Ein rechnergestützter Vergleich von Jacob Baldes *Encomium toruitatis* mit der übrigen lateinischen Literatur

Robert West

6. November 2006

1 Lokale Übereinstimmung von Wortfolgen

Einer der Punkte, die bei der Lektüre von Jacob Baldes *Encomium toruitatis* sofort auffallen, ist, daß Balde sehr häufig Texte früherer Autoren anklingen läßt oder sogar ausdrücklich zitiert. Während der in der lateinischsprachigen Literatur bewanderte Leser bei vielen dieser Stellen den Deut verstehen oder die Quelle erkennen wird, so ist dies bei Lesern mit einem weniger breiten Überblick über die Literatur nicht der Fall – ja es wird sicherlich sogar dem Experten die ein oder andere Ähnlichkeit entgehen.

Daher haben wir ein Computerprogramm das *Encomium* mit einer Vielzahl anderer lateinischer Texte vergleichen lassen. Die Ergebnisse wollen wir hier kurz kommentieren.

Natürlich handelt es sich beim hier Präsentierten nicht um philologische Arbeit im eigentlichen Sinn, aber vielleicht erweist sich die Informationen, die der Rechner herausfiltern konnte, ja als eine nützliche Vorstufe für selbige.

1.1 Vergleichstexte

Das *Encomium toruitatis* wurde mit allen Texten, die in der Online-Bibliothek unter der Adresse `thelatinlibrary.com` verfügbar sind, verglichen; es handelt sich dabei um etwas mehr als 1000 Dokumente. Es wurden lediglich die neulateinischen Texte ausgelassen.

Die Dateien wurden aus dem Internet heruntergeladen und normalisiert, um die Texte mit Balde vergleichbar zu machen. Die Normalisierungen, die sowohl auf dem Baldetext als auch den anderen Dokumenten ausgeführt wurden, eliminierten Information, die nicht wesentlich für den Vergleich der Texte ist. Es handelt sich um folgende Schritte:

- Alle Buchstaben wurden zu Großbuchstaben gemacht.
- ‘U’ wurde zu ‘V’, ‘J’ zu ‘I’.
- Alle diakritischen Zeichen wurden entfernt.
- Satzzeichen und Zeilennummern wurden entfernt.

1.2 Suchkriterium

Es wurden alle Stellen gefunden, die in mindestens drei aufeinanderfolgenden Wörtern mit dem Baldetext übereinstimmen.

Stellen, die in mehr als drei Wörtern übereinstimmen, werden damit automatisch auch gefunden. Dasselbe Verfahren läßt sich natürlich auch mit zwei statt drei Wörtern durchführen, doch ist die Treffermenge dann so dicht, daß die Sichtung der Ergebnisse recht beschwerlich wird.

Es werden also bei weitem nicht alle Ähnlichkeiten – erst recht keine inhaltlichen! – herausgefunden; die Treffermenge beschränkt sich auf die explizite Übereinstimmung von Dreier-Wortgruppen.

Die Ergebnisse liegen diesem Dokument als Anlage bei, und zwar in doppelter Ausfertigung: Sortiert nach dem Vorkommen bei Balde und sortiert nach dem Vorkommen in den anderen Texten.

Die Zeilennumerierung bei Balde ist korrekt, bei den anderen Texten sollte sie eher als Anhaltspunkt dienen (zumal sie bei Prosatexten ohnehin wenig Sinn hat); diese kleine Unvollkommenheit liegt an der elektronischen Darstellung der Texte im Online-Archiv.

Die etwas kryptisch anmutenden Textbezeichnungen sind wie folgt zu verstehen: lautet die Textbezeichnung `XYZ`, so bedeutet dies, daß der entsprechende Text unter

- `http://thelatinlibrary.com/XYZ.html` oder
- `http://thelatinlibrary.com/XYZ.shtml`

verfügbar ist. In den allermeisten Fällen ist aus der Bezeichnung ohne Weiteres ersichtlich, um welchen Text es sich handelt.

1.3 Technisches

Das Programm lief etwa 47 Stunden auf einem Rechner der TU München. Die verwendete Programmiersprache ist “Perl”.

2 Globale Ähnlichkeit gemäß einem Verfahren aus der künstlichen Intelligenz

2.1 Ergebnis

Während das oben beschriebene Verfahren nach *lokalen* Ähnlichkeiten zwischen dem *Encomium toruitatis* und anderen Texten suchte, wollen wir nun noch ein interessantes Verfahren vorstellen, das die *globale* Ähnlichkeit zwischen zwei Texten feststellt.

Das Ziel ist es, eine Rangliste der lateinischen Literatur zu erstellen: Je ähnlicher ein Text zu unserem Baldetext ist, einen umso höheren Platz soll er in der Liste einnehmen. Als Vergleich dienten wiederum die über 1000 bei *The Latin Library* im Internet verfügbaren Texte (in ihrer normalisierten Version). Das *Encomium* wurde mit jedem Text nach dem sogleich zu beschreibenden Verfahren verglichen, wobei ein Abstandswert zwischen 0 und 1 herauskam: Abstand 0 bedeutet, daß die Texte identisch sind, und je näher bei 1 der Wert liegt, für desto verschiedener wurden die Texte befunden.

Das Ergebnis (beschränkt auf die ersten vierzig Platzierungen) ist in Abbildung 1 aufgelistet. Wie man sieht, hält auch der Rechner Balde für einen “deutschen Horaz”! – Die größte Ähnlichkeit wird mit dessen zweitem Satirenbuch festgestellt; ohne den Inhalt der Texte zu verstehen, geschweige denn sie explizit beide als Satiren einordnen zu können, hat unser Verfahren also befunden, daß Baldes Satire von aller wichtigen lateinischen Literatur Horazens Satiren am ähnlichsten ist – umso erstaunlicher, da das zuvor beschriebene Programm kein einziges Zitat aus dem zweiten Satirenbuch mit mindestens drei identischen Wörtern fand! Dies macht deutlich, daß die globale Ähnlichkeit zweier Texte nicht zwangsläufig von einer lokalen Ähnlichkeit herrühren muß.

Daß auch Ovid so häufig vorkommt, nimmt wegen der häufigen Anklänge und Zitate wenig Wunder. Erstaunlich hingegen erscheint die Frequenz von Statius in den oberen Rängen. Eine Interpretation ist mir hier mangels Kenntnis von Statius’ Werk nicht möglich.

2.2 Verwendetes Verfahren

Es soll genügen, die Idee hinter dem verwendeten Verfahren grob zu skizzieren.

Wie bereits erwähnt, sucht der Algorithmus nicht nach lokalen Ähnlichkeiten (etwa in Form von übereinstimmenden Dreier-Wortgruppen wie das anfangs beschriebene Programm); es wird vielmehr die globale Ähnlichkeit abgeschätzt.

Um den grundlegenden Gedanken zu erleuchten, wollen wir zuerst einen Extremfall betrachten. Man stelle sich ein Programm (etwa ZIP) vor, das einen Text komprimiert, um

Rang	**Abstand**	**Text**
1	0.957079680960278	Hor. serm. 2
2	0.957790413014294	Hor. epist. 1
3	0.958698570638869	Lucan. 9
4	0.958777540867093	Stat. Theb. 10
5	0.958935481323541	Ov. met. 15
6	0.959014451551765	Ov. met. 7
7	0.959053936665877	Stat. Achill. 1
8	0.959132906894101	Stat. Siluae 1
9	0.959172392008213	Ov. met. 2
10	0.959211877122325	Silius 13
11	0.959290847350549	Ov. met. 8
12	0.959448787806997	Stat. Theb. 12
13	0.959606728263445	Hor. serm. 1
14	0.959764668719893	Ov. met. 13
15	0.959883124062229	Prop. 3
16	0.959922609176341	Stat. Theb. 4
17	0.959962094290453	Stat. Siluae 5
18	0.960001579404564	Stat. Siluae 2
19	0.960001579404564	Ov. met. 4
20	0.960080549632788	Stat. Theb. 6
21	0.960199004975124	Ov. met. 14
22	0.960228562779081	Sannazaro 1
23	0.96031746031746	Ov. am. 3
24	0.960356945431572	Stat. Siluae 3
25	0.960435915659796	Ov. met. 3
26	0.960554371002132	Prop. 2
27	0.960633341230356	Silius 15
28	0.960791281686804	Verg. Aen. 6
29	0.960791281686804	Ov. fasti 3
30	0.960791281686804	Stat. Theb. 9
31	0.960791281686804	Iuu. 6
32	0.960830766800916	Stat. Theb. 7
33	0.960830766800916	Val. Flacc. 1
34	0.96090973702914	Verg. Aen. 8
35	0.96090973702914	Lucan. 6
36	0.960949222143252	Silius 4
37	0.960949222143252	Lucan. 4
38	0.960949222143252	Claudian. cons. 6
39	0.961028192371476	Stat. Theb. 2
40	0.961067677485588	Rutil. Namat. de reditu suo

Abbildung 1: Die 40 gemäß des im Text beschriebenen Algorithmus zu Baldes *Encomium* ähnlichsten Texte.

Speicherplatz zu sparen. Die Aufgabe eines solchen Programms ist es, einen Text in einen anderen, *kürzeren* (für den Menschen freilich nicht mehr verständlichen) Text zu "übersetzen", der bei Bedarf (beim sogenannten Dekomprimieren) in den ursprünglichen Text zurückübersetzt werden kann. Angenommen der ursprüngliche Text U wird übersetzt in den komprimierten Text K; der entscheidende Punkt ist zu verstehen, was ZIP tut, wenn es nun nicht U sondern UU (den Text, der entsteht, wenn man zweimal U aneinanderhängt) komprimieren soll: um UU vollständig zu beschreiben, genügt *zusätzlich zu* K folgende kurze Information: "Alles Bisherige nochmal!"[1] Diese Zusatzinformation ist unabhängig von der Größe des zu komprimierenden Textes. Es gilt also, daß das komprimierte UU nur sehr wenig größer ist als das komprimierte U (eben so viel Platz die Anweisung "Alles Bisherige nochmal!" benötigt).

Gehen wir einen Schritt in Richtung Abstraktion von diesem Extremfall. Nehmen wir nicht zwei identische sondern zwei ähnliche Texte: Angenommen wir haben einen Text AB, der selbst aus dem Text A gefolgt von B besteht; der zweite Text BA bestehe aus B gefolgt von A. Wie können wir den zusammengesetzten Text $ABBA$ möglichst kurz darstellen? – Wir benötigen die komprimierte Version von AB *zuzüglich* folgender kurzer Information: "Den zweiten Teil vom Bisherigen nochmal; den ersten Teil vom Bisherigen nochmal". Da AB und BA sich sehr ähnlich sind, benötigen wir also auch hier nur wenig mehr Speicher für das komprimierte $ABBA$ im Gegensatz zum komprimierten AB. – Allerdings benötigen wir schon mehr Zusatzinformation als im oben beschriebenen Fall zweier identischer Texte.

Allgemein gilt: Je ähnlicher ein Text Y einem Text X ist, desto weniger Zusatzinformation benötigen wir zusätzlich zum komprimierten X, um Y eindeutig darzustellen.

Dies macht sich der Algorithmus zunutze, den wir verwendet haben, um das *Encomium* (E) mit anderen Texten zu vergleichen. Es wird wie folgt verfahren, wenn z.B. mit der *Ars poetica* (A) verglichen werden soll:

1. Komprimiere E mit ZIP und messe die Größe G_E der resultierenden Datei.
2. Hänge E und A zusammen zu EA.
3. Komprimiere EA mit ZIP und messe die Größe G_{EA} der resultierenden Datei.
4. Errechne die Differenz von G_{EA} und G_E. Diese kann als Abstand von E zu A interpretiert werden. Es handelt sich dabei um die zusätzlich zum komprimierten E benötigte Information.
5. Normalisiere den Abstand (so daß bei allen Vergleichen stets eine Zahl zwischen 0 und 1 herauskommt; nur so kann man die verschiedenen Abstände vergleichen, um die Rangliste zu erstellen).

Auf diese Weise wurde der Abstand vom *Encomium* mit jedem anderen der über 1000 Werke errechnet und in der in Abbildung 1 gezeigten Rangliste geordnet.

2.3 Ausblick: Klassifizierung von Texten

Es sei noch kurz ein interessantes Ergebnis präsentiert, das wir mit demselben Verfahren erzielten und das in gewissem Maße die Plausibilität der oben beschriebenen Rangliste belegen soll.

Ziel ist es, eine Art "Stammbaum" für eine Menge von Texten zu erstellen.

Dazu errechnen wir von jedem Text zu jedem anderen Text den Abstand mittels des oben beschriebenen Algorithmus. Die zwei für am ähnlichsten befundenen Texte werden im Stemma zusammengeführt. Die zwei Texte bilden nun die erste "richtige" Gruppe, während alle anderen Texte noch jeder für sich eine eigene Gruppe bilden. Auf diese Weise nehmen wir in jedem Schritt einen Text in eine Gruppe auf (im Stemma werden der Text und die Gruppe zusammengeführt), bis sich eine Baumdarstellung wie in Abbildung 2 ergibt. In dem

[1] Man beachte, daß auch Menschen im Kopf derartige Verfahren verwenden, um sich Daten zu merken: die Telefonnumer 55555 wird man sich nicht als "Fünf Fünf Fünf Fünf Fünf" sondern als "fünfmal die Fünf" merken.

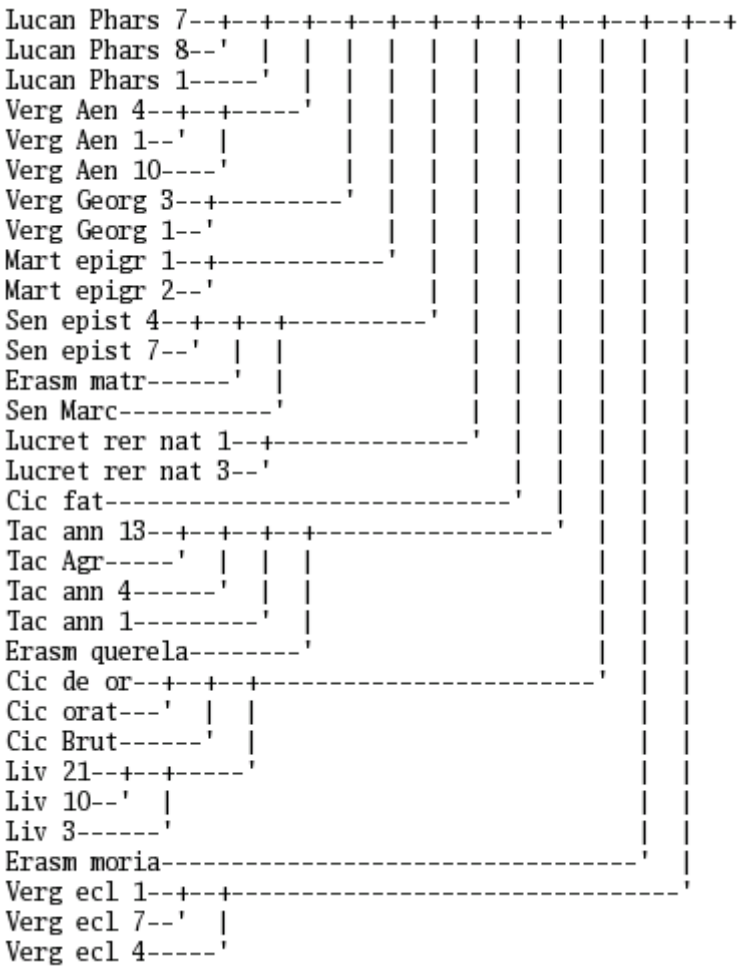

Abbildung 2: Automatische Klassifizierung einer Auswahl von Texten in Stemmadarstellung. (Das Stemma ist um 90 Grad rotiert, so daß sich die Wurzel rechts befindet.)

dort dargestellten Beispiel haben wir 33 mehr oder minder willkürlich ausgewählte Texte klassifiziert.

Beim Betrachten des Ergebnisses fällt auf, daß fast überall zuerst die Bücher eines Werkes[2] und dann die Werke eines Autors gruppiert wurden. Für die wenigen Ausnahmen gibt es oft plausible Erklärungen: So wurde etwa die *Aeneis* mit Lukans *Pharsalia* eher als mit Vergils *Georgica* zusammengeführt; dies rührt wohl daher, daß die *Pharsalia* und die *Aeneis* beide von Bürgerkriegen handeln und daher erstens von vorne herein vieles an gemeinsamem Wortschatz haben dürften und zweitens die *Aeneis* eine Vorbildfunktion für die *Pharsalia* hatte.[3] Die krasseste Ausnahme bilden jedoch die *Eklogen*, die, obwohl vom selben Autor, am weitesten vom restlichen Vergil entfernt angesiedelt werden. Man könnte darin einen Hinweis auf den eigenständigen Charakter des Werkes sehen.[4]

Das Resultat ist recht beeindruckend, wenn man sich ins Gedächtnis ruft, daß überhaupt kein Vorwissen im Algorithmus verwendet wird, weder über die lateinische Sprache noch über die römische Literatur noch über sonst irgend etwas Problemspezifisches. Die festgestellte

[2] Eine merkliche Ausnahme und gleichsam Diaspora bildet lediglich Erasmus.

[3] Zwar habe ich die *Pharsalia* nicht gelesen, vgl. aber M. Fuhrmann, *Geschichte der römischen Literatur*, Stuttgart, o.J., S. 265: "Der Aufbau der *Pharsalia* beruht **wie deren formales Vorbild**, die Vergilische *Aeneis*, auf dem Widerspiel von Hexaden und Tetraden."

[4] Mit seinen bukolischen Dihäresen setzt sich das Werk ja auch rein metrisch von den übrigen hexametrischen Dichtungen im Stemma ab.

Ähnlichkeit ist rein äußerlich und beruht allein auf der Abfolge der Buchstaben im Text. Das gleiche Verfahren kann man ohne jegliche Veränderung verwenden, um etwa die Genome verschiedener Tierarten zu einem Artenstammbaum zu ordnen oder um aus Übersetzungen eines identischen Urtextes (z.B. der UN-Menschenrechtserklärung) einen Stammbaum der Sprachen zu erstellen.

Die Güte des Ergebnisses, das das Verfahren liefert, kann man als Hinweis auf die Plausibiltät der Rangliste aus Abbildung 1 auffassen.