Wort	Anzahl	Wort	Anzahl
et	373596	sunt	44550
in	250758	per	42081
est	171009	se	40733
que	150721	enim	40511
ut	134350	ab	37589
non	131925	nec	36390
cum	108737	etiam	35838
ad	96350	autem	31760
quod	89124	id	31421
qui	71400	atque	30777
si	71196	ac	30579
quae	68120	ne	30039
sed	65265	quid	29619
ex	57516	haec	28993
de	56801	te	27494
a	56216	quo	27143
quam	51254	vel	27108
aut	50196	me	27015
esse	49777	nam	26992
hoc	48359	sit	26983

Tabelle 4. Frequenzsortierte Wortliste BTL als Basis einer Stoppwortliste

Korpusanalyse

Computergestützte Verarbeitung von Sprache

Für die Gewinnung strukturierter Informationen aus Texten kommen, je nach Anwendungsfall, verschiedene Sprachtechnologie-Komponenten zum Einsatz. Bei der Verarbeitung antiker Texte ergeben sich, beispielsweise durch das Fehlen von sogenannten Metadaten, einige Besonderheiten, so dass nicht alle Komponenten berücksichtigt werden. Nachfolgend soll eine grobe Aufstellung der zum Einsatz kommenden Sprachtechnologie gegeben werden.

Grundsätzlich wird innerhalb von Data-Mining bei der Verarbeitung von Sprache von drei Bereichen gesprochen:

- domänenspezifische Verarbeitung
- dokumentspezifische Verarbeitung
- sprachspezifische Verarbeitung

Hierbei handelt es sich um eine rein thematische, nicht chronologische Aufzählung.

UTF-8 (hex.)	Unicode Codepos.		Name	Beta-Code
e1bc80	U+1F00	ἀ	GREEK SMALL LETTER ALPHA WITH PSILI	a)
e1bc81	U+1F01	ἁ	GREEK SMALL LETTER ALPHA WITH DASIA	a(
e1bc82	U+1F02	ἂ	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA	a)\
e1bc83	U+1F03	ἃ	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA	a(\
e1bc84	U+1F04	ἄ	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA	a)/
e1bc85	U+1F05	ἅ	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA	a(/
e1bc86	U+1F06	ἆ	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI	a)=
e1bc87	U+1F07	ἇ	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI	a(=
e1bc88	U+1F08	Ἀ	GREEK CAPITAL LETTER ALPHA WITH PSILI	*)a
e1bc89	U+1F09	Ἁ	GREEK CAPITAL LETTER ALPHA WITH DASIA	*(a
e1bc8a	U+1F0A	Ἂ	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA	*)\a
e1bc8b	U+1F0B	Ἃ	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA	*(\a
e1bc8c	U+1F0C	Ἄ	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA	*)/a
e1bc8d	U+1F0D	Ἅ	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA	*(/a
e1bc8e	U+1F0E	Ἆ	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI	*)=a
e1bc8f	U+1F0F	Ἇ	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI	*(=a
e1bc90	U+1F10	ἐ	GREEK SMALL LETTER EPSILON WITH PSILI	e)
e1bc91	U+1F11	ἑ	GREEK SMALL LETTER EPSILON WITH DASIA	e(
e1bc92	U+1F12	ἒ	GREEK SMALL LETTER EPSILON WITH PSILI AND VARIA	e)\
e1bc93	U+1F13	ἓ	GREEK SMALL LETTER EPSILON WITH DASIA AND VARIA	e(\
e1bc94	U+1F14	ἔ	GREEK SMALL LETTER EPSILON WITH PSILI AND OXIA	e)/
e1bc95	U+1F15	ἕ	GREEK SMALL LETTER EPSILON WITH DASIA AND OXIA	e(/
e1bc98	U+1F18	Ἐ	GREEK CAPITAL LETTER EPSILON WITH PSILI	*)e
e1bc99	U+1F19	Ἑ	GREEK CAPITAL LETTER EPSILON WITH DASIA	*(e
e1bc9a	U+1F1A	Ἒ	GREEK CAPITAL LETTER EPSILON WITH PSILI AND VARIA	*)\e
e1bc9b	U+1F1B	Ἓ	GREEK CAPITAL LETTER EPSILON WITH DASIA AND VARIA	*(\e
e1bc9c	U+1F1C	Ἔ	GREEK CAPITAL LETTER EPSILON WITH PSILI AND OXIA	*)/e
e1bc9d	U+1F1D	Ἕ	GREEK CAPITAL LETTER EPSILON WITH DASIA AND OXIA	*(/e
e1bca0	U+1F20	ἠ	GREEK SMALL LETTER ETA WITH PSILI	h)
e1bca1	U+1F21	ἡ	GREEK SMALL LETTER ETA WITH DASIA	h(
e1bca2	U+1F22	ἢ	GREEK SMALL LETTER ETA WITH PSILI AND VARIA	h)\
e1bca3	U+1F23	ἣ	GREEK SMALL LETTER ETA WITH DASIA AND VARIA	h(\
e1bca4	U+1F24	ἤ	GREEK SMALL LETTER ETA WITH PSILI AND OXIA	h)/
e1bca5	U+1F25	ἥ	GREEK SMALL LETTER ETA WITH DASIA AND OXIA	h(/
e1bca6	U+1F26	ἦ	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI	h)=
e1bca7	U+1F27	ἧ	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI	h(=
e1bca8	U+1F28	Ἠ	GREEK CAPITAL LETTER ETA WITH PSILI	*)h
e1bca9	U+1F29	Ἡ	GREEK CAPITAL LETTER ETA WITH DASIA	*(h
e1bcaa	U+1F2A	Ἢ	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA	*)\h
e1bcab	U+1F2B	Ἣ	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA	*(\h

Tabelle 5. Auszug Beta Code Altgriechisch und die UTF-8-Entsprechung

Domänenspezifische Verarbeitung

Teilaufgabe	Erläuterung
Eigennamenextraktion	Erkennung von spezifischen Entitäten; meist auf der Basis manuell annotierter Datensätze. Hierbei sind nur die für die Domäne (das Korpus) typischen gemeint.[14]
Stoppwortliste erstellen	Eine Stoppwortliste ist eine Liste mit Begriffen, die bei der späteren Verarbeitung ausgenommen werden sollen (■ **Tabelle 4,** *siehe Seite 72*).[15]
Topic-Modellierung	Automatische Zuordnung von Begriffen zu Themen auf Basis von Worteigenschaften und Kontextinformationen.
Faktenextraktion	Vorher definierte Arten von Informationen werden durch die Verarbeitung modelliert. Viele Verfahren nutzen dafür die Abfolge unterschiedlicher Wörter in einem Satz.[16]
Relationsextraktion	Erkennung von Beziehungen zwischen Entitäten in einem Text.

Dokumentspezifische Verarbeitung

Teilaufgabe	Erläuterung
Metadaten erfassen	Metadaten, im Falle der Korpusanalyse z. B. Entstehungsort, Entstehungszeit, Autorenschaft, Editor, Editionszeit usw., sind bei der Textanalyse wertvolle Informationsquellen, um beispielsweise die Auswahl der zu verarbeitenden Daten einzugrenzen.
Bereinigung und Normalisierung	Abhängig davon, wie die Daten erfasst wurden, müssen sie vor der Analyse von allen irrelevanten Informationen, wie z. B. die für Auszeichnungssprachen üblichen Markup Tags, bereinigt werden. Eventuell abweichende Zeichenkodierungen, wie z. B. transkribierter altgriechischer Beta Code, müssen vor der Verarbeitung in eine einheitliche Zeichenkodierung konvertiert werden (■ **Tabelle 5**).

14 Zum Beispiel die im Bühnenstück von Shakespeare „KING HENRY the Fourth" abgekürzten „Speaker"-Segmente „North." und „West." sind Personenbezeichner, keine Himmelsrichtungen.

15 Solche Listen können sowohl domänenübergreifend, beispielsweise typisch für eine Sprache, als auch domänenspezifisch, beispielsweise typisch für eine Autorenschaft, sein. In eAQUA werden diese Listen auf Basis von Wortzählungen des Gesamtkorpus erstellt.

16 In eAQUA ist dies beispielsweise mit der Kookkurrenzanalyse vollzogen worden.

```
<pb n="62"/>
<p>VII— IX. ΜΑΘΗΜΑΤΙΚΑ.</p>
<p>11 l. [ VII 1] ΠΕΡΙ ΔΙΑΦΟΡΗΣ ΓΝΩΜΗΣ ἢ ΠΕΡΙ ψΑΥΣΙΟΣ ΚΥΚΛΟΥ ΚΑΙ
   <note type="marginal">390</note>
   ΣΦΑΙΡΗΣ.</p>
<p>11 m. [ VII 2] ΠΕΡΙ ΓΕΩΜΕΤΡΙΗΣ. Vgl. B 155.</p>
<lb n="5"/> <p>11 n. [ VII 3] ΓΕΩΜΕΤΡΙΚΩΝ &#x003C;A&#x772;B&#x772;?&#x003E;</p>
<p> o. [ VII 4] ΑΡΙΘΜΟΙ.</p>
<p>11p. [VIII 1] ΠΕΡΙ ΑΛΟΓΩΝ ΓΡΑΜΜΩΝ ΚΑ| ΝΑΣΤΩΝ</p>
<p>11 q. [ VIII 2] ΕΚΠΕΤΑΣΜΑΤΑ.</p>
<p>11r. [ VIII 3] ΜΕΓΑΣ ΕΝΙΑΥΤΟΣ ἡ ΑΣΤΡΟΝΟΜΙΗ. ΠΑΡΑΠΗΓΜΑ. Vgl.</p>
<lb n="10"/> <p>B 14, 5; 15 a. Diog. v 43 Theophrasts Schrift Περὶ τῆς Δημοκρίτου
   ἀστρολογίας ᾱ.</p>
<p>12. Censor. 18, 8 est et Philolai annus [32 A. 22] . . . et Demooriti
   ex annis LXXXII cum intercalariis [nämlich mensibus] perinde
   Kallippos] viginti octo.</p>
<lb n="15"/> <p>13. Apollon. de pronom. p. 65, 15 Schneid. καὶ Φερεκύδης ἐν τῇι
   Θεολογίαι καὶ ἔτι Δ ἐν τοῖς Περὶ ἀστρονομίας καὶ ἐν τοῖς
   ὑπολειπομένοις συντάγμασι συνεχέστερον χρῶνται τῇι ἐμεῦ
   καὶ ἔτι τῇι ἐμέο. Vgl. B 29 a.</p>
<p>VII— IX. MATHEMATISCHE</p>
<p>11 r. [ VIII 3] WELTJAHR oder ASTRONOMIE SAMT STECKKALENDER.</p>
<p>12. Das Weltjahr Demokrits besteht aus 82 gewöhnlichen Jahren
   28 Schaltmonaten.</p>
<p>13. Meiner [ kontrahierte und unkontrahierte Form].</p>
<note type="footnote">2 ΓΝΩΜΗΣ] ΓΝΩΜΟΝΟΣ Cobet. Allmann Hennathena IV 206 meint,
   durch die Differenz des Gnomon sei er auf die Anfänge der Infinitesimalmethode
   worden, da die Atomistik der 'schen Monadenlehre verwandt sei:
   ΓωΝιΗο verm. Gromperz; die Überlieferung haltend übersetzt Über Verschiedenheit
   der Auffassung oder über Kreis- und Kugelberührung H. Vgl. Bibl.
   III. F., X (1910) 146. Er sieht darin eine Polemik gegen ' Angriff auf
   die Geometer [74 B 7] 7 Über verhältnislose (nicht irrationelle) Linien
   Atome erklärt H. Vgl a. O. 147; Hultsch verm. κλαστῶν Jahrb. f kl. Phil.
   579 8 vgl. Ptol. geogr. vii 7 ὑπογραφὴ τοῦ ἐκπετάσματος. ὑπογραφὴ δ' ἔσται
   καὶ τῆς τοιαύτης ἐκπετάσεως ἁρμόζουσά τε καὶ κεφαλαιώδης. ἡ τοιαύτη τῆς κρικωτῆς
   σφαίρας ἐπιπέδωι καταγραφή κτλ. Also Projektion der Armillarsphäre
   die Ebene 9 ΠΑΡΑΠΗΓΜΑ] »Steckkalender«, ein ehernes oder marmornes
   Verzeichnis der Tage des Sonnenjahres nach dem Zodiakus nebst den üblichen
   Episemasien ( Wettexeichen). Neben den Tagen befanden sich Löcher, in die
   Tage des bürgerlichen Monats eingesteckt werden konnten, S. *
   aus Milet, Berl. Sitz. B. 1904, 92 ; 266; Heron de Villefosse Comt. Rend.
   de l' Ac. des Inscript. 1898 p, 267. Das Parapegma des Meton und Euktemon
   (27. Juni 432) zeigt bereits genau die Einrichtung des Demokritischen</note>

<pb n="63"/>
<p>14. ÜBERRESTE DES PARAPEGMA DER ΑΣΤΡΟΝΟΜΙΗ.</p>
```

Abbildung 45. Spracherkennung bei Mehrsprachigkeit: Der griechische Text ist mit deutschen Kommentaren und Überschriften versehen. Einzelne Passagen oder Quellenangaben können auch lateinisch sein. XML-Auszug aus: Die Fragmente der Vorsokratiker, Berlin 1912. S. 62f. Open Greek and Latin Project.
URL: https://github.com/OpenGreekAndLatin/fragmentary-dev/blob/master/fragmenteVorsokratiker_2.xml

Sprachspezifische Verarbeitung

Teilaufgabe	Erläuterung
Spracherkennung	Die verwendeten Sprachen werden ermittelt (■ **Abbildung 45**).[17]
Segmentierung	Strukturiert den Text in einzelne Teile, die separat untersucht werden können. Üblich ist die Segmentierung in Sätze anhand der Satzzeichen.
Tokenisierung	Segmentiert auf der Basis der Wortebene in einzelne Teile (Token), indem beispielsweise das Leerzeichen als Wortgrenze aufgefasst wird.
Wortstammreduktion	Die Wörter werden auf ihren Wortstamm zurückgeführt, um bei einer späteren Suche auch Flexionen zu finden.
Lemmatisierung	Die Grundform eines Wortes (Lemma) wird gebildet.
Part-of-Speech Tagging	Zuordnung von Wörtern und Satzzeichen in Wortarten.
Parsing	Der Text wird in eine neue syntaktische Struktur überführt. Dabei ist für den Parser ein Token die atomare Eingabeeinheit.
Koreferenz (Referenzidentität) auflösen	Eine Koreferenz liegt vor, wenn sich innerhalb einer Äußerung zwei sprachliche Ausdrücke auf das gleiche linguistische Objekt beziehen, beispielsweise mittels Verwendung von Pronomen.
Eigennamenextraktion	Bei der Eigennamenerkennung, auch Named Entity Recognition (NER), werden die Begriffe eines Textes bestimmten Typen (z. B. Ort oder Person) zugeordnet.

17 Wenn diese in den Metadaten nicht annotiert sind, ist dies, gerade bei multilingualen Texten, ein nichttriviales Problem, welches häufig durch sprachspezifische (Stich-)Wortlisten gelöst wird.

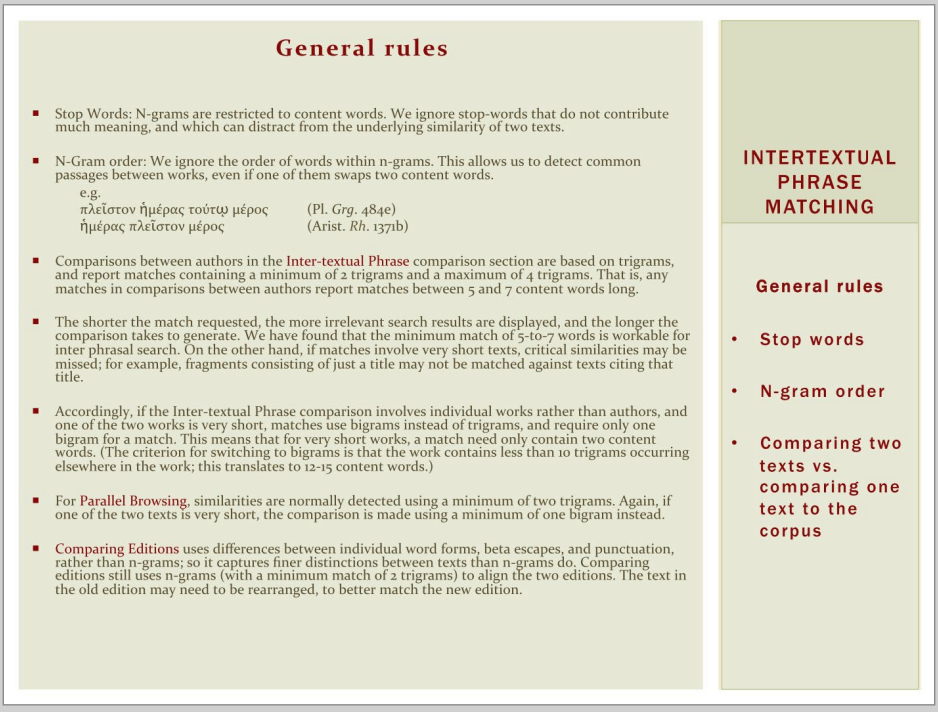

Abbildung 46. Regeln des Inter-textual Phrase-Matching beim TLG-Online[18]

18 URL: http://stephanus.tlg.uci.edu/help.php bzw. http://stephanus.tlg.uci.edu/helppdf/ngrams.pdf.

Parallelstelle, Zitat, Paraphrase, Kookkurrenz

Eine der verständlichsten, wenn auch häufig kontrovers diskutierten Methoden innerhalb der als Digital Humanities bezeichneten Teildisziplin klassischer Wissenschaftszweige ist die Parallelstellen- oder auch Zitationsanalyse. Genaugenommen ist noch nicht einmal die Frage geklärt, was ein Zitat denn letztlich ausmacht: Ist es der genaue Wortlaut oder reicht bereits eine synonyme Umschreibung des Inhalts? Diese eher begriffliche Betrachtungsweise soll nicht Gegenstand der Überlegungen sein. Wir wollen lediglich den Blick auf technische Teilaufgaben bei der Verarbeitung von Sprache richten.

Die Zitationsanalyse beschäftigt sich als Teilgebiet der Bibliometrie mit der qualitativen Untersuchung von zitierten und zitierenden Arbeiten. Im Ergebnis werden Regelmäßigkeiten und Strukturen eines Autors oder einer Autorengruppe aufgezeigt, im ungünstigen Fall führt es zu Plagiatsvorwürfen oder gar zur Aberkennung eines akademischen Grades, wenn in der Abschlussarbeit entsprechende Zitate als solche nicht kenntlich gemacht werden.

Suche über direkte Nachbarn – Nachbarschaftskookkurrenzen

Bei Zitationsanalysen, genaugenommen bei Ähnlichkeitsbestimmungen von Teiltexten, kommen sogenannte String-Matching-Algorithmen häufig zum Einsatz. Dies sind Verfahren, die unter Definition bestimmter Toleranzkriterien nach exakten Übereinstimmungen innerhalb von Texten suchen. Die Art und Weise, wie innerhalb der Zeichenketten (Strings) nach Treffern (Matches) gesucht wird, ist mitunter verschieden, im Ergebnis wird meist aufgrund von Häufigkeiten und Bewertungsmaßstäben (Signifikanzkriterien) geurteilt (■ **Abbildung 46**).

Eine relativ simple Möglichkeit besteht darin, die zu untersuchenden Texte auf wesentliche Terme zu reduzieren, indem beispielsweise häufig benutzte Worte, im Jargon auch Stoppworte (engl. Stopwords) genannt, zusammen mit den Satzzeichen herausgerechnet werden und das so reduzierte Gesamtkorpus auf Nachbarschaftskookkurrenzen, also das gemeinsame Auftreten von Wortgruppen, hin untersucht wird. Wenn anfänglich von einer simplen Möglichkeit gesprochen wurde, dann deshalb, weil hier gewisse linguistische Feinheiten außer Acht gelassen werden und nur der genaue Wortlaut ausschlaggebend ist. Beispielsweise können Synonyme oder Reflexivpronomen dabei nicht berücksichtigt werden. Ausschlaggebend ist die genaue Abfolge der Terme, wobei eines der schwierigsten Probleme dabei ist, innerhalb des Gesamtkontextes sowohl den Anfang als auch das Ende einer Parallelstelle zu finden. Es wird in dem Zusammenhang auch von „gierigen" Suchausdrücken gesprochen, die im Ergebnis eine größere Fundstelle liefern, als tatsächlich vorhanden.

$$sim = \frac{n_{ab} \times 2}{n_a + n_b}$$

Formel 1. Similar-Text.

Hier geben n_a und n_b jeweils die Länge der jeweiligen Zeichenketten und n_{ab} die Anzahl identischer Zeichen, also die Differenz zur Levenshtein-Distanz,[19] an.

$$sim = \frac{\big(max(n_a, n_b) - lev(a,b)\big) \times 2}{n_a + n_b}$$

Formel 2. Similar-Text mit Angabe der Levenshtein-Distanz

Beispiel: Similar-Text			$sim = \frac{(max(n_a, n_b) - lev(a,b)) \times 2}{n_a + n_b}$		
n_a	n_b	$lev(a,b)$	$max(n_a, n_b) - lev(a,b)$	sim	Similar-Text
a = Beispieltext 1 b = Beispiel Text 2					
14	15	3	12	$\frac{12 \times 2}{14 + 15} = \frac{24}{29}$	0,83
Zeichenkette a = The quick brown fox jumps over the lazy dog Zeichenkette b = The fox jumps over the lazy dog					
43	31	12	31	$\frac{31 \times 2}{43 + 31} = \frac{62}{74}$	0,84

Abbildung 47. Beispielberechnung Similar-Text

19 Eine von dem russischen Mathematiker Vladimir I. Levenshtein 1965 eingeführte Methode, zwei Zeichenketten zu vergleichen, indem die minimale Anzahl von Einfüge-, Lösch- und Ersetz-Operationen gezählt wird, um die eine in die andere umzuwandeln.

Bewertung der Übereinstimmung von Parallelstellen

Wenn die Grenzen der (möglichen) Parallelstelle gefunden sind, muss der Grad der Übereinstimmung bewertet werden. Eine unkonventionelle Berechnungsmethode basiert auf der Editierdistanz oder auch Levenshtein-Distanz.[20] Sie besagt, wie viele Einfüge-, Lösch- und Ersetzoperationen notwendig sind, um eine Zeichenkette in eine andere zu verwandeln.

Der Vorteil eines solchen Signifikanzmaßes ist, dass es sich um einen absoluten Wert zwischen 0 und 1 (bzw. wahlweise eines Prozentwertes 0–100) handelt. Nachteilig bemerkbar macht sich hier die unterschiedliche Länge der zu untersuchenden Zeichenketten. Je größer der Längenunterschied, umso weniger signifikant ist die Parallelstelle, selbst bei exakter Übereinstimmung.

Die Parallelstellen werden in eAQUA schlussendlich unter Verwendung der Editierdistanz mit einem Similaritätswert belegt, der zwischen 0 = nicht identisch und 1 = vollständig identisch liegt. Berechnet wird nach einem Algorithmus Similar-Text, der bei Ian Oliver[21] mittels eines Pseudo-Codes beschrieben ist (■ **Formel 1**, ■ **Formel 2**).

Die berechneten Similaritätswerte beziehen sich immer auf die komplett tokenisierten Segmente, nicht allein nur auf die Suchmaske. Dies führt dazu, dass auch komplett identische Passagen mit einem von 1 abweichenden Wert belegt werden können, wenn sie innerhalb eines größeren Segments benutzt werden. Im zweiten Beispiel Similar-Text ergeben sich die Abweichungen lediglich durch den Einschub quick brown (■ **Abbildung 47**).

Similar-Text-Berechnungen sind nur bei kurzen Segmenten, wie der Satz-Tokenisierung in eAQUA, sinnvoll, da die Werte mit der Länge der untersuchten Segmente tendenziell abnehmen.

20 Vladimir I. Levenshtein: Binary codes capable of correcting deletions, insertions, and reversals. In: Doklady Akademii Nauk SSSR. Band 163, Nr. 4, 1965, S. 845–848. Englische Übersetzung: Soviet Physics Doklady, 10(8), 1966, S. 707–710.

21 Ian Oliver: Programming Classics: Implementing the World's Best Algorithms. Prentice Hall PTR New York, 1993.

Abbildung 48. N-Gramm basierte Suche im TLG-Online[22]

22 URL: http://stephanus.tlg.uci.edu.

n-Gramm basierte Suche

Eine Variante, die Suche nach Parallelstellen über einfache Vergleiche von Elementen der Texte zu realisieren, ist, die Wörter in eine Aneinanderreihung von Buchstabenkombinationen zu zerlegen – es wird hier von der Bildung von n-Grammen gesprochen. Das n steht für eine zu definierende Zahl. Gebräuchlich bei der Textverarbeitung sind Unigramm, Bigramm und Trigramm (■ **Abbildung 48**).

Das Verfahren für eine Suche würde exemplarisch wie folgt aussehen:

- Häufig benutzte Wörter (Stoppwörter) werden aus den Texten herausgerechnet.
- Beide Texte werden in einzelne Sequenzen zerlegt. Üblich können zum Beispiel ganze Sätze sein. Es ließen sich aber auch kleinere Einheiten bilden, indem Segmente mit fünf aufeinander folgenden Wörtern gebildet werden, unabhängig von Satzzeichen, die zuvor entfernt wurden.
- Die Sequenzen werden in einzelne Terme (Token) gesplittet.
- Für jeden Term werden Trigramme gebildet. Ist dieser zu kurz, bleibt er so, wie er ist, stehen.
- Die n-Gramme mit maximaler Größe von drei des einen Segments werden auf das Vorhandensein im zweiten Segment hin untersucht.
- Ist innerhalb eines Terms ein n-Gramm gefunden, wird dieser Term als Fundstelle markiert und der nächste Term genommen.
- Werden innerhalb des Segmentes insgesamt vier Treffer erzielt, gilt das Segment als Parallelstelle.
- Die Reihenfolge einzelner Fundstellen innerhalb der zu überprüfenden Segmente spielt keine Rolle.
- Anschließend wird die Ähnlichkeit beider Segmente bewertet, indem mittels Editierdistanz nach Levenshtein ein Prozentwert ermittelt wird.

Der Vorteil dieser Variante der Annäherung an die Parallelstellen ist, dass automatisch einige, mitunter aufwendige Verfahren der Normalisierung des Textes wegfallen. So kann, zumindest der Theorie nach, auf Stammwortreduktion oder Lemmatisierung verzichtet werden, da durch die Bildung von Trigrammen automatisch der Wortstamm erfasst würde. Denn vielfach werden Wörter nach einem einfachen Schema [Präfix] + Wortstamm + [Suffix] gebildet. Wobei Prä- oder Suffix optional sind. Deswegen sind sie in eckige Klammern gesetzt (in der IT insbesondere beim Programmieren werden optionale Parameter so gekennzeichnet). Ein Nachteil dieses Vorgehens wird aber auch sofort klar. Trigramme können auch Prä- oder Suffixe sein, so würde in der deutschen Sprache allein die Wortendung „ung“ zu einem Treffer führen, so wie beispielsweise in lateinischen Texten das Suffix *ion*, wodurch eine

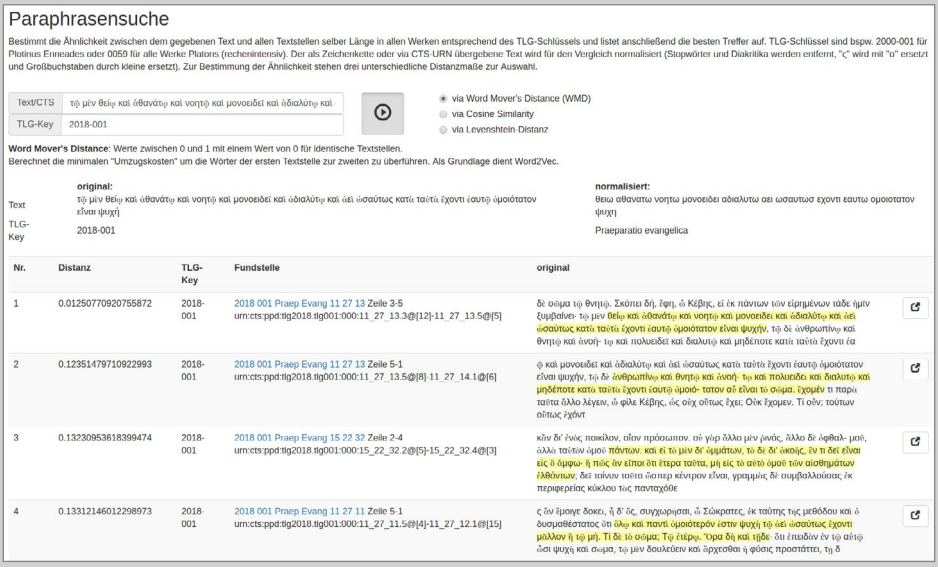

Abbildung 49. Paraphrasensuche mit der Word Mover's Distance

erheblich längere Liste von möglichen Parallelstellen zustande kommt, die über eine Similaritätsberechnung bewertet und abschließend reduziert werden muss.

Der Ansatz einer n-Gramm basierten Suche ist als ein erster halbwegs praktischer Versuch anzusehen, ohne Beachtung und Kenntnis der benutzten Sprache, rein über linguistische Statistik und ohne semantische Interpretation, zu Ergebnissen zu gelangen.

Vektorenbasierte Vergleiche

Einen weiteren Ansatz, auf computerlinguistische, und damit sprachspezifische, Vorverarbeitung weitgehend zu verzichten, bilden Verfahren auf Basis der distributionellen Hypothese.[23] Dabei wird von der Prämisse ausgegangen, dass Wörter, die in einem ähnlichen semantischen Kontext benutzt werden, auch eine ähnliche Bedeutung besitzen müssen. Die distributionelle Semantik repräsentiert Wortbedeutung mittels sogenannter Kontextvektoren, die eine statistische Verteilung eines Wortes über relevante sprachliche Kontexte erfassen. Mittels Verfahren aus der linearen Algebra können aus den Kontextvektoren semantische Ähnlichkeiten einzelner Wörter oder gar die Bedeutung komplexer Phrasen berechnet werden.

In dem durch die Volkswagen Stiftung geförderten Verbundprojekt „Digital Plato“,[24] welches sich der Untersuchung der Rezeption und Nachwirkung des platonischen Werkes in der griechischen Literatur bis zur Spätantike widmet und in dem einer der Projektpartner die Alte Geschichte der Universität Leipzig ist, werden beispielsweise mit Hilfe der Word Mover's Distance alle Textstellen eines ausgewählten Werkes mit der zu suchenden Passage verglichen und die ähnlichsten Treffer ausgegeben[25] (■ **Abbildung 49**).

23 Zellig S. Harris (1954): Distributional Structure, WORD, 10:2-3, 146–162, DOI: https://doi.org/10.1080/00437956.1954.11659520.

24 URL: https://digital-plato.org/.

25 Marcus Pöckelmann, Jörg Ritter, Eva Wöckener-Gade, Charlotte Schubert: Paraphrasensuche mittels word2vec und der Word Mover's Distance im Altgriechischen. In: Digital Classics Online, DCO 3,3 (2017), S. 24–36. DOI: https://doi.org/10.11588/dco.2017.0.40185.

Korpus	Anzahl Kookkurrenzen	Kookkurrenzen freq = 1	in Prozent
BTL[26]	137.486.214	110,876,836	80,65
MPL[27]	580.247.568	398.935.822	68,75
Perseus Shakespeare[28]	6.746.602	5.027.170	74,51
TLG[29]	355.021.014	258.961.566	72,94

Tabelle 6. Gesamtmenge von Kookkurrenzen diverser Korpora im Verhältnis zur Menge mit der Häufigkeit 1

26 Bibliotheca Teubneriana Latina, Online-Version, Stand Februar 2014.
27 Patrologia Latina Database, CD-ROM Version, November 1995c.
28 William Shakespeare in Perseus Digital Library, Renaissance Materials, Stand Mai 2013.
29 TLG-E, CD-ROM Version aus dem Jahre 1999.

Signifikanzmaße bei der Beurteilung von Kookkurrenzen

In der Statistik wird unter Signifikanz eine Kennzahl verstanden, welche die Wahrscheinlichkeit eines systematischen Zusammenhangs zwischen Variablen, im Falle von Textanalysen also zwischen Teiltexten (z. B. Wörtern), bezeichnet. Die Signifikanz drückt aus, ob ein scheinbarer Zusammenhang rein zufälliger Natur sein könnte oder mit hoher Wahrscheinlichkeit tatsächlich vorliegt.

Zur Berechnung werden abhängig vom Untersuchungsgegenstand unterschiedliche Formeln herangezogen, welche in erster Linie aus der Computerlinguistik stammen. Die Signifikanzmaße sollen dabei helfen, wichtige von unwichtigen Kookkurrenzen zu trennen. Dabei werden statistische Kenngrößen wie Korpusgröße, Häufigkeit der einzelnen Wörter oder Frequenz des gemeinsamen Auftretens ins Verhältnis gesetzt.

Eines der einfachsten Signifikanzmaße ist eine frequenzsortierte Kookkurrenzliste, die die Häufigkeit des gemeinsamen Auftretens zweier Worte im Gesamtkorpus angibt. Ein Nachteil frequenzsortierter Listen ist, dass nach dem Zipfschen Gesetz, dem Beginn der quantitativen Linguistik, sehr viele Wörter sehr selten auftreten. Demzufolge lassen sich mit einem Schwellenwert größer 1, also dem mehrmaligen gemeinsamen Auftreten eines Wortpaares, etwa zwei Drittel der Kookkurrenzen herausfiltern. Berechnet von den eAQUA-Tools sieht dies für ausgewählte Korpora wie in nebenstehender Tabelle aus (■ **Tabelle 6**).

Wie aus der kleinen Übersicht zu erkennen ist, sind ein Großteil der gefundenen Kookkurrenzen eher als niedrigfrequent zu bezeichnen. Um daraus die wichtigen zu filtern, sind Berechnungsmethoden erforderlich, von denen hier einige vorgestellt werden.

$$dice_{ab} = \frac{2 \times n_{ab}}{n_a + n_b}$$

Formel 3. Dice

Beispiel: Dice	$dice_{ab} = \frac{2 \times n_{ab}}{n_a + n_b}$
Bigramm	Trigramm
a = Tür b = Tor	
$a = \{\S T\ T\text{ü}\ \text{ür}\ r\S\}$ $b = \{\S T\ To\ or\ r\S\}$ $d_{Tür,Tor} = \frac{2 \times 2}{4 \times 4} = \frac{4}{8} = 0{,}5$	$a = \{\S\S T\ \S T\text{ü}\ Tür\ \text{ür}\S\ r\S\S\}$ $b = \{\S\S T\ \S To\ Tor\ or\S\ r\S\S\}$ $d_{Tür,Tor} = \frac{2 \times 2}{5 + 5} = \frac{4}{10} = 0{,}4$
a = Spiegel b = Spargel	
$a = \{\S S\ Sp\ pi\ ie\ eg\ ge\ el\ l\S\}$ $b = \{\S S\ Sp\ pa\ ar\ rg\ ge\ el\ l\S\}$ $d_{Spiegel,Spargel} = \frac{2 \times 5}{8 + 8} = \frac{10}{16} = 0{,}625$	$a = \{\S\S S\ \S Sp\ Spi\ pie\ ieg\ ege\ gel\ el\S\ l\S\S\}$ $b = \{\S\S S\ \S Sp\ Spa\ par\ \arg rge\ gel\ el\S\ l\S\S\}$ $d_{Spiegel,Spargel} = \frac{2 \times 5}{9 + 9} = \frac{10}{18} \approx 0{,}556$

Abbildung 50. Beispielberechnung Dice

Dice

Beim Dice-Koeffizienten[30] wird die Ähnlichkeit zweier Terme mittels einer Zahl zwischen 0 und 1 angegeben. Berechnungsgrundlage sind sogenannte n-Gramme. Ermittelt wird die Anzahl der n-Gramme, die in beiden Termen vorhanden sind, um diese ins Verhältnis zur Gesamtzahl der n-Gramme zu setzen.

Berechnet wird nach der nebenstehenden Formel (■ **Formel 3**), wobei n_{ab} die Schnittmenge beider Terme und n_a bzw. n_b die Anzahl der gebildeten n-Gramme pro Term angibt (■ **Abbildung 50**).

Bei der Bewertung von Kookkurrenzen kann der Dice-Koeffizient genutzt werden, indem die Häufigkeiten (Frequenzen) der Wörter ins Verhältnis gesetzt werden. n_a und n_b sind dabei die Frequenzen der Terme, n_{ab} die Anzahl des gemeinsamen Auftretens.

Aus der angeführten Formel ergeben sich relativ einfache Bewertungsmaßstäbe. Je frequenter die beiden Begriffe gemeinsam benutzt werden, umso mehr nähert sich der Wert 1. Treten beide Begriffe nur gemeinsam auf, wird die höchste Signifikanz mit 1 erreicht. Wie oft diese Kookkurrenz im Korpus zu finden ist, spielt dabei keine Rolle. Daraus ergibt sich eine wichtige Eigenschaft des Dice-Koeffizienten: Kookkurrenzen, die selten zusammen auftreten, bei denen ein Wort hoch- und das andere niedrigfrequent sind, werden als unsignifikant bewertet.

30 Auch als Sørensen-Dice-Koeffizient bezeichnet, benannt nach den Botanikern Thorvald Sørensen und Lee Raymond Dice.

$$jaccard_{ab} = \frac{n_{ab}}{n_a + n_b - n_{ab}}$$

Formel 4. Berechnung Jaccard-Koeffizient

Beispiel: Jaccard	$jaccard_{ab} = \frac{n_{ab}}{n_a + n_b - n_{ab}}$
Bigramm	Trigramm
a = Tür b = Tor	
$a = \{\S T\ T\text{ü}\ \text{ür}\ r\S\}$ $b = \{\S T\ To\ or\ r\S\}$ $j_{Tür,Tor} = \frac{2}{4+4-2} = \frac{2}{6} \approx 0{,}334$	$a = \{\S\S T\ \S T\text{ü}\ Tür\ \text{ür}\S\ r\S\S\}$ $b = \{\S\S T\ \S To\ Tor\ or\S\ r\S\S\}$ $j_{Tür,Tor} = \frac{2}{5+5-2} = \frac{2}{8} = 0{,}25$
a = Spiegel b = Spargel	
$a = \{\S S\ Sp\ pi\ ie\ eg\ ge\ el\ l\S\}$ $b = \{\S S\ Sp\ pa\ ar\ rg\ ge\ el\ l\S\}$ $j_{Spiegel,Spargel} = \frac{5}{8+8-5} = \frac{5}{11} \approx 0{,}455$	$a = \{\S\S S\ \S Sp\ Spi\ pie\ ieg\ ege\ gel\ el\S\ l\S\S\}$ $b = \{\S\S S\ \S Sp\ Spa\ par\ \text{arg}\ rge\ gel\ el\S\ l\S\S\}$ $j_{Spiegel,Spargel} = \frac{5}{9+9-5} = \frac{5}{13} \approx 0{,}385$

Abbildung 51. Beispielberechnung für den Jaccard-Koeffizienten

n_a	n_b	n_{ab}	Dice	Jaccard
100	100	1	0,01	0,005
100	100	10	0,1	0,05
100	100	50	0,5	0,33
100	100	90	0,9	0,82
100	100	100	1	1

Tabelle 7. Vergleich Dice- und Jaccard-Koeffizient bei 100 n-Grammen und verschiedenen Schnittmengen

Jaccard

Beim Jaccard-Koeffizienten (nach dem Botaniker Paul Jaccard) wird die Ähnlichkeit zweier Terme mittels einer Zahl zwischen 0 und 1 angegeben. Berechnungsgrundlage bei Text Mining-Verfahren sind sogenannte n-Gramme. Bei n-Grammen wird ein Term bzw. ein Text in gleich große Teile zerlegt. Diese Fragmente können Buchstaben, Phoneme, ganze Wörter oder ähnliches sein.

Ermittelt wird die Anzahl der n-Gramme, die in beiden Termen vorhanden sind, um diese ins Verhältnis zur Gesamtzahl der n-Gramme zu setzen (■ **Formel 4**). Berechnet wird nach der nebenstehenden Formel, wobei n_{ab} die Schnittmenge beider Terme und n_a bzw. n_b die Anzahl der gebildeten n-Gramme pro Term angibt (■ **Abbildung 51**).

Für die Bewertung von Kookkurrenzen gilt beim Jaccard-Koeffizienten ähnliches wie beim Dice-Koeffizienten. Beide berechnen den Signifikanzwert ähnlich, die relative Ordnung der Kookkurrenzen bleibt gleich, nur der absolute Signifikanzwert unterscheidet sich marginal.

Eine Modell-Berechnung mit mittlerer Frequenz von 100 ist in nebenstehender Tabelle vorgenommen (■ **Tabelle 7**). Bei einer Schnittmenge von 50 übereinstimmenden n-Grammen bei gleicher Länge der Ausdrücke von 100 n-Grammen wird der Dice-Koeffizient mit ½, der Jaccard-Koeffizient dagegen mit ⅓ errechnet.

$$poisson_{n,k} = \frac{1}{k!}\gamma^k \times e^{-\gamma}$$

Formel 5. Poisson-Verteilung

$$poisson\left(n_a, n_b, k, n\right) = \frac{k \times \left(\log k - \log \gamma - 1\right)}{\log n}$$

Formel 6. Poisson-Maß

$$\gamma = \frac{n_a \times n_b}{n}$$

Formel 7. Grundannahme vor der Umstellung

$$poisson = \frac{n_{ab} \times \log \frac{n_{ab} \times n}{n_a \times n_b} - n_{ab}}{\log n}$$

Formel 8. Berechnung Poisson-Maß

Poisson

Ein Ansatz zur Berechnung von signifikanten Kookkurrenzen basiert auf der Poisson-Verteilung,[31] einer diskreten Wahrscheinlichkeitsverteilung (■ **Formel 5**).

Auf der Basis der Poisson-Verteilung geben Quasthoff / Wolff[32] das Poisson-Maß mit der nebenstehenden Formel an, welche beispielsweise für Berechnungen von Korpora im Wortschatz-Portal[33] genutzt wurde, und in der die zwei Faktoren n (Anzahl der Sätze im Korpus) und k (Häufigkeit des gemeinsamen Auftretens, auch n_{ab} bezeichnet) maßgeblich sind (■ **Formel 6**).

Nach einer Umstellung und der Grundannahme ergibt sich schlussendlich die Berechnung (■ **Formel 7**, ■ **Formel 8**).

Somit ließe sich das Poisson-Maß auf die Differenz zwischen Local Mutual Information und Frequenz reduzieren.

31 Benannt nach dem Mathematiker Siméon Denis Poisson.

32 Uwe Quasthoff, Christian Wolff. The Poisson Collocation Measure and its Applications. In Second International Workshop on Computational Approaches to Collocations, 2002. URN: http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:bvb:355-epub-68241. Vgl. ebenso: Gerhard Heyer, Uwe Quasthoff, Thomas Wittig. Text Mining: Wissensrohstoff Text: Konzepte, Algorithmen, Ergebnisse. Herdecke; Bochum: W3L-Verl. 2006. S. 338 ff.

33 URL: http://wortschatz.uni-leipzig.de/de.

$$p(K=k) = p^k (1-p)^{n-k} \binom{n}{k}$$

Formel 9. Binomialverteilung[34]

$$\begin{aligned} -2\log\lambda = &\left[\log L(p_1,k_1,n_1) + \log L(p_2,k_2,n_2)\right. \\ &\left. -\log L(p_1,k_1,n_1) - \log L(p_2,k_2,n_2)\right] \end{aligned}$$

Formel 10. Log likelihood[35]

$$\log L(p,n,k) = k\log p + (n-k)\log(1-p)$$

Formel 11. Log likelihood Voraussetzung[36]

34 Dunning, S. 64.
35 Dunning, S. 67.
36 Ebd.

Log-Likelihood

Eines der populärsten Signifikanzmaße bei der Analyse großer Textkorpora ist nach Dunning[37] das Log-Likelihood-Maß, welches auf der Binomialverteilung, einer der wichtigsten diskreten Wahrscheinlichkeitsverteilungen, basiert (■ **Formel 9**).

Dunning kommt schließlich bei der Berechnung von log likelihood zu der Formel unter der Log-Likelihood Voraussetzung (■ **Formel 10,** ■ **Formel 11**).

Charakteristisch für das Log-Likelihood-Maß ist, im Gegensatz beispielsweise zum Poisson-Maß, die Gleichbehandlung von signifikant häufigen und signifikant seltenen Ereignissen. So finden sich in den Digitalisaten vom TLG in der Version TLG-E bei rund 73,8 Millionen Wörtern etwa 1,3 Millionen Kookkurrenzen, die nur einmal auftreten und trotzdem mit einem lgl-Wert von 30 und ein wenig mehr belegt sind. Einen ähnlich großen Wert von 34,553 haben zum Beispiel καί und τό, die zusammen 14311 Mal gezählt wurden.

37 Ted Dunning: „Accurate Methods for the Statistics of Surprise and Coincidenc". In: Computational Linguistics 19, 1 (1993), 61–74. URL: http://aclweb.org/anthology/J/J93/J93-1003.pdf.