

## *Arethusa*<sup>1</sup>

*Robert Lichtensteiner*

**A**rethusa ist eine Online-Plattform, die zum Erlernen und Erforschen der lateinischen, griechischen und anderer Sprachen wertvolle Beiträge liefert. Woraus sich diese Plattform zusammensetzt, wie sie funktioniert und welche Hintergründe der Entwicklung zugrunde liegen, sei im Folgenden dargelegt.

### *Hinc omne principium, huc refer exitum*

Im Mittelalter erforderte das Sammeln und Tradieren von Handschriften antiker Texte sehr viel Mühe. Zuerst bedurfte es eines Exemplars als Vorlage, dann eines oder mehrerer fleißiger Kopisten – Mönche –, die den Text Buchstaben für Buchstaben abschrieben, und zuletzt einen sicheren Aufbewahrungsort. Fast alle antiken Texte, die wir heute besitzen, kamen auf diesem Wege zu uns. Zahlreiche Manuskripte schafften es jedoch nicht, in eine derartige „Sammlung“ aufgenommen zu werden, und sind auf ewig verloren. Die Erfindung des Buchdrucks in der frühen Neuzeit erleichterte zumindest das Tradieren der Texte, die Suche nach Handschriften, in denen antike Autoren festgehalten wurden, allerdings nicht.

In der heutigen, computergestützten Zeit erscheint das Sammeln von (den „übriggebliebenen“) antiken Texten unproblematisch. Zahlreiche Online-Projekte bieten für Interessierte einen einfachen Zugang zu lateinischen und griechischen Texten. Zwar sind noch lange nicht alle Autoren und ihre Werke digitalisiert, auch unterscheiden sich die einzelnen Sammlungen hinsichtlich ihrer Qualität (Textkritik!), dennoch erweckt es den Eindruck, als sei das antike Schriftgut in der Welt der neuen Medien angekommen.

### *Docendo discimus*

Lateinlehrer/innen kostet es heute kein bisschen Mühe im Internet einen Text zu finden, der nicht in gebundener Form im eigenen Bücherregal steht. Oftmals jedoch ist die Arbeit damit nicht getan: Was, wenn ich als Lehrer/in mit meinen Schüler/innen eine Ausschnitt aus einem Originaltext lesen möchte, bei dem ein besonderes Grammatikphänomen in einer hohen Dichte vorkommt? Was, wenn ich beispielsweise mit einer Klasse einen Pliniustext behandle und dabei das Gerundium wiederholen möchte? Muss ich dafür alle Briefe durchlesen? Oder wenn ich dasselbe bei Livius finden will?

Vor diesem Problem standen auch wir. Als mein Kollege Gernot Höflechner und ich einmal mehr in den Sommerferien Studierende auf ihre Latinum-Prüfung vorbereiteten, kämpften wir wie immer gegen die Zeit: 60 Stunden mussten genügen, um die gesamte lateinische Grammatik zu lehren und auf 3000 zu übersetzende Wörter vorzubereiten. Es war beinahe unmöglich, täglich 38 (so viele Teilnehmer saßen im Kurs) Hausübungen zu kontrollieren, sich auf die nächste Einheit vorzubereiten und gleichzeitig noch individuelle Betreuung zu gewährleisten. Wir dachten zuerst an ein System, das automatisch Übungen kontrolliert, waren uns auch bewusst, dass es derartige Lösungen bereits gibt. Das Problem bei diesen

---

<sup>1</sup> Dieser Artikel steht auf unserer homepage [www.latein-forum.tsn.at](http://www.latein-forum.tsn.at) mit den farbigen Abbildungen zur Verfügung.

vorgefertigten Programmen liegt jedoch daran, dass sie nicht individuell gestaltet werden können. Was nützt eine Übung, die aus Caesar entnommen ist, wenn die Studierenden hauptsächlich mit Hygin arbeiten? Es wäre wohl auch nicht allzu schwierig gewesen, selbst ein Übungsprogramm herzustellen. Daran musste allerdings ausgesetzt werden, dass die Übungen alle manuell ausgewählt werden müssen.

In unserer Kühnheit phantasierten wir über ein Programm, das lateinische Texte automatisch hinsichtlich ihrer Syntax und Morphologie analysiert, sodass schnell und problemlos nach einzelnen Phänomenen gesucht werden kann; ein Programm, das aus eingegebenen Texten automatisch verschiedenste Übungsarten kreiert, die Lösungen zu all diesen Übungen kennt und dem Nutzer individuelle Hilfestellung bietet; ein Programm, das letztendlich den Umgang mit der lateinischen Sprache in Schule und Wissenschaft revolutionieren sollte. In diesem Augenblick höchsten Übermutes wurde das Latin-Language-Toolkit-Projekt (LLT-Projekt) geboren.

### **Prima ratio**

Bevor wir jedoch diese individuellen und dynamischen Übungen in Angriff nehmen konnten, mussten wir dem Computer Latein „beibringen“ – und uns selbst programmieren. Bis dahin hatten wir kaum Kontakt zu dieser Materie, weshalb wir in dieser Phase auch sehr dankbar für die Hilfe unseres Freundes Christof Sirk waren, der als erfahrener Softwareentwickler in unser Projekt einstieg und uns half, unsere Fähigkeiten zu entwickeln und Probleme technischer Natur zu lösen.

So begann also die praktische Arbeit. Wir versuchten einen sogenannten Parser (von lat. *pars* = Teil der Rede) zu programmieren, der einen lateinischen Satz ohne Zutun eines Menschen syntaktisch und morphologisch analysieren sollte. Ohne es anfangs zu wissen, stolperen wir in das Gebiet der Computerlinguistik und des natural language processings (NLP). Tatsächlich gab es auch schon vor uns Versuche, die lateinische Sprache für Computer verständlich zu machen. Diese näherten sich dem Problem auf statistischen Wegen, die sich – vereinfacht erklärt – auf die Wortstellung im Satz konzentrieren. Dabei sollte man sich jedoch vor Augen halten, dass sich diese bestehenden Werkzeuge dadurch massiv einschränken, dass sie eben vom Autor, der Zeit und der Textgattung abhängig sind, was letztendlich zur Minderung der Effektivität führt.

Die LLT Methode dagegen ist keine statistische. Sie versucht, sich dem Problem „objektorientierter“ und „kontextgebundener“ zu nähern. Mit anderen Worten, der LLT-Parser imitiert die menschlichen Gedankengänge beim Übersetzen. Während der morphologischen Analyse werden den Formen „Funktionen“ zugeteilt, z.B. ein finites Verb trägt die Funktion „Prädikat“, ein Substantiv im Nominativ „Subjekt“ usw. Bei der syntaktischen Analyse wird in einer bestimmten Reihenfolge gezielt nach diesen Funktionen gefragt und so ein Ergebnis generiert. Dieses Sprachverständnis ermöglicht die Unabhängigkeit von den Faktoren Textgattung, Autor, Stil und Epoche.

Das Programm analysiert beispielsweise folgenden Satz aus Caesar (Bellum Gallicum 2,9) fehlerfrei:

*Ubi neutri transeundi initium faciunt, secundiore equitum proelio nostris Caesar suos in castra reduxit.*

(Üs.: Als keines von beiden [= Heere] begann [erg.: den Sumpf] zu überqueren, führte Caesar nach einer für uns günstigen Reiterschlacht seine Leute ins Lager zurück.)

Es erkennt alle syntaktischen Beziehungen in diesem Beispiel problemlos: *reduxit* wird als Prädikat des übergeordneten Satzes betrachtet, *Caesar* als dessen Subjekt, *suos* als dazugehöriges Objekt usw.

Auf dieser Grundlage konnten nun erstmals aus syntaktisch nicht allzu komplexen Sätzen dynamisch Übungen generiert werden. Aus

*Marcus rosam videt*

wurde

*Marcus rosam \_\_\_\_\_ (videre),*

nachdem dem Computer mitgeteilt worden war, dass eine Lückentextübung für das Prädikat erstellt werden soll. Zusätzlich analysierte das Programm auch die Antwort der User. Gibt ein User *vident* als Lösung an, so erscheint der Hinweis auf den Numerus des Subjekts zu achten.

Es zeigte sich jedoch recht bald – und wir waren uns dieser Herausforderung stets bewusst –, dass vor allem umfassende und (für Computer) vertrackte Sätze, bei denen die zahlreichen morphologisch möglichen Formen eines Wortes eine schnelle und einfache Lösung verweigern, nicht immer exakt analysiert werden können.

Die manuelle Suche nach solchen syntaktischen und morphologischen Fehlern wurde sehr bald zu mühsam und zeitaufwendig, wobei wir uns jedoch nicht des Eindrucks erwehren konnten, dass sich unsere Lateinkenntnisse (nach vier bzw. sechs Jahren Latein an der Schule und 12 Semestern auf der Uni!) aufgrund der intensiven Beschäftigung mit der Materie verbesserten. Dies war eine große Erkenntnis und eröffnete uns eine neue Sprachbetrachtungsmöglichkeit, die wir bei dieser Arbeit so nicht erwartet hatten. Wenn Latein Lernende durch das Korrigieren von syntaktischen Analysen, die ein Computer automatisiert erstellt hat, bzw. durch das eigenhändige Zerlegen lateinischer Sätze auf einem derartigen mikroskopisch exakten Niveau, wie das Programm es zu tun pflegt, ihr Sprachverständnis auf eine höhere Stufe heben könnten, war unser nächster Schritt eindeutig vorgegeben: ein System bauen, mit dem Sätze von Schüler/innen verbessert bzw. überhaupt (manuell) analysiert werden können.

So begann die Entwicklung von Arethusa. Unterstützung fanden wir hier bei dem Digital Humanities Departement an der Universität Leipzig und der Tufts University in Boston sowie dem Alpheios Project, das auf dem Gebiet des e-learning für Latein bereits eine Umgebung mit einigen Tools entwickelt hatte.

Arethusa wird zurzeit aktiv – hauptsächlich von Studenten an amerikanischen Universitäten – genutzt. Die Arbeit am System ist aber noch nicht abgeschlossen. Stets werden Verbesserungsvorschläge von aktiven Usern umgesetzt und Ideen für neue Funktionalitäten implementiert. Arethusa ist darauf ausgelegt, Erweiterungen problemlos zu integrieren, sodass zwei unabhängig voneinander entstandene Features gemeinsam mit- und nebeneinander verwendet werden können. Doch dazu später etwas mehr.

Soweit sei die Hintergrundgeschichte von Arethusa und seinen Entwicklern preisgegeben. Betrachten wir nun jedoch die Plattform und ihre Funktionsweise.

### *Labor est etiam ipsa voluptas*

Arethusa ist ein Annotationsframework. Es ermöglicht, lateinische Wörter mit bestimmten Informationen zu erweitern, zu annotieren. Dadurch wird aus einem einfachen Text, der nichts von sich selbst weiß, eine intelligente Datenstruktur, die Auskunft über gewisse Sachverhalte liefert. Beispielsweise können syntaktische und morphologische Daten für jedes einzelne Wort gespeichert werden. So weiß der Satz „*Exigua pars est vitae, qua vivimus.*“ von selbst, nachdem ein User seine syntaktischen und morphologischen Eigenheiten annotiert hat, wo sein Subjekt, Objekt und Prädikat sind. Arethusa erlaubt also, das manuell zu tun, was der LLT-Parser automatisch tut: eine morphologische und syntaktische Analyse von lateinischen (und auch griechischen) Sätzen. Werfen wir zuerst einen Blick auf die gesamte Oberfläche, welche sich dem Nutzer zeigt, wenn er genannten Satz annotieren möchte.

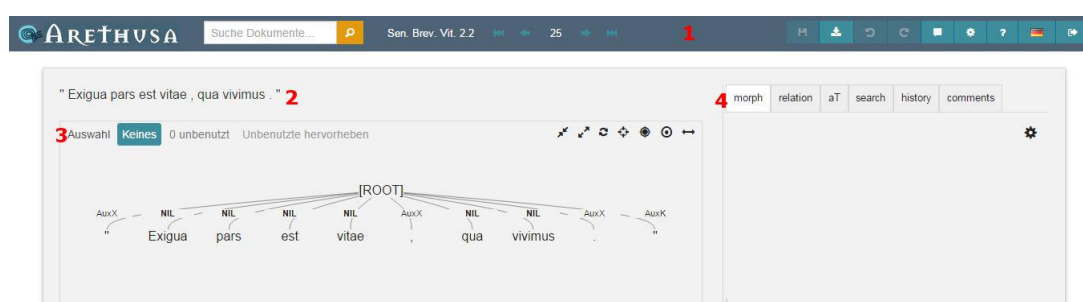


Abb. 1

Zuoberst befindet sich eine Menüleiste (1), die unter anderem das Navigationstool und Einstellungsoptionen (z.B. Spracheinstellungen) beinhaltet. Die eigentliche Arbeitsfläche ist zweigeteilt: links befindet sich die Textanzeige (2) und die graphische Darstellung der Analyse (3), rechts der Platz für Analysewerkzeuge (4).

Der angezeigte Satz ist soweit noch nicht annotiert. In einem ersten Schritt könnte man die Morphologie eines jeden Wortes bestimmen. Klickt man beispielsweise mit der Maus auf das Wort *vitae*, erscheint rechts unter dem Tab mit dem Titel *morph* eine morphologische Analyse. Diese wurde von einem externen Service bereitgestellt, Arethusa zeigt nur dessen Ergebnis an. Bereits an diesem Exempel lässt sich die Funktionsweise der Plattform gut erklären: Arethusa selbst kann mit Latein nichts anfangen. Es stellt nur einen Raum zur Verfügung, in dem verschiedene externe Tools, die die lateinische oder griechische Sprache verarbeiten können, zusammenkommen und interagieren. Gäbe es beispielsweise einen Service, der die Metrik lateinischer Wörter analysiert, könnte dieser mit Arethusa verbunden werden, um ihn in Kombination mit einem anderen Tool, z.B. dem Morphologie-Service, zu verwenden. Das Programm ist genau auf diese Modularität hin ausgerichtet: Einzelne, konfigurierbare Bausteine können nach individuellem Bedarf zusammengesetzt werden. Für unsere vorliegende Aufgabe benötigen wir den Baustein „lateinische Morphologie“ (und nicht etwa die griechische). Für *vitae* liefert uns der Service vier Möglichkeiten, die Arethusa allesamt anzeigt. Dem User obliegt es nun, die richtige zu wählen. Aus dem Kontext kann man erschließen, dass es sich hier um einen Genetiv handelt.

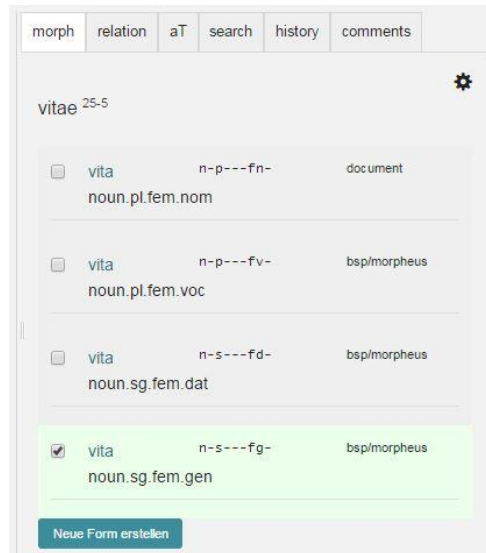


Abb. 2

Sollte der externe Service bei der Analyse einer Form fehlerhaft sein, z.B. könnte die richtige Form nicht aufgelistet sein, bietet Arethusa die Möglichkeit, die fehlende Form manuell nachzutragen.

Führt man die Auswahl der morphologischen Formen nun bei allen Wörtern durch, erkennt man, dass sie ihre Farben entsprechend ihrer Morphologie wechseln. In unserem Beispiel würde die optische Unterstützung so aussehen:

" Exigua pars est vitae , qua vivimus . "

Abb. 3

Die Einfärbung ist optional; es besteht kein Zwang, die Farben zu nutzen.

Als nächstes wagen wir uns an die graphische Analyse des Satzes. Die dahinterliegende Idee basiert auf dem Dependenzmodell. Dieses ist, wenn nicht in der Theorie, sicherlich aus der Praxis des Lateinlehrens bekannt. Es äußert sich in der hierarchischen Beantwortung folgender Fragen: Wo ist das Prädikat im Hauptsatz? Wo ist das Subjekt? Wo das Objekt? usw. Das Modell ordnet die Wörter des Satzes nach ihren Dependenz an, wobei ein Wort nur von einem einzigen abhängen kann. Die höchste Instanz ist die sogenannte Root-Node, die den Satz als gesamten repräsentiert. Das Endergebnis ist ein sogenannter Dependenzbaum. Möchte man in Arethusa einen solchen Baum zeichnen, so klickt man zuerst auf das Wort, welches man positionieren möchte, und dann auf dasjenige, von welchem es abhängt.

Zusätzlich zur Anordnung der Wörter in einem solchen Baum können die einzelnen Relationen (= Dependenzverbindungen) von Wort zu Wort mit einem sogenannten Label bezeichnet werden, das die Art der Verbindung näher bestimmt. Auch diese Aufgabe kann mit Arethusa erfüllt werden. Auf der rechten Seite der Oberfläche wird hierzu das richtige Tool unter dem Tab *relation* bereitgestellt.

In vorliegendem Beispiel ist *est* von der Root-Node abhängig. Die Art der Verbindung ist als Prädikat zu bezeichnen, im Baum als *PRED* dargestellt. *pars* ist dazu das Subjekt (*SBJ*), *exigua* das Prädikatsnomen (*PNOM*). *Vitae* wiederum ist ein Attribut (*ATR*) zu *pars*. Der Relativsatz beschreibt *vitae* näher, weshalb dessen Prädikat das Label *ATR* bekommt, das Relativpronomen ist eine adverbielle Ergänzung zu *vivimus*.

Der fertige Baum für unser Beispiel würde wie folgt aussehen:

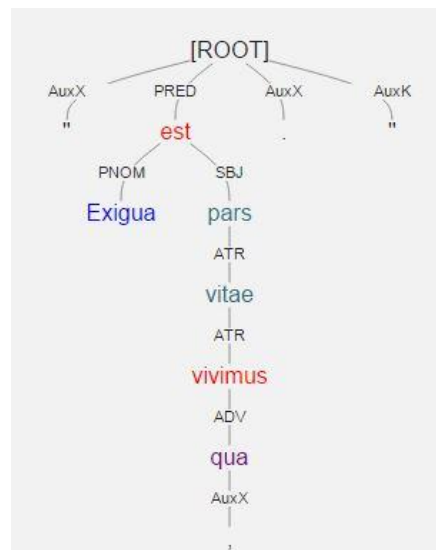


Abb. 4

Für die Anordnung und Benennung der einzelnen Dependenz gibt es verschiedene Ansichten. Das hier dargestellte Beispiel orientiert sich an den Annotationsregeln, die von der Tufts University in Boston herausgegeben wurden, die sich wiederum das Modell der *Prague Dependency Treebank* (<http://ufal.mff.cuni.cz/pdt2.0/>) zum Vorbild genommen hat. Als Lehrer/in ist man grundsätzlich nicht gezwungen, im Schulgebrauch dieses System blind zu übernehmen. Es ist möglich, sich für seine individuellen Bedürfnisse eigene Labels zu erstellen.

Die graphische Darstellung dieses einfachen Satzes sieht recht unspektakulär aus. Dennoch sind die wichtigsten Dinge auf einen Blick zu erkennen. Wagt man sich nun an die Übersetzung des Satzes, wird man kaum noch auf Hindernisse stoßen.

Je länger und verschachtelter lateinische Sätze werden, desto größer ist der Nutzen, den die Anordnung der syntaktischen Relationen in einem Dependenzbaum für Lernende bringt. Auf einen Blick kann an den Ästen des Baumes abgelesen werden, welches Wort zu welchem syntaktischen Konstrukt gehört.

Eugippius liefert hierfür ein anschauliches Beispiel (*Vita Sancti Severini* 29):

*Per idem tempus Maximus Noricensis, cuius fecimus in superioribus mentionem, fidei calore succensus media hieme, qua regionis illius itinera gelu torpente clauduntur, ad beatum Severinum audaci temeritate vel magis, ut post claruit, intrepida devotione venire contendit, conductis plurimis comitibus, qui collo suo vestes captivis et pauperibus profuturas, quas Noricorum religiosa collatio profligaverat, baiularent.*

Dieser Satz besteht aus 53 Wörtern, fünf Nebensätzen, einem Ablativus Absolutus und einem Hauptsatz, dessen Bestandteile weit verstreut auftauchen. Die graphische Analyse sieht wesentlich komplexer aus als die des obigen Satzes.

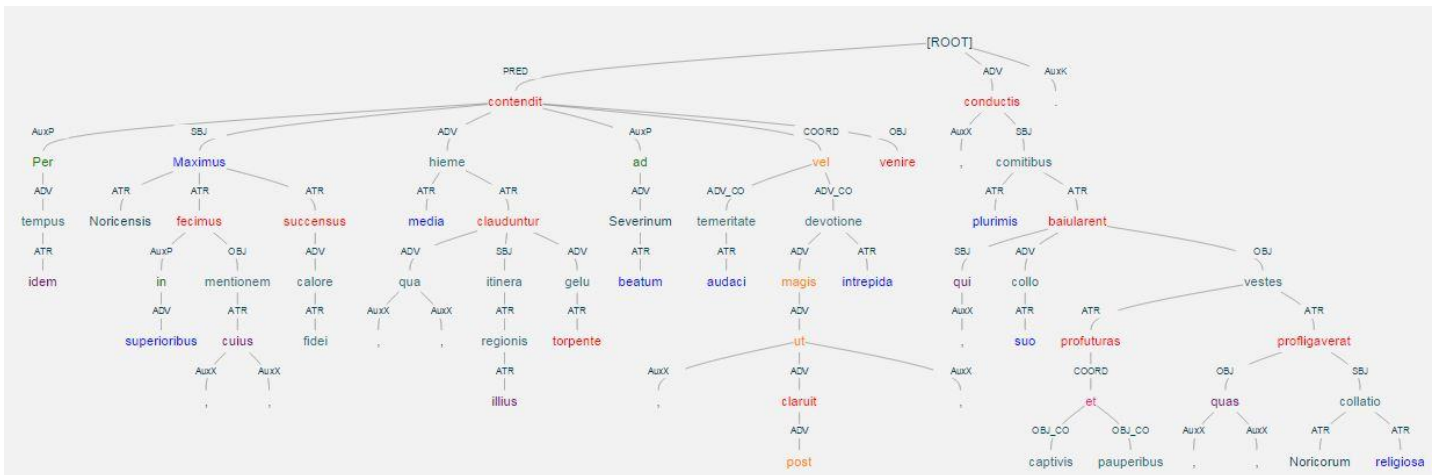


Abb. 5

Auf den ersten Blick erkennt man jedoch sofort die zwei Hauptäste des Baumes: Einer startet bei *contendit*, der andere beim Partizip des Ablativus Absolutus *conductis*. An dieser Graphik erkennt man auch die Bedeutung einer solchen Konstruktion: Sie hat mit dem Hauptsatz nichts zu tun und ist von diesem in allen Belangen getrennt ...

Das Subjekt zu *contendit* mit all seinen dazugehörigen Abhängigkeiten ist ebenso gleich ersichtlich wie die Präpositionalgruppen um *per* und *ad* und die adverbielle Ergänzung *hieme*.

Auf diese Weise kann ein „obskurer“ Satz um ein vielfaches zugänglicher aufbereitet werden. Die mikroskopische Analyse bietet eine ausgezeichnete Basis für eine Übersetzung.

### Cui bono?

An diesem Punkt mag einigen die Frage nach dem Nutzen in die Köpfe schießen. Man könnte Arethusa entgegenhalten, dass man graphische Analysen auch auf einem Stück Papier herstellen kann und dass man bei dieser im Grunde zeitaufwendigen Methode damit vielleicht sogar schneller wäre als am Computer.

Um dieses Gegenargument zu entkräften, bedarf es einer allgemeinen Überlegung zu Zeit und Schnelligkeit beim (Latein-)Lernen. Zweifellos können Sätze ohne diese Analyse rascher übersetzt werden. Jedoch steht bei dieser Methode das exakte Analysieren im Vordergrund. Die Übersetzung ist „nur“ ein Folgeprodukt. Ziel ist, die Struktur lateinischer Sätze zu erfassen, womit der Zeitaufwand ohne Weiteres zu rechtfertigen ist. Durch diese Strukturanalysen ergeben sich ungeahnte Erkenntnismöglichkeiten. (Viele dieser Erkenntnisse sind übrigens nicht nur für Latein relevant. Sie lassen sich auch auf andere Sprachen anwenden, wodurch allgemein das Sprachvermögen verbessert wird.)

Zusätzlich darf nicht aus den Augen verloren werden, dass ein Abhängigkeitsbaum nicht nur direkt dem Ersteller eine Hilfe bietet, sondern auch allen anderen, mit denen diese Lösung geteilt wird. Es muss nicht jede/r Schüler/in den gleichen Satz mehrmals annotieren – sie profitieren vom Werk eines einzelnen, der den Satz in einer sogenannten Treebank speichert (eine Datenbank für Abhängigkeitsbäume). Sie müssten dessen Analyse aber nicht kritiklos übernehmen. Arethusa bietet nämlich auch eine Kommentarfunktion an, mittels derer Diskussionen über Abhängigkeiten, Labels und Anderes geführt und – aus Lehrersicht nicht uninteressant – verfolgt werden können. Dieses gesamte Setting ermöglicht für Schule und Wissenschaft ungeahnte Chancen zur Kollaboration. Wollte man bisher gemeinsam einen lateinischen Text analysieren, mussten alle Beteiligten zur selben Zeit am selben Ort sein.

Arethusa hebt diese Einschränkung auf und ermöglicht eine asynchrone und atopische Zusammenarbeit: Je nach Arbeitsteilung kann ein User einen Teil bearbeiten, der für alle "MitarbeiterInnen" auch sogleich zu sehen ist. Es bedarf keiner lästigen E-mailkorrespondenz über die Dinge, die bereits erledigt sind. So kann beispielsweise User A die Morphologie aller Wörter bestimmen, während User B die graphische Analyse erstellt und User C die Abhängigkeiten benennt. Alle drei lesen den Text und tragen durch ihre Arbeit zu einer gemeinsamen Lösung bei.

Auf diese Weise werden auch Probleme, die durch die unterschiedliche Lesegeschwindigkeit entstehen, hinfällig. Jeder kann sein eigenes Tempo gehen.

Ein zusätzlicher Effekt ist die Kontrolle. Entweder existieren zu meinen Ergebnissen bereits Lösungsvorschläge anderer Nutzer, die ich auf Knopfdruck vergleichen kann, oder ein weiterer User nimmt sich meiner Resultate an und korrigiert diese.

Allgemein kann Arethusa dazu beitragen, dass Latein Lernende Texte öfter gemeinsam analysieren und in der Folge übersetzen. Unserer Beobachtung nach gibt es zwar Lerngruppen, die gemeinsam Inhalte oder Grammatiktheorie erarbeiten, gemeinsames Übersetzen ist jedoch äußerst selten.

Wenn Nutzer des Systems gemeinsam arbeiten und intensiv über gewisse Textpassagen diskutieren, in dessen Folge auch mehrere graphische Analysen zu ein und demselben Satz erstellt werden, bietet Arethusa ein „Review“-Tool an, das es erlaubt, Bäume leicht miteinander zu vergleichen. Dieses Feature ist vor allem auch für Lehrer/innen interessant. Sollte die Lehrperson beispielsweise im Rahmen einer Hausübung den Schüler/innen den Auftrag erteilen, einen Satz mit Hilfe von Arethusa zu dekonstruieren, kann sie eine zuvor selbst erstellte Lösung per Mausklick mit den Ergebnissen der Aufgabe kontrollieren (und mit der Kommentarfunktion Anmerkungen hinterlassen). Für einen Beispielsatz aus Caesars *Bellum Gallicum* präsentiert sich das Review-Tool so:

The screenshot displays the 'review' tool interface. At the top, there is a navigation bar with buttons for 'review', 'comments', 'search', 'morph', 'relation', and 'aT'. Below this, there is an 'Auto-compare' checkbox and a green bar indicating '4 tokens and 5 attributes with differences'. A 'Compare' button is located on the right side of the interface. The main area shows two syntax trees for the Latin sentence: 'Cum esset Caesar in citeriore Gallia in hibernis, ita uti supra demonstravimus, crebri ad eum rumores adferebantur que litteris item Labieni certior fiebat omnes Belgas, quam tertiam esse Galliae partem dixeramus, contra populum Romanum coniurare que obsides inter se dare'. The left tree is the user's solution, and the right tree is the system's reference solution. The trees are color-coded to show differences between the two solutions. The interface also includes a 'Auswahl' section with a 'Keines' button and a 'Toggle Diff' button.

Abb. 6



Auf der rechten Seite befindet sich die Lösung, auf der linken der zu vergleichende Baum. Unterschiede werden, um sie leichter sichtbar zu machen, in Rot dargestellt. Auf dieser Basis lassen sich Fehler leicht quantifizieren, qualitativ beurteilen und in einer Konsequenz daraus auch beheben bzw. – sofern die Lösung nicht allgemein anerkannt ist – über die Analyse disputieren.

### ***Curiositas enim experiendi incitamentum facit***

Eine solche Diskussion bewegt sich bereits auf wissenschaftlichem Boden. Darin liegt ein weiterer Vorteil in der Nutzung von Arethusa: die Hinführung zu einem wissenschaftlichen Verantwortungsgefühl und zum wissenschaftlichen Arbeiten generell. Denn jede Annotation kann (freiwillig) veröffentlicht werden. Von derartigen Publikationen kann die Wissenschaft enorm profitieren, da sie lateinische Texte quantifizierbar macht. Wenn ein Text morphologisch und syntaktisch vollständig annotiert ist, ist er maschinenlesbar, also für einen Computer in einem gewissen Maße verständlich. Dadurch können erstmals Fragen gestellt werden, an die man zuvor niemals dachte und die dazu noch in einem Augenblick beantwortet werden können. Beispielsweise wie oft gewisse Verben ein Objekt im Akkusativ nach sich ziehen oder wie oft sich Caesar im *Bellum Gallicum* als Subjekt einer aktiven Handlung bezeichnet. Vielleicht mag vielen eine solche Untersuchung sinnfrei erscheinen. Doch allein zu wissen, dass die Möglichkeit zur raschen Beantwortung solcher Fragen existiert, ist ein enormer Fortschritt in der Klassischen Philologie. Und letztendlich kann nie vorausgesagt werden, welche Teilergebnisse gemeinsam ein größeres Rätsel einer zukünftigen Forschungsfrage lösen können.

Aus diesem Blickwinkel könnten allgemeine Fragen zur Stilistik eines Autors oder gar die Entwicklung der lateinischen Sprache hinsichtlich Syntax und Grammatik mit vielen annotierten Werken neu in Angriff genommen werden.

User, die ihre wissenschaftlichen Analysen und Kommentare unter anderem für solche Zwecke publizieren, tragen aktiv etwas zur Forschung bei. Schüler/innen lernen nicht nur Latein für sich, jedes analysierte Wort vermag Wissenschaftler/innen rund um den Globus helfen ihre Forschung voranzutreiben. Dieses Gefühl, auf einem bestimmten Gebiet etwas vollbracht zu haben und zukünftig vollbringen zu können, den Schüler/innen näherzubringen, muss Ziel des Unterrichtens sein. Damit sie erkennen, dass Latein nicht tot ist und ihre (Lern)Arbeit dabei hilft, die Kultur, die Denkweise und die Menschen hinter dieser Sprache zu erhalten, in der sich eine Quelle der europäischen Kultur versteckt.

Arethusa ist nichts anderes als ein Werkzeug, das beim Sammeln antiker Texte hilft, und seine Nutzer die Mönche des Mittelalters, die in mühevoller Arbeit Handschrift um Handschrift kopiert haben, um sie nicht der Vergessenheit anheimfallen zu lassen – mit dem Unterschied, dass wir heute nicht auf schöne Manuskripte Wert legen, sondern auf Daten, die uns dabei helfen, diese Texte besser zu verstehen.

**Infobox:****LLT – Latin Language Toolkit**

Gernot Höflechner: Studium der Alten Geschichte und Klassischen Philologie in Graz, zurzeit als Softwareentwickler tätig

Robert Lichtensteiner: Studium der Alten Geschichte, Klassischen Philologie und Geschichte (letztere auf Lehramt)

Christof Sirk: Studium der Biomedizintechnik in Graz, Softwareentwickler

<http://latin-language-toolkit.net/>

**Arethusa**

<http://arethusa.latin-language-toolkit.net/>

(Landing Page)

<https://github.com/latin-language-toolkit/arethusa>

(Open Source Code)

<http://sosol.perseids.org/sosol/>

(Arethusa integriert in der Perseids-

Umgebung: Anmeldung über Google, Yahoo oder AOL Account erforderlich)

**Partner und Kooperationen**

Perseus Digital Library

<http://www.perseus.tufts.edu/hopper/>

Perseids Collaborative Editing Platform

<http://www.perseids.org/>

Digital Humanities Leipzig

<http://www.dh.uni-leipzig.de/wo/>

Alpheios Project

<http://alpheios.net/>

**Kontakt**

Gernot Höflechner

[1986gh@gmail.com](mailto:1986gh@gmail.com)

Robert Lichtensteiner

[robert.lichtensteiner@gmail.com](mailto:robert.lichtensteiner@gmail.com)

Christof Sirk

[christof.sirk@gmail.com](mailto:christof.sirk@gmail.com)

LLT

[latin.language.toolkit@gmail.com](mailto:latin.language.toolkit@gmail.com)