programme, grant agreement no. 322849.

but also German, Spanish, French, Italian, and some others; to a far lesser degree, such oriental languages as Chinese and Standard Arabic).

As for historical languages, it is only relatively recently that they have started to be touched upon by computational linguistics, and few can boast significant digital text corpora, let alone annotated ones. This has a series of methodological consequences for research, as most algorithms and tools that have been developed so far take into account the features shared by Indo-European languages and/or using alphabetic (primarily Latin) scripts. Among these consequences we can list:

— clear set of rules for identifying and segmenting sentences and words (tokenization) have emerged that are shared by most Indo-European languages;
— part of speech sets (noun, verb, etc.) as well as their features (gender, number, etc.) accepted in computational linguistics are the ones used to describe Indo-European languages;
— the predefined morphological paradigm implies that word generation is based on inflection derivation and compounding;
— lexicon units are seen as lemmata;
— syntax is described by a set of well documented rules.

Most little spoken and in particular historical languages suffer from lack of adequate digital resources and tools which would model their particularities.

Until recently the approach followed by DH when dealing with non-standard data, i.e. data which do not fit exactly to existent algorithms, was to force this data to be accepted as input for already implemented processes. This would mean, among other things, dropping-down linguistic features which did not occur in standard cases; performing just a shallow annotation; pre-processing data (for example splitting compounds, agglutinating part of speech in a common ancestor denomination, eliminating long time dependencies), etc.

This kind of approach can help in language engineering applications where data mining is used just to grasp the general purpose of a document. However, it is useless if the goal of data processing is a scientific quantitative or qualitative analysis of the respective language.

Classical Ethiopic (Gəʿəz) is a prototype example of a language that is of extreme importance for assessing and interpreting the Early Christian period but at the same time lacks electronic resources. Graphical and linguistic particularities make the use of existent tools impossible.

Gəʿəz texts are preserved in manuscripts, some of them hardly accessible; some have been edited in publications from different periods. In both cases, digitization and further work on digitized material seem to be the only way to ensure not only preservation but also a collaborative research, comparisons across editions, and a diachronic language analysis. A linguistically annotated corpus of Gəʿəz linked to a machine-readable lexicon is the basic premise for any computer-aided research framework. Until now this kind of resources have been completely missing. The project *TraCES*: *From Translation to Creation*: *Changes in Ethiopic Style and Lexicon from Late Antiquity to the Middle Ages*[1] aims at filling this gap by providing the first integrated digital framework for collection, annotation and visualization of Ethiopic texts. As outcome of the project, a diachronic Gəʿəz corpus annotated at four levels (linguistic, text structure, important proper names and edition) will be made available for advanced intelligent search and visualization of search results.

The development of a digital framework faces a series of challenges due to the particularities of Gəʿəz. Our general approach is to develop tools and resources which take into consideration the language particularities and not to constraint the texts into predefined models. Thus modelling the data and software specification and design was a considerable part of the work. This paper focuses on the development of a multi-level tagging tool that allows a very fine-grain linguistic annotation of Gəʿəz texts. In addition, the tool can be easily adapted to other languages following similar paradigms as Gəʿəz.

The paper is organized as follows: in § 2, it gives a brief overview of challenges resulting from language specificities and describe the data model. In § 3, the functionality of the annotation tool and its integration in the general digital framework is presented. § 4 concludes with an overview of further developments to be carried on.

§ 2. Data and annotation model

A computer assisted diachronic analysis of a language should be conducted on a deep linguistically annotated corpus. For that, the data model must first be defined and applied. In the case of the *TraCES* project, four steps preceded the choice of data and annotation model:

(1) at the linguistic annotation level: define the set of part of speech and their features, which are able to describe any morphological phenomena in Gəʿəz;[2]

(2) identify the smallest units to which morphological annotation may be associated;

(3) define the other annotation levels (text structure, named entities and edid tion) and the features to be annotated there;

(4) identify the minimal annotation units for the three additional levels.

1 <https://www.traces.uni-hamburg.de/>, last accessed 25 November 2016.

2 On the set defined, see the contribution of Susanne Hummel and Wolfgang Dickhut in this volume.

The following basic terminology has been applied within the project:

— *graphic unit*: a sequence of characters in Gəʿəz script (*fidal*) or their transliteration between two empty spaces. The *fidal* sequence within a sentence usually ends with the word divider ' ፡';

— *token:* the smallest sequence of characters to which a morphological annotation can be attached.

One graphic unit may consist of more than one token. For example, the graphic unit ወቤቱ ፡ (*wabetu*) corresponds to three tokens (*wa-bet-u*, Conjunction-Noun-Pronoun). It is obvious from this example that the latter token boundary must be drawn inside one *fidal* syllabic grapheme (ቱ, *tu*). Thus, because of the syllabic character of the script, visual splitting of tokens (marked by a hyphen '-') and the morphological annotation is only possible on the transliteration.

Yet, the preservation of the original *fidal* text is also important for philological reasons. Therefore the original *fidal* is preserved after the transliteration, and not only preserved, but is also synchronized, so that any changes (corrections, deletions, insertions) in transliteration are automatically reflected in *fidal*, and vice versa. Such synchronization feature between original script and transliteration is not offered at the moment by any available annotation software.

Named entity is the term used to describe proper names (personal names, toponyms, book titles) but also dates, abbreviations, currency units, etc. A named entity can correspond to one or several tokens and cross the graphic unit boundaries. For example, the phrase ወኢየሱስ ፡ ክርስቶስ ፡, which consists of two graphic units and three tokens (*wa-ʾiyasus krəstos*) contains the named entity (personal name) *ʾiyasus krəstos* (Jesus Christ).

Text structure elements (chapter, sentence, verse, etc.) are associated with sequences of graphic units. Edition elements[3] (line, page, and paragraph breaks, occasionally brackets marking reconstructed or missing characters or character sequences) are associated with single *fidal* characters.

The challenge has been to design a data and annotation model that allows users to decide dynamically at which level they annotate. The chosen tree data structure is illustrated by the example in fig. 1.

— The graphic unit is at the root of the tree. The word divider ' ፡' is treated here as an editorial mark.

— The root is linked with descending nodes corresponding to the *fidal* characters.

— Each *fidal* node is descended by transliteration nodes. As a rule, each glyph is linked to two transliteration nodes (consonant + vowel), but occasion-

3 The annotation at the 'edition' level is kept minimal for the moment, only few selected elements are marked.

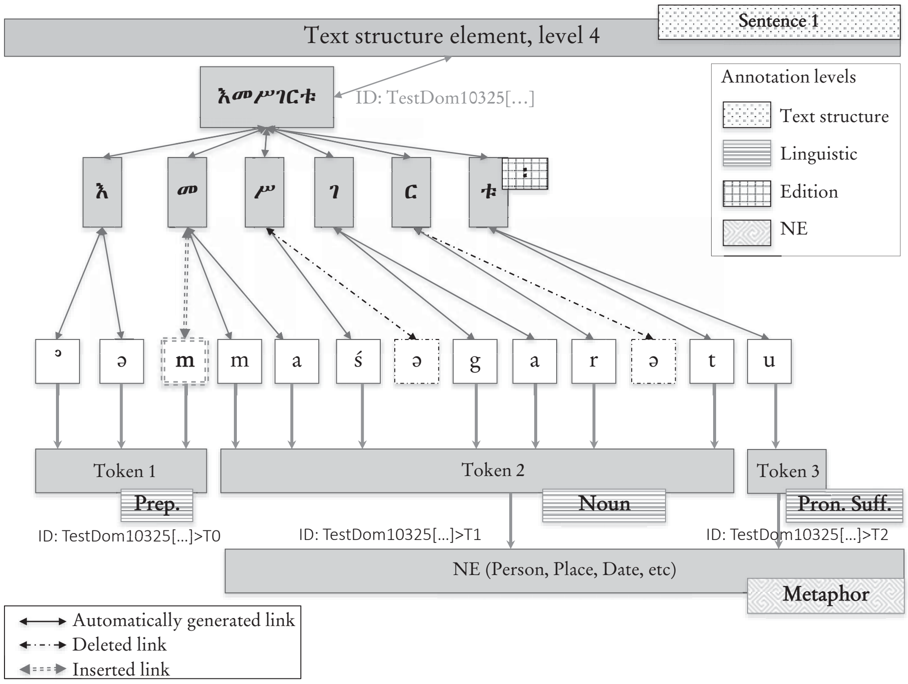

Fig. 1. Data and annotation model.

ally also to one (only consonant, when the sixth order vowel is reduced to zero), or three (when the consonant is geminated).

— The token in this data structure corresponds to a span over a set of transliteration nodes.
— A named entity spans over several tokens, and a text structure division spans over several graphic unit nodes.

All elements—graphic units, tokens, named entities and text structure divisions—receive unique IDs, all descending from the text ID. Thus, the graphic unit ID consists of the text ID (described in the metadata of the text) followed by a universally unique identifier (a randomly generated alphanumeric sequence). The token ID consists of the graphic unit ID followed by the number of the token within the graphic unit. For example,

Text:	*Testamentum Domini* (TestDom)	
Graphic unit:	**እመሥገርቱ**	TestDom10325caf1d4241eb920de40c983c95b0
Tokens:	*ʾəm*	TestDom10325caf1d4241eb920de40c983c95b0>T0
	maśgart	TestDom10325caf1d4241eb920de40c983c95b0>T1
	u	TestDom10325caf1d4241eb920de40c983c95b0>T2

The tree data model allows for the dynamic insertion and deletion of nodes at each level. It offers the major advantage of separating the graphical representation of a graphic unit from its internal organization. The *fidal* and transliteration characters are just labels associated with each node element, thus they can be changed without losing or damaging other information associated to the nodes (i.e. the annotation). This guarantees full flexibility in performing correction and changes in transliteration and annotation.

§ 2.1. *Encoding format*

The TEI XML format[4] has become the most widely accepted format for encoding texts data in digital humanities. TEI standard is particularly powerful for encoding diplomatic transcriptions of texts, as well as critical editions, and describing text carriers. For scholars working in digital text editing and manuscript cataloguing, it has allowed for a considerable degree of interoperability. Yet, the TEI specifications are exceedingly general, may be hard to parse, and building applications manipulating links between TEI elements is often difficult.

Considering the selected data model, it has been decided to encode the data in JSON format.[5] The textual data and certain levels of annotation can then eventually be converted and exported to TEI XML for the purposes of data exchange with other projects.

The data is thus stored in JSON objects, respectively collections of JSON objects. JSON documents can be easily parsed and are in plain text format.

(1) The first JSON collection is represented by the set of JSON objects encoding the internal structure of each graphic unit. A graphic unit JSON object contains a collection of JSON objects encoding the information on each *fidal* letter. Each *fidal* letter JSON object is a collection of JSON objects encoding information on corresponding transliteration letters. Each JSON transliteration letter object contains a pointer to the ID of the token to which it belongs. Initially (before user tokenization), each graphic unit is automatically a token. A graphic unit JSON object includes also pointers to textual divisions to which it belongs. (2) The second JSON collection is represented by the set of JSON objects encoding the tokens. It records the morphological annotation as well as pointers to the named entities to which it belongs. (3) The third JSON collection is represented by the named entities. (4) The fourth JSON Collection is represented by the text structure divisions.

4 Text Encoding Initiative, <http://www.tei-c.org/Guidelines/P5/>.
5 Javascript Object Notation, <http://www.w3schools.com/js/js_json_intro.asp>.

§ 3. System functionality

Annotation software which is currently available does not allow manipulation of such complex structures as described in § 2. Thus, it was necessary to design and implement a tailored annotation tool that not only is able to manage our data structure but also facilitates a rapid and consistent annotation process. The GeTa tool (see fig. 3) has been therefore developed.

The big number of parts of speech and a great degree of variance of features makes a fully automatic annotation through machine learning algorithms impossible. Besides, there is no training corpus that would be adequate in size to attempt machine learning. Therefore, the tool has been designed for manual annotation.

Yet, a certain degree of automation is possible, and it could both speed up the annotation process and increase the degree of consistency across texts. Therefore, a semi-automatic controlled annotation procedure has been adopted. 'Semi-automatic' means the possibility of batch annotation when the user decided that the linguistic annotation could be applied to all instances of a segment within the text. 'Controlled' implies that the user always has the possibility to distinguish batch annotated from manually annotated elements and implement corrections when necessary.

In addition to that, the GeTa annotation tool offers the following features:

— Transliteration. Texts in original script are accompanied by a synchronized transliteration (produced initially automatically following the transliteration convention represented in fig. 2). Each *fidal* symbol is transliterated as a syllable (consonant+vowel) except for the word final position, where the sixth order is rendered as zero. Correct gemination and sixth order disambiguation can only be implemented, in most cases, once the linguistic analysis has been carried out, that is in the course or after the linguistic annotation process.[6] Therefore the possibility of manual correction of transcription during the linguistic annotation is essential.[7]

— Linguistic annotation. The annotation scheme contains 33 part of speech tags, most of them further specified by additional features. Many existing tools[8] impose a single field for linguistic annotation, resulting in the necessity of creating (prior to the annotation) long strings encoding all possible combination of features (for example for the noun, NCom-FemN-

6 On the complexity of sixth order transliteration see also the contribution by Maria Bulakh in this volume.

7 Among the many existing annotation tools, only few support this. One of them is CorA, <https://www.linguistics.ruhr-uni-bochum.de/comphist/resources/cora/index.html>, which, however, does not support synchronization with other visualization levels should one want to keep the *fidal* script.

8 Including the aforementioned CorA tool.

ከ	*ka*	ሀ	*ha*	ቈ	*qʷa*
ወ	*wa*	ለ	*la*	ኈ	*ḫʷa*
ዐ	*ʿa*	ሐ	*ḥa*	ኰ	*kʷa*
ዘ	*za*	መ	*ma*	ጐ	*gʷa*
የ	*ya*	ሠ	*śa*	ቨ	*va*
ደ	*da*	ረ	*ra*	ቸ	*ča*
ገ	*ga*	ሰ	*sa*	ጀ	*ǧa*
ጠ	*ṭa*	ቀ	*qa*	ጨ	*č̣a*
ጰ	*p̣a*	በ	*ba*	ቐ	*q̱a*
ጸ	*ṣa*	ተ	*ta*	ኸ	*ḵa*
ፀ	*ḍa*	ኀ	*ḫa*	ሸ	*ša*
ፈ	*fa*	ነ	*na*	ኘ	*ña*
ፐ	*pa*	አ	*ʾa*	ዠ	*ža*

Vowel orders

1	2	3	4	5	6	7
a	*u*	*i*	*ā*	*e*	*ə*	*o*

Fig. 2. *TraCES* transliteration convention (only Unicode symbols have been used).

Sg-Nom-Abs, NCom-FemN-Sg-Nom-Cons, NCom-FemN-Sg-Acc-Abs, NCom-FemP-PlEx-Nom-Abs, etc. etc. etc.) In our case, this would mean several hundred possible combinations, with each chain being up to 20 digits long. Not only of little practicality in annotation, this also makes the targeted search for an external person very difficult, as one has to remember not only the name of the specific feature but also the sequence in which the features are coded in the strings. Thus, the requirement and in the same time the challenge for the GeTa annotation tool was to provide the user with specialized annotation masks for each part of speech.

— Other levels of annotation. In the *TraCES* corpus, the annotation is done at four levels: text structure, named entities, edition, and linguistics. Thus, a multi-level solution for the annotation tool was required.[9]

The system we designed (GeTa annotation tool) performs the following controlled automatic operations:

— Tokenization. Each automatically tokenized graphic unit is italicized. In order to minimize the human checks we assume that the assignment of a morphological annotation to a token is implicitly a validation of the tokenization.

— Correction of transliteration. Corrections can be carried out both prior to tokenization and annotation and during or after the annotation process.

9 Among the existing tools, WebAnno <https://webanno.github.io/webanno/> supports multi-level annotation. However, in WebAnno no corrections can be carried out in the text once the annotation process has started.

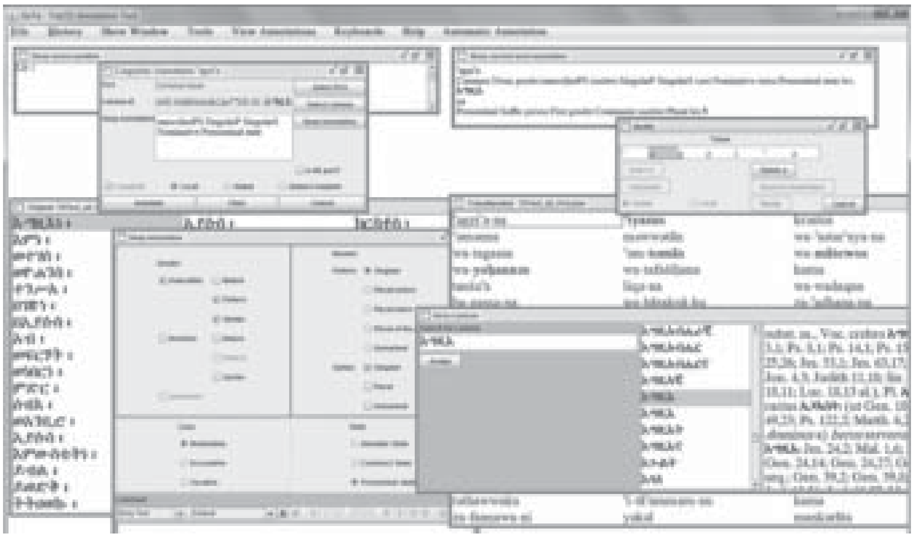

Fig. 3. The *GeTa* annotation tool. Most of the tokens in the transliteration screen are in blue showing the completed annotation process.

The disambiguation of the sixth order and the introduction of consonantal gemination are possible even after a morphological annotation has been assigned to a token. Other corrections (replace a *fidal* letter, insert or delete a letter) implies the deletion of the annotation if performed after the mark up, as they would mean a change in the word meaning and subsequently its morphological description.

— Assignment of linguistic features. As mentioned above, the tool supports manual annotation with a possibility of semi-automated controlled batch annotation. Non-annotated transliterated text appears in black regular font. Manually performed incomplete annotations are in bold, completely annotated units appear in blue. Batch annotated items appear in red. In some cases, when automatically performed mark up does not require additional control (for example punctuation signs), the user can check the 'global complete' box in the annotation mask before performing the batch annotation. In this case, the items will turn to blue.

Additional functionality of the GeTa tool as of November 2016 includes:

— a search mask allowing basic browsing of the text. Search is possible both in *fidal* script and in transliteration. User can define the size of the context to be shown. Search results are highlighted and it is possible to jump directly to the corresponding position in the text. Search options include part of speech, start position for the search as well as search level (token or graphic unit);

— a statistics module allowing graphical visualization of the annotation progress;
— a graphic unit visualization function, which allows the graphical representation of each graphic unit;
— an automatic management of line breaks numbering. The user only needs to insert (or delete) a line break, the numbering is performed automatically;
— a commentary function at each level (graphic unit, token, edition, text structure). Additionally, graphic units can be highlighted in different colours.

While the dictionary tool is still under development, each token can already be linked to a lemma extracted from the digitized version of the *Lexicon* by August Dillmann. In its final form the GeTa tool will have a direct interface to a fully fledged electronic dictionary of Gəʿəz. For the moment, a minimal version has been implemented by extracting automatically the lemmata from the lexicon and assigning unique IDs to each lemma. A basic general user interface ensures the browsing of the lexicon. Entries are ordered lexicographically, and users can see the translations and explanations provided by Dillmann, preserved with their original formatting (italic and superscript) maintained.[10]

§ 4. Conclusions and further work

GeTa is an innovative powerful annotation tool that allows multi-level annotation of texts, tailored particularly for the needs of the Gəʿəz language. It provides for a possibility of correcting the text during the annotation process. Besides the linguistic annotation level (for a very fine-grained linguistic annotation), it supports three more annotation levels. The annotation is possible in semi-automatic controlled modus which ensures consistency and prevents errors. The tool is easy to use and fast, and has a user-friendly interface. It is implemented in Java with data saved in JSON.[11]

10 See C. Vertan, 'Towards a digital lexicon of Ethiopic: the TraCES experience', in A. Bausi, A. Gori, D. Nosnitsin, and E. Sokolinski, eds, *Essays in Ethiopian Manuscript Studies. Proceedings of the International Conference Manuscripts and Texts, Languages and Contexts: the Transmission of Knowledge in the Horn of Africa, Hamburg, 17–19 July 2014*, Supplement to Aethiopica, 4 (Wiesbaden: Harrassowitz Verlag, 2015), 13–15.

11 The tool runs safely on multiple platforms. It has been tested for Windows 7, Windows Vista, and Linux operating systems.

Adaptations of the tool to other languages and scripts are possible. As a first attempt, the tool has been adapted to read the texts of Gəʿəz inscriptions, in unvocalized Gəʿəz script but also in Epigraphic South Arabian script. In particular, the South Arabian adaptation demonstrated the flexibility that the system was capable of coping with: (1) a different script; (2) a different transliteration rule set; and (3) a different writing direction (right to left). This experiment has shown that further extensions (for example for Syriac, Arabic, Hebrew) are possible. However, in this case an adaptation of the linguistic features to be annotated should be considered. For the moment, we are conducting some experiments concerning the adaptability of morphological annotation to Amharic.

Currently, the only form of automatic operation is the replication of a user's manual operation. Once a critical mass of annotations is performed, it shall be possible to implement a module which extracts 'automatic paths', with a 'path' standing for one graphic unit–transliteration–tokenization–annotation chain. The underlying principle is that each graphic unit can correspond to different transliterations, each transliteration can correspond to several tokenization possibilities, and for each tokenization, at least one morphological annotation is available. Recording all possible paths for each graphic unit will lead to a further automatization step. If only one path is available then it is applied by the tool, if multiple paths are available, the user has to perform the disambiguation. This automatization step will increase considerably the annotation speed without losing control or risking creation of additional errors.

The final purpose of the annotation process is naturally the possibility to perform qualitative and quantitative corpus analysis. While it does contain a basic search module, the GeTa tool is primary designed for annotation, and not for work with text corpora. To respond to this need, a plug-in is being developed to export the GeTa annotated data to the ANNIS Internet tool for corpus visualization and search.[12] Subsequently, further conversion filters to export the data to TEI XML format shall be implemented.

12 See <http://corpus-tools.org/annis/>. The conversion filters are currently being developed by Stephan Druskat (Berlin).