Statistische Aspekte von Suffixbäumen natürlichsprachiger Texte

Felix Golcher

Abschlussarbeit, vorgelegt am CIS
Centrum für Informations- und Sprachverarbeitung
Ludwig Maximilan Universität München

24. Februar 2005

Meiner Schwester Henriette

Für Diskussionen und Anregungen, ohne die meine Arbeit nicht in dieser Form zustandegekommen wäre, danke ich Herrn Prof. Schulz, Sebastian Nagel, Johannes Goller, Karsten Tabelow, Johannes Stiehler, Aleksandra Wasiak und Maryia Vitusevic.

Mein Dank geht auch an Imre Sueveges und Tomaz Jäger für ihre Hilfe bei der Auswahl der finnischen und ungarischen Korpora.

Ganz spezieller Dank gebührt den WissenschaftlerInnen, die Korpora wie die hier verwendeten zusammenstellen und sie der wissenschaftlichen Öffentlichkeit frei zur Verfügung stellen.

Inhaltsverzeichnis

1 Einleitung

In der vorliegenden Arbeit stelle ich eine sprachstatistische Größe vor, die in sehr unterschiedlichen natürlichen Sprachen ein überraschend uniformes Verhalten zeigt. In nicht natürlichsprachigen Texten tritt dieses Phänomen dagegen nicht auf.

Die untersuchte Größe basiert auf dem Begriff des Suffixbaumes: Der Suffixbaum eines Textes ist eine baumartige Indexstruktur, die gewöhnlich für sehr schnelle Suchverfahren in großen Texten verwendet wird. Daneben aber erlaubt ein Suffixbaum unmittelbaren und vollständigen Zugriff auf die Information, welche Zeichenketten im zugrundeliegenden Text vorkommen und wie sie mit anderen Zeichenketten kombiniert werden.

Die Zahl der Verzweigungen im Suffixbaum gibt Auskunft über das Ausmaß an Wiederholungen im zugrundeliegenden Text: Viele Verzweigungen stehen für viele Wiederholungen, wenige Verzweigungen für wenige Wiederholungen.

Ich untersuche in dieser Arbeit die Zahl der Verzweigungen in Suffixbäumen bezogen auf die Länge des eingelesenen Textes. Diese Größe bezeichne ich mit V. Die zentrale Aussage der vorliegenden Arbeit ist die Beobachtung, dass V für natürlichsprachige Texte eine Konstante darstellt. Ihr Wert schwankt von Sprache zu Sprache nur unerheblich und ist auch von der Textlänge im Wesentlichen unabhängig.

Ich verwende Daten aus insgesamt 21 Sprachen, die vier verschiedenen Sprachfamilien angehören und mit einem breiten Spektrum an qualitativ unterschiedlichen Schriftsystemen geschrieben werden.

Das gleichförmige Verhalten der Größe V vergleiche ich mit dem Zipfschen Gesetz, einem schon seit langem bekannten sprachstatistischen Phänomen, das sich in allen natürlichsprachigen Texten ausreichender Länge findet. Der Vergleich gibt Einblick in die Bedeutung der beiden Phänomene und begegnet dem Verdacht, dass die Konstanz von V lediglich eine neue Manifestation des Zipfschen Gesetzes ist.

Im Verlauf der Arbeit wird V auch für verschiedene Klassen nicht natürlichsprachiger Texte untersucht: Für rein zufällig erstellte Texte, für Programmcode und für den Output eines evolutionär arbeitenden Computerprogramms, mit dem sich Texte mit bestimmten statistischen Merkmalen erzeugen lassen. Es ergeben sich jeweils eindeutige qualitative und quantitative Unterschiede zwischen dem Verhalten von V für natürlichsprachige und nicht natürlichsprachige Texte.

Es folgt ein kurzer Überblick über den Inhalt der Arbeit.

In Kapitel 2 wird der Begriff des Suffixbaumes präzisiert und anschaulich gemacht. Es folgt die Definition der Größe V. Dieser einleitende Teil wird von einer Übersicht über die untersuchten Sprachen abgeschlossen.

Bemerkungen zur Methodik und die grundlegenden experimentellen Ergebnisse finden sich in Kapitel 3. Die Resultate werden kritisch diskutiert und ihre Implikationen in einem kurzen Abschnitt (Kapitel 3.5) zusammengestellt.

In Kapitel 4 wird der oben angesprochene Vergleich mit dem Zipfschen Gesetz durchgeführt. Kapitel 5 untersucht die Größe V und das Zipfsche Gesetz für zufällig erstellte Texte. Das Verhalten von V für Programmcode behandelt Kapitel 6.

Kapitel 7 enthält eine Zusammenfassung der experimentellen Ergebnisse und der Schlussfolgerungen, die man aus ihnen ziehen kann.

In den Anhängen am Schluss der Arbeit finden sich ergänzende Untersuchungen und detaillierte Informationen zu den untersuchten Sprachen, die für das grundlegende Verständnis der Arbeit nicht entscheidend sind.

Eine klärende Anmerkung ist angebracht: In dieser Arbeit bezeichnet der Terminus "natürliche Sprache" ohne Ausnahme natürliche Sprachen in ihrer schriftlichen Form.

2 Grundlegendes

2.1 Der Begriff des Suffixbaumes

Unter einem Baum versteht man in Mathematik und Informatik im Allgemeinen eine Struktur mit genau einem Wurzelknoten, von dem ausgehend sich einzelne Pfade immer weiter verzweigen, bis sie in Blättern enden. Von einem Baum spricht man, da es von der Wurzel aus gesehen keine Zusammenführungen gibt, sondern nur Verzweigungen. Gewöhnlich haben die Verbindungen von einem Knoten zum anderen (die Kanten) Beschriftungen.

Unter Suffix bzw. Präfix verstehen wir hier nicht die entsprechenden morphologischen Begriffe wie sie allgemein in der Linguistik verwendet werden, sondern ein beliebiges End- bzw Anfangstück eines Textes. In diesem Sinne ist 'ird gut' ein Suffix des Textes 'alles wird gut'. Als Text wiederum wird jede Aneinanderreihung beliebiger Zeichen bezeichnet, unabhängig von ihrer Länge und unabhängig davon, ob es sich um natürliche Sprache handelt.

Der Suffixbaum eines Textes ist eine Baumstruktur, in der jeder Pfad von der Wurzel zu einem beliebigen Blatt ein Suffix des Textes repräsentiert: Hängt man die Beschriftungen aller Kanten auf dem Weg von der Wurzel bis zum Blatt aneinander, so entsteht ein Suffix des eingelesenen Textes.

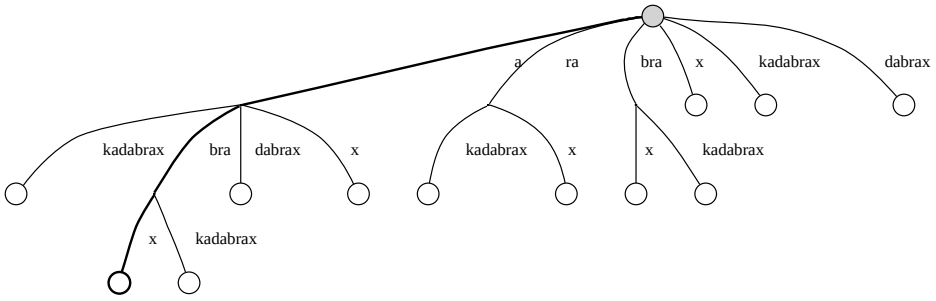


Abbildung 1: Der Suffixbaum des Textes "abrakadabrax". Das Beispiel aus dem Text ist hervorgehoben.

Betrachten wir den Text "abrakadabrax". Die Liste der Suffixe dieses Textes (im oben beschriebenen Sinne) ist[1]

```
abrakadabrax
brakadabrax
rakadabrax
akadabrax
kadabrax
adabrax
dabrax
abrax
brax
rax
ax
x
```

Der gesuchte Suffixbaum ist derjenige Baum, der alle diese Zeichenketten enthält und nur diese. Er ist in Abbildung 1 dargestellt. Das Suffix `abrax` zum Beispiel ergibt sich als `a+bra+x`. Es ist im Bild graphisch hervorgehoben.

Nicht nur jedes Suffix des Textes ist im Baum enthalten, sondern überhaupt jeder beliebige Teil des Textes, da jedes Teilstück des Gesamttextes auch Präfix eines Suffixes ist. Der Unterschied ist nur, dass echte Suffixe an Blättern enden, bloße Teilstücke des Textes aber nicht.

Dieser Umstand macht Suffixbäume zu einem extrem schnellen Hilfsmittel für die Suche in großen Texten, da die Zeit, die benötigt wird, von der Wurzel her bis zu einer gewissen Tiefe in den Baum einzudringen, unabhängig von der Gesamttiefe des Baumes ist. Daher ist auch die Zeit, die benötigt wird, um zu

[1] Wir verzichten auf den leeren String, der strenggenommen auch ein Suffix jeden Textes ist.

überprüfen, ob eine Zeichenkette im Text enthalten ist, unabhängig von der Größe des Textes. Sie ist lediglich proportional zur Länge der gesuchten Zeichenkette.

2.2 Zur Konstruktion von Suffixbäumen

Die Komplexität C eines Algorithmus A wird mit Hilfe der funktionalen Abhängigkeit der Rechenzeit R von der Länge n der Eingabe[2] angegeben: Gilt beispielsweise $R_A(n) = a^{bn}$ $(a, b > 0)$, so ist A von exponentieller Komplexität. Dies ist der ungünstigste Fall, da mit solchen Algorithmen nur Probleme bis zu einer bestimmten, von b abhängigen, Größe n_{max} behandelt werden können. Danach steigt der Rechenaufwand so schnell ins Unermessliche, dass selbst eine Steigerung der Rechenleistung n_{max} kaum erhöht. In der Praxis ebenfalls häufig sind Algorithmen kubischer bzw. quadratischer Komplexität: $R_A(n) = an^m$, mit $a > 0$ und $m \in \{2, 3\}$. Dies ist vor allem für große n erheblich günstiger. Aber immer noch wird es für große n_{max} immer aufwändiger, diese Grenze noch weiter zu erhöhen. Dieses Problem besteht bei linearer Komplexität nicht mehr, dh., wenn gilt $R_A = an$, mit $a > 0$. Hier wird bei verdoppelter Rechenleistung immer auch die Grenze n_{max} verdoppelt.

Obwohl ein Suffixbaum eine aufwändige Datenstruktur ist, gibt es dennoch verschiedene Verfahren, Suffixbäume in linearer Zeit zu erstellen. Zwei davon sind in [Gusfield1997] sehr anschaulich dargestellt. Ich greife in dieser Arbeit auf die dortigen Erläuterungen zu Ukkonens Algorithmus [Ukkonen1995] zurück. Ukkonens eleganter Algorithmus hat für unsere Zwecke einen großen Vorteil: Die Zeichen werden eines nach dem anderen in in den Baum eingefügt wie sie im Text erscheinen. Dabei ist der entstehende Baum zu jedem Zeitpunkt ein vollständiger Suffixbaum des bisher eingelesenen Textes.[3] Man kann dem Baum gewissermaßen beim Wachsen zusehen, während der Text Zeichen für Zeichen eingelesen wird.

Ich verzichte an dieser Stelle auf eine eigene Darstellung von Ukkonens Verfahren und verweise direkt auf die Ausführungen in [Gusfield1997] bzw. in [Ukkonen1995].

Pseudocode meiner eigene Implementierung des in der angegebenen Literatur eher abstrakt beschriebenen Algorithmus ist in Anhang A.4 abgedruckt.

Das Programm selbst ist in C++ geschrieben.

2.3 Motivation für diese Untersuchung

Warum diese Untersuchung? Was deutet darauf hin, dass man aus Suffixbäumen natürlichsprachiger Texte etwas Interessantes lernen kann?

Suffixbäume verwandeln die lineare Abfolge des Textes in eine globale Struktur: Ein Pfad von der Wurzel zu einem der inneren Knoten repräsentiert nicht ein

[2] n sauber zu definieren ist häufig nicht einfach, in unserem Falle ist es die Länge des Textes, aus dem ein Suffixbaum erstellt werden soll.

[3] Dies wird als die Onlineeigenschaft des Algorithmus bezeichnet.

bestimmtes Vorkommen einer Zeichenkette im Text, sondern alle ihre Vorkommen zugleich. Die Teilbäume unterhalb dieses Knotens fassen alle existierenden Fortsetzungen dieser Zeichenkette in einer gemeinsamen Struktur zusammen. In diesem Sinne macht ein Suffixbaum globale Eigenschaften des eingelesenen Textes sichtbar.

Damit hat man eine vollständige Übersicht, was im Text vorkommt und was sich wiederholt. Man kann sogar ohne viel Aufwand protokollieren, was sich wie oft wiederholt. Durch diese Vollständigkeit kann man hoffen, sehr feine statistische Korrelationen aufzudecken. Herkömmliche statistische Maße wie Mutual Information [Li1989] nutzen im Vergleich dazu nur einen Bruchteil der vorhandenen Information aus, da sie z.B. nur Kontext einer festen Größe verwenden. Möglicherweise resultiert daraus auch die Schwäche der so beobachtbaren Korrelationen (ebd.). Die Struktur von Suffixbäumen reflektiert die statistischen Verhältnisse auf jeder Ebene der Sprache: Phänomene auf der Ebene der Zeichen sind gleichermaßen zugänglich wie solche auf Wort- oder Satzebene.

Das in dieser Arbeit behandelte Phänomen ist ein Zufallsfund. Eher nebenbei stieß ich auf eine unerwartete Ähnlichkeit in der Struktur von Suffixbäumen aus Texten in verschiedenen europäischen Sprachen. Als sich dieselbe Ähnlichkeit auch in Suffixbäumen von Texten typologisch weit entfernter Sprachen fand, erschien mir das als ausreichend interessant, um diese Arbeit darauf aufzubauen. Es scheint möglich, dass es sich nur um eine von vielen Besonderheiten in der Struktur von Suffixbäumen natürlicher Sprachen handelt.

2.4 Die untersuchte Größe

Als *innere Knoten* eines Baumes seien alle Knoten außer dem Wurzelknoten bezeichnet. Im Folgenden bezieht sich der Terminus *Knoten* im Allgemeinen nur auf innere Knoten.

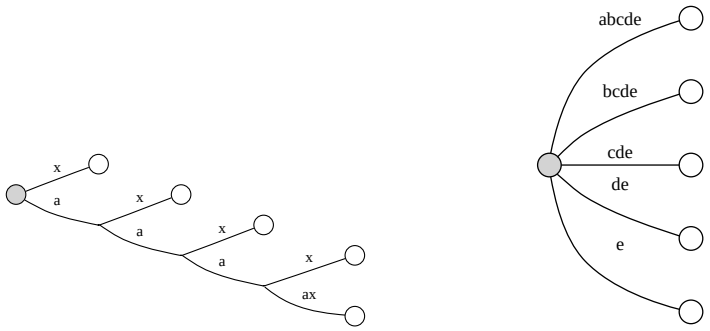
Die Zahl der (inneren) Knoten K eines Suffixbaumes wird im Allgemeinen mit der Zeichenzahl T des eingelesenen Textes wachsen. Der genaue Wert von K dagegen hängt stark vom eingelesenen Text ab.

K kann für Texte der Länge T maximal $T-2$ betragen[4]. Als Beispiel für diesen Extremfall betrachten wir den Text, der aus der fünf Zeichen langen Kette "aaaax" besteht. Der Suffixbaum zu diesem Text ist in Abbildung 2(a) dargestellt.

Die Frage nach dem Baum mit der minimalen Anzahl an (inneren) Knoten ist im allgemeinen Fall sehr viel schwerer zu beantworten. Sicherlich ist sie aber durch 0 nach unten begrenzt: Wiederholt sich kein Symbol des Textes, so wird dieses Minimum für die Zahl der inneren Knoten auch erreicht (siehe Abbildung 2(b) als Beispiel).

Betrachten wir nun das Verhältnis V von Knotenzahl zu Textlänge: $V = K/T$. V liegt immer zwischen 0 und 1, da der Wert für K zwischen 0 und $T-2$ liegt

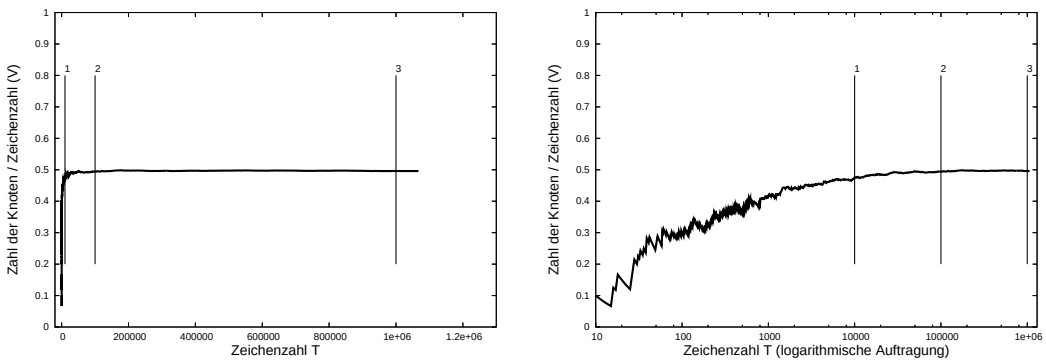
[4] Auf einen Beweis wird hier verzichtet. Er kann mittels vollständiger Induktion leicht geführt werden.



(a) (b)

Abbildung 2: Einfache Beispiele für einen Baum mit der maximal (a) und minimal (b) möglichen Anzahl innerer Knoten.

und $(T-2)/T$ für wachsende T schnell gegen 1 strebt.



(a) (b)

Abbildung 3: Das Verhältnis V von Knotenzahl K zu Textlänge T für den russischen Korpus. Beide Graphiken zeigen dieselben Daten für denselben Wertebereich. In beiden Fällen ist V über der Textlänge T aufgetragen. In Teilbild (b) ist die x-Achse in logarithmischem Maßstab dargestellt. Die eingezeichneten Balken dienen nur der Illustration. Sie befinden sich in beiden Teilbildern an denselben Stellen.

Diese Arbeit behandelt das Verhalten von V in der Abhängigkeit von T für Texte natürlicher Sprachen. Dabei nutzen wir aus, dass in jedem Teilschritt von Ukkonens Algorithmus ein gültiger Suffixbaum entsteht (vergleiche Kapitel 2.2). So können wir V über der wachsenden Zeichenzahl T des nach und nach eingelesenen Textes auftragen.

Ein Beispiel ist in Abbildung 3(a) dargestellt. Die Kurve beginnt in der unteren linken Ecke bei einem V von 0, da der Baum zu Beginn noch keine Knoten hat. Beinahe sofort springt sie jedoch auf einen konstanten Wert sehr nahe bei $0,5$.

Dieses markante Verhalten findet sich nicht nur für Russisch, sondern auch in allen anderen von mir untersuchten Sprachen. Dieses Phänomen bildet den Kern der vorliegenden Arbeit.

Man kann damit rechnen, dass es ein erheblicher Unterschied ist, ob wir zwei Texte der Längen 10 und 110 Zeichen vergleichen, oder zwei Texte der Längen 1000 und 1100, obwohl die Differenz jeweils 100 Zeichen beträgt. Dagegen erwartet man intuitiv ähnliche statistische Unterschiede zwischen Texten der Längen 10 und 100 wie zwischen solchen der Länge 1.000 und 10.000. Daher ist es wünschenswert, dass gleiche Abstände auf der x-Achse gleichen Faktoren in der Zunahme von T entsprechen und nicht einem Wachstum der absoluten Zeichenzahl.

Dies erreicht man durch logarithmische Skalierung der x-Achse. Ein Beispiel ist Abbildung 3(b). Es sind dieselben Daten im selben Wertebereich dargestellt wie in Teilbild (a). Die numerierten Balken dienen nur dem Vergleich. Sie haben in Teilbild (b) gleichen Abstand voneinander, obwohl sich von Balken zu Balken die Zeichenzahl verzehnfacht. Alle weiteren Graphiken, die den Verlauf von $V(T)$ darstellen, sind –sofern nicht anders vermerkt– logarithmisch in der x-Achse.

2.5 Die Bedeutung von $V(T)$

V ist definiert als $V = K/T$, wobei K die Zahl der (inneren) Knoten im Suffixbaum ist und T die Zeichenzahl des Textes. Doch was sagt ein bestimmter Wert von V aus über die Eigenschaften des Textes?

Die *Tiefe* eines Knotens k in einem Suffixbaum sei wie üblich definiert als die Zahl der Knoten auf dem Weg von k zum Wurzelknoten[5]. Die *Zeichentiefe* von k bezeichne die Summe der Längen der Zeichenketten an allen Kanten zwischen k und der Wurzel.

Die Existenz eines Knotens im Suffixbaum bedeutet immer, dass sich eine Zeichenkette z im Text wiederholt und jeweils unterschiedlich fortgesetzt wird. Dann wiederholt sich aber die Zeichenkette z' ebenfalls, die aus z durch Wegschneiden des ersten Zeichens entsteht. Auch hier gibt es einen Knoten, denn auch z' wird im Text unterschiedlich fortgesetzt. Für jeden Knoten der Zeichentiefe n im Baum gibt es also $n-1$ Knoten der Zeichentiefe $n-1$, $n-2$, ..., 1.

Die Zahl der insgesamt vorhandenen Knoten hängt also eng mit dem Ausmaß an Wiederholungen im Text zusammen: Viele lange Wiederholungen führen zu vielen Knoten im Baum, wenige kurze Wiederholungen bilden wenige Knoten. Einen bildlichen Eindruck von dieser Tatsache konnte man bereits aus Abbildung 2 bekommen.

Abbildung 4 ist eine detailliertere Illustration für den Zusammenhang zwischen V und den Eigenschaften des Textes. Den Beginn des Textes bildet ein etwa 200 Zeichen langer Satz des Deutschen. Er beginnt folgendermaßen: "dies ist immer wieder nur die eine selbe zeile[...]". Erst das 6. Zeichen[6], ein 'i' ist eine

[5] Der Wurzelknoten habe die Tiefe 0.

[6] Leerzeichen werden ja mitgezählt. Sie sind im folgenden als '_' dargestellt.

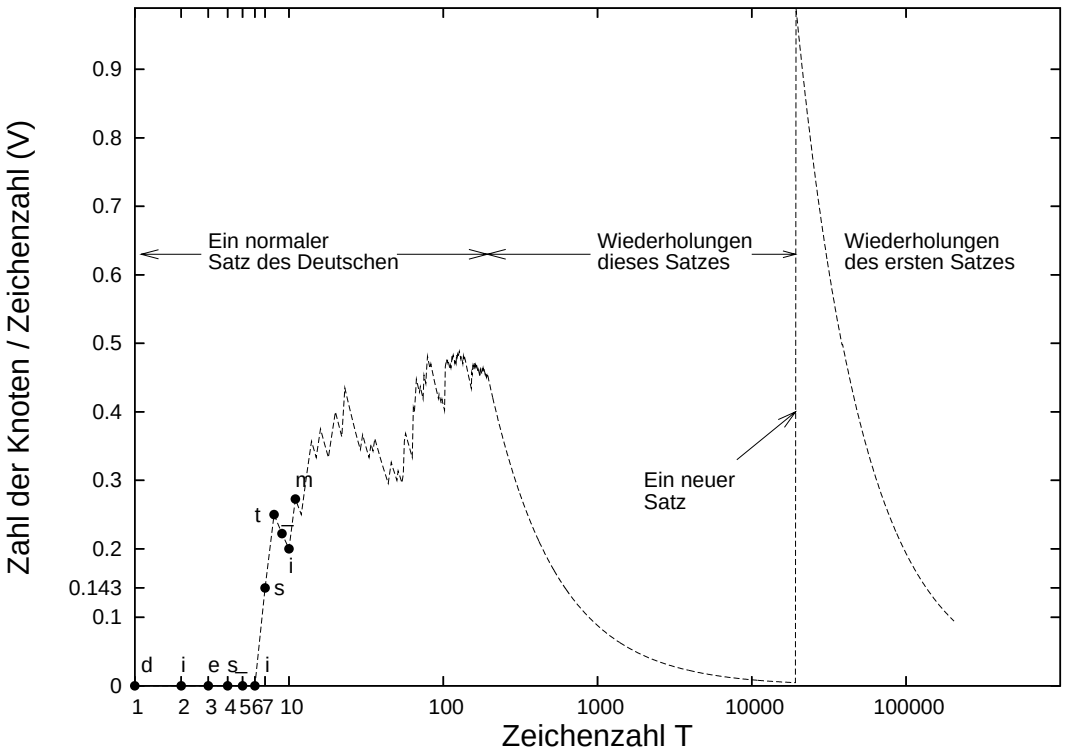


Abbildung 4: Ein konstruiertes Beispiel, das helfen soll, die Bedeutung von V als relatives Maß der Wiederholungen in einem Text nachzuvollziehen.

Wiederholung, vorher gibt es keine Knoten, V ist 0. Auch beim Einlesen des 6. Zeichens selber ändert sich daran nichts, da die einzige Wiederholung bis dahin noch nicht zu Ende ist. Dies ändert sich mit dem 7. Zeichen, nun gibt es die beiden Zeichenketten "ie" und "is". Es entsteht ein Knoten nach dem 'i' und es gilt $V = K/T = 1/7 \approx 0,143$. das 's' ist selbst eine Wiederholung. Sie endet mit dem 8. Zeichen: Nun gibt es "s_" und "st". Ein neuer Knoten entsteht, es gilt $V = 2/8 = 0.25$. Der 9. und der 10. Buchstaben ('_' und 'i') bilden wiederum eine Wiederholung, die aber erst mit dem 11. Buchstaben, dem 'm' endet. Deswegen sinkt V bis $T = 10$ wieder auf $2/10 = 0,2$ ab, um erst danach wieder anzusteigen.

Bis zum Ende des ersten Satzes verläuft die Kurve gezackt. In unregelmäßiger Folge werden Knoten in den Baum eingebaut, da sich kleinere Zeichenfolgen und ganze Worte wiederholen, dann aber unterschiedliche Fortsetzungen folgen.

Anschließend wiederholt sich der erste Satz immer und immer wieder, 99 mal. Hier entstehen keine neuen Knoten, da es keine neuen Fortsetzungen bereits bekannter Zeichenketten gibt. Daher sinkt das Verhältnis $V = K/T$ immer weiter ab.

Dies ändert sich nach 20.000 Zeichen schlagartig. Hier beginnt ein neuer Satz. Mit einem Mal entstehen Knoten mit erheblicher Zeichentiefe im Baum, da es nun extrem lange Suffixe gibt, die sich nur an ihrem Ende voneinander unterscheiden.

Wie weiter oben dargestellt, führt dies zur Produktion sehr vieler weiterer neuer Knoten näher an der Wurzel.

Es ergibt sich ein Baum mit Strukturen die dem in Abbildung 2(a) (Seite 7) dargestellten Extrembeispiel ähneln. Entsprechend springt der Wert von V auf einen Wert sehr nahe bei 1.

Den Rest des Textes bildet eine Aneinanderreihung immer weiterer Wiederholungen des ersten Satzes. Da es keine neuen Zeichenketten mehr gibt, müssen auch keine neuen Knoten in den Baum eingefügt werden. V sinkt wieder auf einen Wert nahe 0.

$V(T)$ ist demnach ein relatives – d.h. auf die Textlänge bezogenes – Maß für die Wiederholungen im Text. Ein Anstieg in V bedeutet das Ende einer Wiederholung, je steiler, desto länger war die Wiederholung. Ein Abfall von V dagegen zeigt dagegen neue Wiederholungen an. Hier ist nur ein hyperbolischer[7] Abfall möglich, da die Zahl der Knoten im Baum mit wachsendem T nie kleiner werden kann.

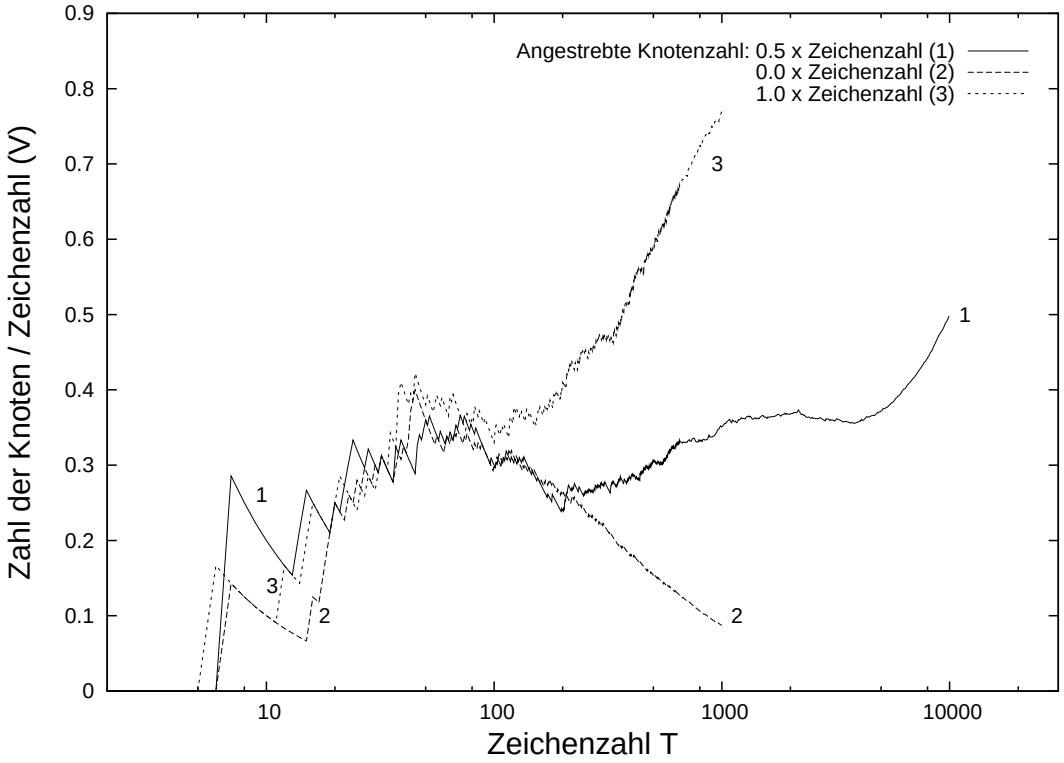


Abbildung 5: Der Verlauf von $V(T)$ für Texte, die mit dem dargestellten Mutationsverfahren auf ein bestimmtes Verhältnis von Knoten- zu Zeichenzahl hin erzeugt wurden.

Die meisten bisherigen Beispiele sind künstlich. Sie wurden erdacht, um die grundlegenden Eigenschaften von Suffixbäumen im Allgemeinen und der Größe

[7] $f(x) = 1/x$ ist das einfachste Beispiel für eine hyperbolische Funktion.

V im Besonderen möglichst klar herauszustellen. Was für Werte kann V unter realistischeren Bedingungen annehmen? Um diese Frage zu beantworten, habe ich Texte mit dem Computer erzeugt, die einem vorgegebenen V möglichst nahe kommen.

Startpunkt des Verfahrens waren mit Hilfe eines Zufallsgenerators erstellte Texte. Solche Texte werden in Kapitel 5 noch genauer untersucht.

In einem zweiten Schritt wird V für diesen zufälligen Text ermittelt. Anschließend werden an willkürlich gewählten Stellen Zeichen durch andere ersetzt, aus "...aughj**a**hgfkl..." wird beispielsweise "..aughj**g**hgfkl...". Ist V für den veränderten Text näher am angestrebten Wert, wird die Veränderung beibehalten, sonst wird sie verworfen. Es ist gut, sich klarzumachen, dass ich hier nur V für den gesamten Text berücksichtigt habe. Sonst in dieser Arbeit liegt der Schwerpunkt auf dem Verlauf von V als Funktion von T (geschrieben $V(T)$).

Dieses Experiment wurde für folgende Zielwerte von V durchgeführt: 0, 0,5 und 1. Die Alphabetgröße war jeweils 25. Die Ergebnisse sind in Abbildung 5 dargestellt. Der Computer rechnete jeweils so lange, bis V für den gesamten Text entweder den angestrebten Wert erreicht hatte, oder bis nach etwa 20 Stunden Rechenzeit der Prozeß unterbrochen wurde. Nur Text Nr. 1 erreichte den vorgegebenen Wert von $1/2$. Da dieses Ergebnis sogar sehr schnell zustande kam, konnte die Textlänge hier 10 mal höher gewählt werden als in den beiden Extremfällen 0 und 1.

Man sieht, dass sich V unter solchen Bedingungen in einem Wertebereich von etwa 0,1 bis etwa 0,8 bewegt. Extremere Werte sind mit Texten, die nicht per Hand erstellt wurden, schwer zu erreichen. Die Kurven in Abbildung 5 werden in Kapitel 3.3 auf Seite 21 noch einmal thematisiert werden, dort im Vergleich mit $V(T)$ für Texte natürlicher Sprache.

2.6 Die betrachteten Sprachen

In die Ergebnisse dieser Arbeit sind Daten aus 21 verschiedenen Sprachen aus Europa, Asien und Indien eingeflossen. Sie entstammen den vier Sprachfamilien Indo-Europäisch, Sino-Tibetisch, Dravidisch und Finno-Ugrisch. Die Schriftsysteme dieser Sprachen verwirklichen vier verschiedene Möglichkeiten wie die Zeichen der Schrift und die Laute der Sprache aufeinander abgebildet werden können.

2.6.1 Einordnung in Sprachfamilien

Die Einteilung der 21 Sprachen in Sprachfamilien ist bis auf hier unwichtige Details unumstritten. Die in Tabelle 1 auf Seite 12 dargestellte Typologie habe ich aus den Angaben der umfangreichen Enzyklopädie [Asher1994] zusammengestellt.

1. Indo-Europäisch
    - Slawische Sprachen
        - *Russisch*
    - Germanische Sprachen
        - Westgermanische Sprachen
            - *Deutsch*
            - *Englisch*
    - Romanische Sprachen
        - *Französisch*
    - Indo-Iranisch
        - Indo-Arisch
            - *Assamese*
            - *Bengali*
            - *Gujarati*
            - *Hindi*
            - *Marathi*
            - *Oriya*
            - *Punjabi*
            - *Sinhala*
            - *Urdu*
            - Dardische Sprachen
                - *Kashmiri*
2. Dravidisch
    - Süddravidisch
        - *Tamil*
        - *Kannada*
        - *Malayalam*
    - Telugu-Kui
        - *Telugu*
3. Sino-Tibetisch
    - *Chinesisch*
4. Uralische Sprachen
    - Finno-Ugrische Sprachen
        - Finno-Lappisch
            - *Finnisch*
        - Ugrisch
            - *Ungarisch (Magyar)*

Tabelle 1: Genealogische Typologie der untersuchten Sprachen

2.6.2 Typologie der Schriftsysteme

Es gibt in der Literatur einige Klassifikationen von Schriftsystemen, die sich meist nicht völlig widersprechen, aber dennoch zum Teil erhebliche Unterschiede aufweisen.

Ich übernehme die Klassifikation in [Daniels1996, 4], da sie systematisch ist und frei von ideologischen Vorurteilen wie der Höherstellung des (europäischen) Alphabets.

> ... half a dozen fundamentally different types of writing systems have been devised with respect to how symbols relate to the sounds of languge (and there's no reason more types could not be invented). In a *logosyllabary*, the characters of a script denote individual words (or morphemes) as well as particular syllables. In a *syllabary*, the characters denote particular syllables, and there is no systematic graphic similarity between the characters for phonetically similar syllables. In a consonantary, here called an *abjad* as a parrallel to "alphabet" (the word is formed from the first letters of the most widespread example, the Arabic script, in their historic order, [...]), the characters denote consonants (only). In an *alphabet*, the characters denote consonants and vowels. In an *abugida*, each character denotes a consonant accompanied by a specific vowel, and the other vowels are denoted by a consistent modification of the consontant symbols, as in Indic scripts. (The word is Ethiopic, from the first four consonants and the first four vowles of the traditional order of the script, [...]. In a *featural* system, like Korean [...], the shapes of the characters correlate with distinctive features of the segments of the language.

In Bezug auf die verwendeten Schriftsysteme teilen sich die betrachteten Sprachen in folgende Gruppen auf, die sich nur grob mit der oben angegebenen genetischen Verwandtschaft decken:

1. **Abugidas:** Die meisten Sprachen des indischen Sprachraums verwenden Abudigas. Sie stammen letztlich alle von der Brahmischrift ab, die sich bis ins dritte Jahrhundert vor Christus zurückverfolgen läßt. Dies gilt für Assamese, Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriyia, Sinhala, Tamil und Telugu. Die allermeisten dieser Sprachen verwenden verwandte aber unterschiedliche Schriften. Nur Hindi und Marathi verwenden beide die Schrift *Devanagari*. Die beiden dravidischen Sprachen Telugu und Kannada haben ebenfalls sehr eng verwandte Schriften. Dasselbe gilt auch für Tamil und Malayalam, die allerdings verschiedenen Sprachfamilien angehören. Dabei nimmt Tamil jedoch eine Sonderrolle ein, da es starke alphabetische Züge entwickelt hat. (Vergleiche hierzu das in Kapitel 3.4.2 gesagte).

2. Einige der indischen Sprachen allerdings verwenden Schriftsysteme, die auf dem Arabischen basieren. Dies betrifft von den hier untersuchten Sprachen

Urdu, Kashmiri und Punjabi. Das Arabische ist nach [Daniels1996] ein **Abjad**. Dasselbe gilt nach derselben Quelle auch für Urdu [Daniels1996, 754]. (zu Kashmiri siehe unter 3.) Punjabi wird in zwei verschiedenen Schriften geschrieben. Einerseits in Gurmukhi, abstammend von Brahmi wie die meisten indischen Schriften (siehe 1). Andererseits wird - vor allem im pakistanischen Teil des Panjab ([Daniels1996, 395]) die persoarabische Schrift verwendet. Das Persoarabische wiederum ist ein Abjad [Daniels1996, 747]. Wir verwenden hier nur den Teil des Punjabikorpus aus [Emille2004], der in Persoarabisch verfasst ist.

3. **Alphabetisch geschriebene Sprachen:**

   - Die für das Kashmirische verwendete Schrift leitet sich wie Urdu zwar vom Persischen Abjad her, wird in [Daniels1996, 753] aber als Alphabet bezeichnet. Wir folgen dieser Einteilung.
   - Deutsch, Englisch, Finnisch, Französisch und Ungarisch werden mit teilweise unterschiedlichen Varianten des romanischen Alphabetes geschrieben. Dasselbe gilt für Pinyin, eine romanisierte Umschrift der chinesischen Schriftzeichens. Mehr zu Pinyin in Kapitel 3.4.1.
   - Russisch wird mit dem kyrillischen Alphabet geschrieben.

4. Das System der chinesischen Schriftzeichen bildet eine **Logosyllabery**[8].

2.6.3 Korpora

Als Korpus werden in der Sprachwissenschaft sehr allgemein Texte und vor allem Textsammlungen bezeichnet, die für eine Sprache, einen Teilaspekt einer Sprache oder von Sprache allgemein repräsentativ genug sind, um wissenschaftlich (oft statistisch) fundierte Erkenntnisse daraus gewinnen zu können. Es folgt eine Auflistung der hier verwendeten Korpora.

- Die Korpora der indischen Sprachen Assamese, Bengali, Gujarati, Hindi, Kannada, Kashmiri, Malayalam, Marathi, Oriya, Punjabi, Sinhala, Tamil, Telugu und Urdu entstammen allesamt der Emille-Sammlung [Emille2004]. Emille ist ein Gemeinschaftsprojekt der Universität von Lancaster und dem "Central Institute of Indian Languages (CIIL)" in Indien. Sein monolingualer Teil besteht aus insgesamt 96 Millionen Worten, davon etwa 3 Millionen gesprochene Sprache. Ich habe für diese Arbeit einen Teil der schriftsprachlichen Daten zur Auswertung herangezogen. Der größte Teil davon stammt aus den Webseiten verschiedener indischer Tageszeitungen.

[8] Ich werde hier keinen der englischen Fachbegriffe übersetzen. das Wort *Logosyllabery* ist auf Deutsch leider nicht angenehm, dies schien mir aber eher hinzunehmen als eine unpassende Übersetzung.

- Der verwendete chinesische Korpus [LCMC2004] stammt ebenfalls aus Lancaster. Er ist gedacht als Gegenstück zum Freiburg-LOB Corpus of British English (FLOB), in dem Sinne, dass dasselbe Format verwendet wird und eine möglichst vergleichbare Textzusammenstellung. Es handelt sich um eine ausgewogene Sammlung von Texten aus Kategorien von Religion bis Humor.

- Die deutschen Texte entstammen der Süddeutschen Zeitung. Es handelt sich um eine automatisch erstellte Artikelsammlung der SZ-online aus dem Zeitraum von August 2002 bis Oktober 2003.[9] Die Daten wurden damals automatisch am CIS heruntergeladen. Die Extraktion der Texte aus den rohen HTML-Dateien und die genaue Zusammenstellung habe ich mit Hilfe eines Perl-Skriptes durchgeführt.

- Für das Englische habe ich den Browncorpus [Brown1998] herangezogen.

- Der verwendete finnische Korpus ist ein Provisorium. Es handelt sich um eine sehr heterogene Textmischung, geschickt von einem Freund. Sie besteht aus Kinderbüchern, technischen Anleitungen und wissenschaftlichen Abhandlungen. Für eine so kleine Textsammlung (etwa 500.000 Zeichen) ist diese Vielfalt sicherlich zu groß, um von einem Korpus im eigentlichen Sinne zu sprechen.

Als Beispiel für französisch, russisch und ungarisch dienten mir elektronische Ausgaben von Romanen.

- Französisch: Der erste Band "Unterwegs zu Swann" aus Marcel Prousts Zyklus "Auf der Suche nach der verlorenen Zeit" [Proust2001] in der elektronischen Ausgabe des Gutenbergprojektes.

- Russisch: "Schuld und Sühne" von Fjodor Dostojewski in der elektronischen Ausgabe der "Biblioteka Maksima Moshkova" (http://lib.ru).

- Ungarisch: "Egri csillagok" von Gárdonyi Géza (1863-1922).

[9] http://www.sz-online.de

3 Das experimentelle Resultat

In diesem Kapitel stelle ich die experimentellen Ergebnisse zum Verlauf von $V(T)$ in den untersuchten Sprachen vor.

Das Kapitel beginnt mit technischen Bemerkungen, die die Einschätzung der Resultate erleichtern sollen. Es folgen Überlegungen, welches Verhalten von V man intuitiv erwartet. Diese bilden einen Kontrast zu den tatsächlichen empirischen Ergebnissen. Genauer gehe ich auf Chinesisch und Tamil ein, zwei Einzelfälle, die sich auf den ersten Blick nicht recht ins Bild fügen. Abschließend für dieses Kapitel werden die zentralen Aussagen, Hypothesen und Fragen dieser Arbeit zusammengefasst.

3.1 Methodologische Anmerkungen

3.1.1 Auswahl der Korpora

Gegenstand dieser Arbeit ist ein Aspekt der Statistik natürlichsprachiger Texte. Damit stand ich vor einem Standardproblem der Computer- und Textlinguistik: Es wird ein Korpus benötigt, aus dem verlässliche statistische Daten gewonnen werden können.

Für die meisten computerlinguistischen Anwendungen bedarf es eines Korpus, der für die betrachtete Sprache oder Teile davon repräsentativ ist. Deshalb sind wissenschaftliche Korpora meist Sammlungen kleinerer Texte, deren Themen und Stile innerhalb des angestrebten Rahmens möglichst breit gefächert sind.

Diese Bedingung, dass ein Korpus ausreichend breit gestreut sein sollte, muss hier nicht erfüllt werden: Die Grundthese dieser Arbeit ist –vereinfacht zusammengefasst–: Eine gewisse statistische Größe verhält sich in allen natürlichsprachigen Texten uniform. Um eine These dieser Art über ein breites Spektrum von Sprachen zu testen, ist es nicht erforderlich, für alle Sprachen weit gefächerte Korpora zur Verfügung zu haben. Trifft die These nicht für alle Texte in allen Sprachen zu, so zeigt sich dies auch anhand einzelner Texte aus einzelnen Sprachen. Daher war es z.B. unproblematisch, für russisch, französisch und ungarisch Romane als Korpora zu verwenden.

Auf der anderen Seite muss für die meisten Fragestellungen darauf geachtet werden, dass die Textzusammenstellung des Korpus den gesteckten Rahmen nicht verlässt: Wenn beispielsweise ein Korpus der Standardsprache erstellt werden soll, sind Varietätentexte zu vermeiden. Auch diese Bedingung muss im vorliegenden Fall nicht allzu exakt eingehalten werden. Dies liegt in Natur der Größe V selbst begründet. In Anhang A.2 wird genauer dargelegt, dass und warum sich selbst extrem heterogene Texte nicht abweichend von den übrigen verhalten.

Eine dritte Forderung, die meist implizit an wissenschaftliche Korpora gestellt wird, ist dagegen auch hier von größter Bedeutung: In Kapitel 2.5 (siehe vor allem Abbildung 4, Seite 9) wurde besprochen, in welchem Sinne $V(T)$ ein

Maß für Wiederholungen im Text darstellt. Es ist im Lichte dieser Überlegungen entscheidend, dass unsere Korpora ein natürliches Maß an Wiederholungen enthalten. Was aber ist in diesem Zusammenhang ein "natürliches Maß an Wiederholungen", oder was ist ganz allgemein natürlicher Sprachgebrauch? Über eine intuitive Definition kann hier nicht hinausgegangen werden: Normal ist, was als normal empfunden wird: In diesem Sinne ist ein Zeitungsartikel oder eine Email normaler Sprachgebrauch, ein Telefonbuch oder die dreimalige Wiederholung desselben Zeitungsartikels nicht.

Daher teile ich die Korpora[10] in zwei Gruppen ein: Diejenigen, die dieses Kriterium der Natürlichkeit dem Augenschein nach "offensichtlich" erfüllen und die, für die das nicht ohne weiteres gilt.

In die erste Gruppe fallen die folgenden Korpora (siehe für mehr Details):

1. Englisch: Der Browncorpus [Brown1998] wird häufig für wissenschaftliche Zwecke verwendet verwendet und gilt allgemein als vertrauenswürdig.

2. Deutsch: Diesen Korpus habe ich selbst aus automatischen Downloads der SZ-online erstellt. Dabei achtete ich sorgfältig darauf, denselben Text nicht zweimal aufzunehmen.[11] Auch Wiederholungen, die sich aus den täglich gleichen Teilen einer Zeitung ergeben (gewisse Überschriften, das Impressum etc.), wurden eliminiert.

3. Französisch, Russisch und Ungarisch: Hier handelt es sich um Romane. Da ein Roman gewöhnlich ein fortlaufender und vom Autor absichtlich genau in dieser Form erstellter Text ist, handelt es sich per se um natürlichen Sprachgebrauch.

Alle anderen Korpora sind entweder zu kurz (finnisch) oder in ihrer Herkunft ohne Muttersprachler schwer zu überprüfen und zu großen Teilen aus Onlinepublikationen erstellt (die indischen Korpora [Emille2004]). Typische Probleme solcher Korpora wurden soeben angesprochen (siehe Punkt 2 und Fußnote 11).

Diese übrigen Korpora sind für unsere Zwecke nicht wertlos, aber die aus ihnen gewonnenen Erkenntnisse sind schwerer zu bewerten.

3.1.2 Vorbehandlung der Korpora

Aus den rohen Korpora wurden alle leeren Zeilen, alle Zeilenumbrüche und alle Häufungen von Leerzeichen entfernt. Der Vorteil dieser Maßnahme ist eine gewisse Standardisierung: Parameter wie die Zeilenlänge werden als nichtsprachliche Aspekte des Textes behandelt. Als Nachteil kann man die Veränderung des Originals sehen. Der Einfluss dieser Vereinheitlichung ist aber gering. Dies verdeutlicht Abbildung 6.

[10] Für eine Auflistung der verwendeten Korpora siehe Kapitel 2.6.3

[11] Dies passiert leicht, wenn der selbe Text über mehrere Tage im Netz bleibt oder in mehreren Versionen existiert, die sich nur minimal unterscheiden.

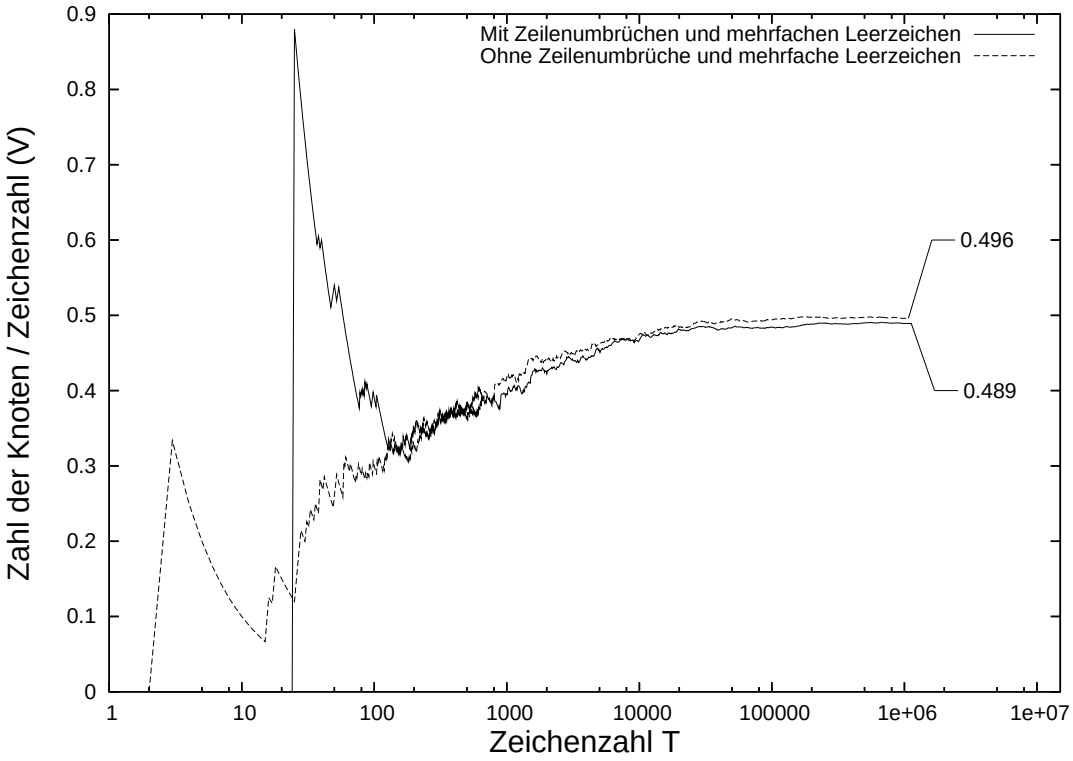


Abbildung 6: Wie stark beeinflusst das Entfernen jeder Formatierung das Ergebnis? Beide Kurven zeigen $V(T)$ für den russischen Text ("Schuld und Sühne"). Der Unterschied in der Länge des Textes hat seine Ursache natürlich im Entfernen der Formatierungszeichen.

3.1.3 Einlesen in den Suffixbaum

Zur korrekten Konstruktion des Suffixbaumes eines Textes kommt es lediglich auf zwei Dinge an: Die Texte müssen zeichenweise eingelesen werden und die eingelesenen Zeichen müssen miteinander vergleichbar sein. Eine Rückübertragung in einen konkreten Zeichensatz findet dagegen nicht statt. Daher ist es ausreichend, die Bitfolgen der einzelnen Zeichen intern als (Binär)zahlen zu interpretieren. Dies ist nötig, wenn ein Zeichen im Text mehr als ein Byte Raum einnimmt, da der c++ Standarddatentyp `char` intern durch nur ein Byte repräsentiert wird. `integer`-Variablen dagegen besitzen vier Byte.

Die englischen, deutschen, französischen und russischen Texte stellen kein Problem dar, da hier jedes Zeichen genau ein Byte verbraucht.

Ein Byte sind acht Bit. Damit können höchstens $2^8 = 256$ Zeichen kodiert werden.[12] Dies ist für viele Schriftsysteme nicht ausreichend. Es scheint wünschenswert, alle Schriftsysteme der Welt in einer einzigen Kodierung zusammenzufassen.

[12] In einigen Codierungen werden aus technischen und historischen Gründen nur die ersten 7 Bit verwendet. Dann sind sogar nur 128 Zeichen darstellbar. ASCII ist ein Beispiel hierfür.

Eine Lösung des Problems ist Unicode. Dieser internationale Standard vergibt für alle bekannten Zeichen eine eindeutige Nummer. Eine Kodierung, die diesen Standard umsetzt, kann keine Einbytekodierung mehr sein.

Die indischen Texte lagen in einer Unicodekodierung vor, die jedes Zeichen durch zwei Byte[13] repräsentiert [Emille2004]. Ich habe sie in ucs-4 umgewandelt, wo jedes Zeichen genau vier Byte verbraucht. Diese 4-Byte-Blöcke können direkt als (`integer`-)Zahl eingelesen werden.

3.2 Welches Ergebnis erwartet man für natürliche Sprachen?

Natürliche Sprachen haben einen extrem unterschiedlichen morphologischen Reichtum. Während in isolierenden Sprachen wie dem Englischen für jedes Wort höchstens einige wenige Formen existieren, zählt man in anderen Sprachen oft Tausende von verschiedenen Formen für ein einziges Paradigma. Diese Sprachen mit reicher Flexion kann man weiterhin unterteilen: In agglutinierenden Sprachen wie dem Finnischen sind die Endungen weitestgehend modular aufgebaut. Sie sind in kleinere Teile zerlegbar, die jeweils genau eine grammatische Kategorie repräsentieren. In sogenannten flektierenden Sprachen wie Latein oder Russisch dagegen bilden Endungen einen einheitlichen Block und kodieren als ganzes ein Bündel grammatischer Kategorien.

Alle existierenden Sprachen sind mehr oder weniger Mischungen dieser verschiedenen Typen.[14]

Deutsch kann als eine weitgehend flektierende Sprache bezeichnet werden, obwohl isolierende Züge bestehen. Erstellen wir in Gedanken einen Suffixbaum aus unserem deutschen Korpus (siehe Kapitel 2.6.3 auf Seite 14): Dort finden sich die Textfetzen “heult kurz auf”, “heulte auch er mit den Wölfen”, “heulend zusammengebrochen” und “heulten Kinder”. Ein Ausschnitt des Suffixbaumes sieht also aus wie in Abbildung 7(a) dargestellt. In einem ausreichend großen Korpus wird sich wenigstens für die häufigeren Verben jeweils ein Unterbaum bilden, der das gesamte Formenparadigma repräsentiert. Die Unterbäume der verschiedenen Verben werden sich untereinander ähneln und beispielsweise Unterschiede zu den Unterbäumen der Nominavorkommen zeigen.

Im (englischen) Browncorpus dagegen kommen folgende Zeichenketten vor: “airport remarks that politicians in the state are all the same”, “airport to express mortification to the Colombian foreign minister” und “airport in Vermont”. Es ergibt sich ein Teilbaum wie in Abbildung 7(b). Allgemein erwartet man von Wort zu Wort sehr unterschiedliche Unterbäume, die sich innerhalb der Worte kaum, an Wortgrenzen aber sehr stark verzweigen.

[13] Dies würde nicht für den gesamten Unicodezeichenvorrat ausreichen, wohl aber für die indischen Skripte

[14] Selbstverständlich gibt es neben dieser klassischen Einteilung noch viele andere Typologien (siehe z.B. [Bußmann2002, 634]).

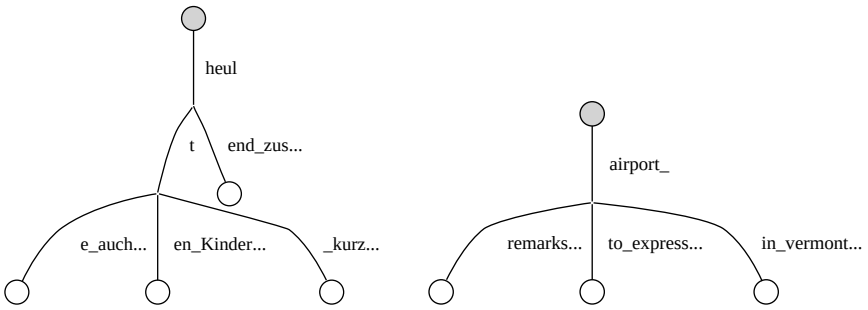


Abbildung 7: Beispiele aus den entsprechenden Korpora für typische Verzweigungen im Suffixbaum für deutsch (a) und englisch (b). Leerzeichen wurden um besserer Lesbarkeit willen durch Unterstriche ersetzt. Es sind nur sehr kleine Ausschnitte der vollen Suffixbäume zu sehen. Die ausgefüllten Kreise repräsentieren den Wurzelknoten, die leeren Blätter.

Bereits aufgrund der völlig verschiedenen morphologischen Strukturen natürlicher Sprachen ist also mit strukturell unterschiedlichen Suffixbäumen zu rechnen.

Man erwartet daher auch, dass der Verzweigungsgrad der Suffixbäume und seine Abhängigkeit von der Textgröße von Sprache zu Sprache stark variiert. Damit sollte sich auch das Verhalten von $V(T)$ jeweils gravierend unterscheiden, umso stärker, je weiter die betrachteten Sprachen typologisch voneinander entfernt sind.

Diese plausible Vermutung jedoch erweist sich als falsch.

3.3 Das experimentelle Ergebnis

In Abbildung 8 ist $V(T)$ für alle 21 Korpora[15]zusammengefasst. Die Kurven ähneln sich untereinander stark, sie konvergieren schnell gegen einen festen Wert und dieser Wert liegt sehr nahe bei $1/2$.[16]

Wie in Abbildung 4 und Abbildung 5 zu sehen war, kann $V(T)$ für verschiedenartige Texte sehr unterschiedliche Formen annehmen. Vor diesem Hintergrund ist es eine bemerkenswerte Tatsache, dass sich die Kurven für Texte aus so verschiedenen natürlichen Sprachen derart ähnlich sehen. Im Verlauf dieser Arbeit wird sich der Eindruck weiter verstärken, dass $V(T)$ für natürlichsprachige Texte im Vergleich zur selben Größe für andere Textarten ein eigentümliches Verhalten zeigt.

Die Tatsache, dass $V(T)$ mit wachsendem T konvergiert und dass die Konvergenzwerte für die einzelnen Korpora so nahe beieinander liegen, deutet darauf hin, dass es einen Mechanismus im System der Sprache gibt, der dafür sorgt, dass V

[15]Kapitel 2.6

[16]Zwei Sonderfälle sind in Kapitel 3.4 besprochen.

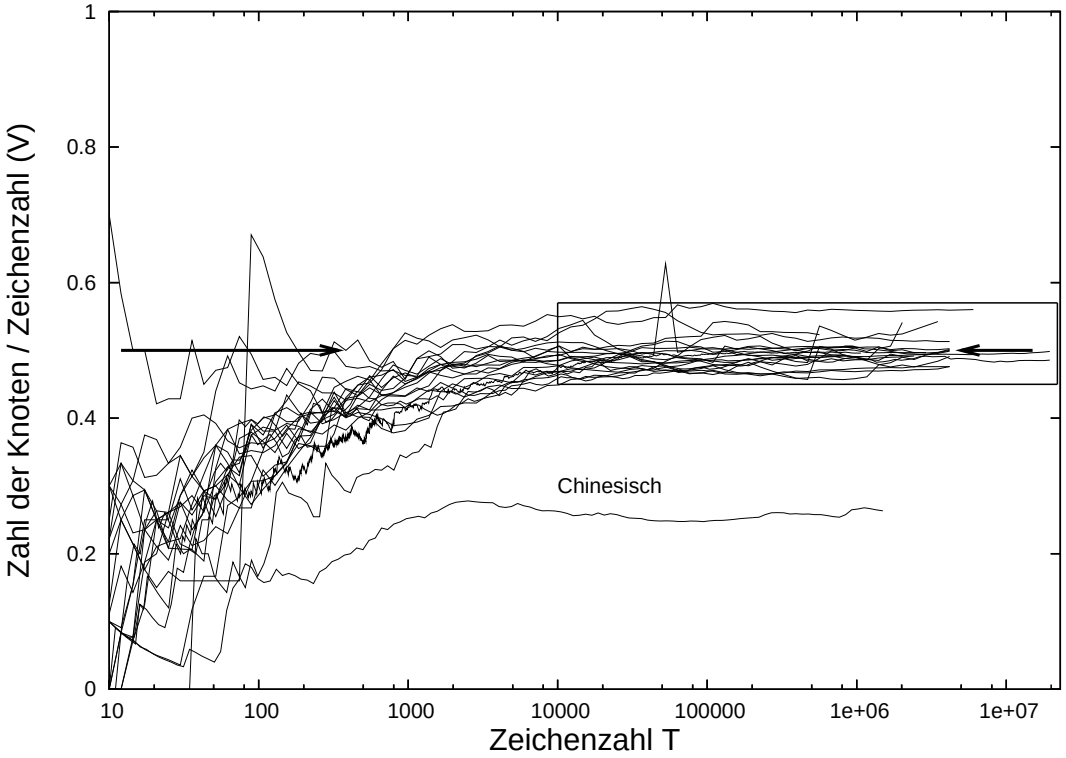


Abbildung 8: Übersicht zu allen betrachteten Sprachen. Auf der x-Achse ist die Textlänge T in logarithmischem Maßstab aufgetragen, das heißt, jeder Abschnitt auf der x-Achse entspricht einer Verzehnfachung der Textlänge. Auf der y-Achse ist das Verhältnis $V = K/T$ aufgetragen, wobei K die Zahl der (inneren) Knoten im Suffixbaum ist.

gegen $1/2$ strebt. Er könnte ein ökonomisches Prinzip verwirklichen, das erfordert, dass in einer Sprache nicht zu viel und nicht zu wenig Wiederholungen auftreten. Es ist plausibel, dass dieser Mechanismus in den verschiedenen Sprachen gleichartig ist. Ob es einen solchen Mechanismus tatsächlich gibt, auf welcher Ebene er arbeitet und wie er theoretisch zu beschreiben und experimentell genauer zu fassen ist, liegt nicht mehr im Rahmen dieser Arbeit.

$V(T)$ konvergiert für natürlichsprachige Texte nicht nur gegen einen einheitlichen Wert, die Konvergenz geht auch außerordentlich schnell vonstatten. Bereits nach etwa 10.000 Zeichen –das entspricht drei Seiten dieser Arbeit– verlaufen die Kurven innerhalb des engen Bereiches von 0.50 ± 0.05. Dies unterscheidet Texte natürlicher Sprachen ebenfalls stark von anderen Zeichenketten. In Abbildung 5 war $V(T)$ für Texte gezeigt, die eigens so erzeugt wurden, dass sie als Ganzes ein vorgegebenes V_{global} realisieren. Wie die übrigen Kurven dort verläuft auch diejenige für $V_{global} = 1/2$ bis kurz vor Ende des Textes deutlich unter $1/2$, bevor sie plötzlich nach oben abknickt. Das frühe Einsetzen der Konvergenz für natür-

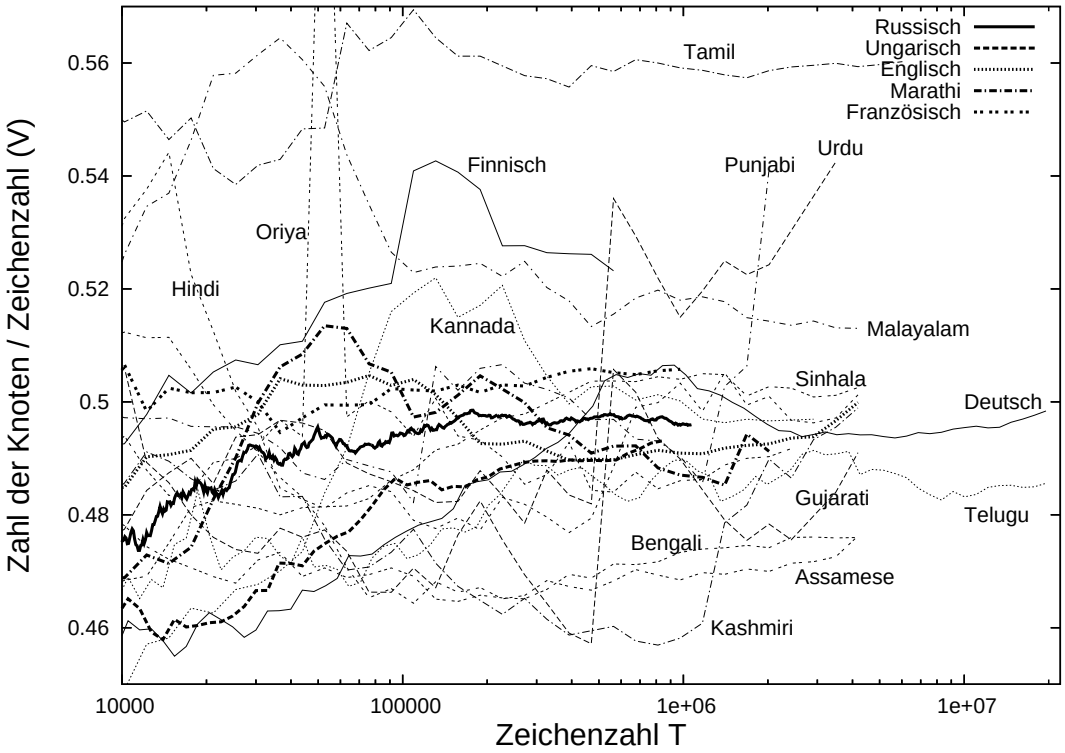


Abbildung 9: Auch hier ist $V(T)$ für alle 21 betrachteten Sprachen dargestellt. Die Art der Auftragung ist identisch zu Abbildung 8. Zu sehen ist der dort eingezeichnete Ausschnitt.

lichsprachige Texte bedeutet, dass sich das relative Maß an Wiederholungen (zur Bedeutung von V siehe Kapitel 2.5 auf Seite 10) bereits für sehr kurze Texte an das Niveau beliebig großer Korpora annähert.

Aber bereits ab dem Beginn der Messungen bei 10 Zeichen, also lange, bevor sie den Konvergenzpunkt erreichen, ist für die natürlichsprachigen Korpora deutlich ein mittleres $V(T)$ zu erkennen, wo die Kurven trotz des starken statistischen Rauschens besonders dicht beieinander verlaufen. Dieses einheitliche Verhalten bereits für sehr kurze Texte deutet stark darauf hin, dass es zwischen den Bestandteilen der Texte Wechselwirkungen mit sehr kurzer Reichweite gibt. Mit *Bestandteilen* sind hier Zeichenketten beliebiger Länge bezeichnet. Diese Lokalität des untersuchten Phänomens wird in Kapitel 4.4 noch einmal unter einem anderen Blickwinkel thematisiert werden.

Da $V(T)$ als ein relatives Maß für die Wiederholungen im Text bis zur Länge T interpretiert werden kann[17], ist auf der anderen Seite das uniforme Verhalten von $V(T)$ für längere und sehr lange Texte ein Indiz dafür, dass auch zwischen weit auseinanderliegenden Textteilen Wechselwirkungen bestehen, die V beeinflussen.

[17] Kapitel 2.5 auf Seite 10

Art und funktionale Abhängigkeiten für diese Wechselwirkungen zu formulieren wird einen wichtigen Ansatzpunkt für Folgeprojekte darstellen.

Abgesehen davon, dass die Kurven sich nicht völlig decken, verlaufen einige (z.B. für das Russische) sehr viel glatter als andere (Urdu). Dies erkennt man besser in der in Abbildung 9 dargestellten Vergrößerung des in Abbildung 8 markierten Ausschnitts.

Wie wir aus Kapitel 2.5 wissen, werden Sprünge in V durch längere Wiederholungen im Text hervorgerufen. Das Auftreten solcher Sprünge in vielen der indischen Korpora bestätigt die in Kapitel 3.1.1 formulierten Vorbehalte gegen diese Korpora. Keiner, der von uns als besonders vertrauenswürdig eingestuften Korpora (Englisch, Ungarisch, Deutsch, Russisch und Französisch) weist derartige Unregelmäßigkeiten auf. Leider gehören diese fünf Sprachen nur zwei der vier insgesamt untersuchten Sprachfamilien an.

Der Verlauf von $V(T)$ für diese Untermenge an Sprachen ist in Abbildung 10 dargestellt. Alle fünf Kurven verlaufen innerhalb eines mit wachsendem T immer schmaler zulaufenden Bereiches, dessen Spitze für sehr große T bei $1/2$ endet.

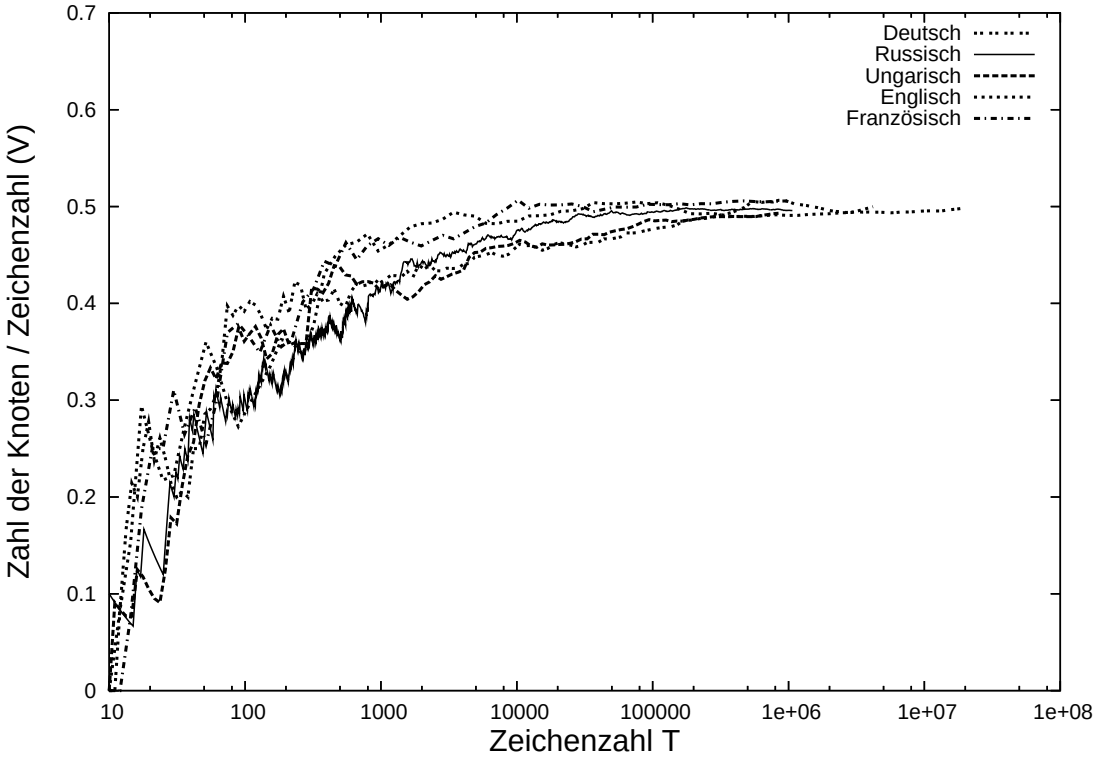


Abbildung 10: Der Verlauf von $V(T)$ für diejenigen Sprachen, deren Korpora als besonders vertrauenswürdig und für unsere Zwecke passend eingestuft wurden.

3.4 Sonderfälle

Es gibt zwei Sonderfälle. Chinesisch hat einen extrem niedrigen, Tamil einen ziemlich hohen Konvergenzwert.

3.4.1 Chinesisch

Die chinesische[18] Schrift besitzt als Logosyllabery[19] ein fundamental unterschiedliches Verhältnis zum gesprochenen Chinesisch, als alle anderen untersuchten Sprachen.

Bei allen Schriftsystemen, bis auf Logosyllaberies besteht eine relativ enge Beziehung zwischen Zeichen und Lauten der Sprache. Obwohl zum Beispiel im Englischen dasselbe Zeichen verschiedene phonetische Realisationen haben kann und umgekehrt, so besteht doch ein qualitativer Unterschied zum Chinesischen, wo ein und dieselbe phonetische Silbe je nach Kontext mit Hunderten von Schriftzeichen geschrieben werden kann.

Das chinesische Schriftsystem ist außerdem die einzige hier untersuchte Schrift, in der jedes Zeichen genau eine volle Silbe repräsentiert[20].

Auch die Gesamtzahl der Zeichen ist im Chinesischen um Größenordnungen höher als in den übrigen Sprachen. Im verwendeten Korpus [LCMC2004] kommen 4609 verschiedene Zeichen vor, verglichen mit 76 Zeichen im englischen Browncorpus [Brown1998]. Für eine genaue Statistik der chinesischen Schriftzeichen, und eine kurze Diskussion der interessanten Details siehe Anhang A.1 und dort vor allem Abbildung 21.

Bei derart gravierenden Unterschieden zwischen dem chinesischen Skript und den Schriftsystemen aller anderen untersuchten Sprachen verwundert es nicht, dass sich diese Unterschiede auch an $V(T)$ ablesen lassen.

In der Tat liegt der Konvergenzwert für Chinesisch im Bereich von 0.25 und damit etwa 50% tiefer als für die übrigen Sprachen. Dass die Kurve des Chinesischen tiefer liegt, als die übrigen, ist nicht verwunderlich, da V das relative Maß an Wiederholungen im Text mißt. Da Chinesisch so viele Schriftzeichen besitzt, ist damit zu rechnen, dass sich weniger Wiederholungen finden als in Sprachen mit einem geringeren Zeichenreichtum.

Liegt aber die Abweichung tatsächlich lediglich in den Besonderheiten der chinesischen Schrift begründet, oder bildet die chinesische Sprache an sich eine Ausnahme? In diesem Fall wäre die Konvergenz von V gegen $1/2$ nicht mehr universal und die Aussagekraft des untersuchten Phänomens entsprechend geringer.

Wie können wir diese Frage entscheiden? *Pinyin* ist eine standardisierte romanisierte Umschrift für die traditionellen chinesischen Schriftzeichen. Sie verwendet

[18] Wenn in dieser Arbeit von der chinesischen Sprache die Rede ist, spreche ich ausschließlich vom Standardchinesischen.

[19] Kapitel 2.6.2 auf Seite 13

[20] Siehe [Daniels1996, 202] zu den wenigen Ausnahmen in der chinesischen Schrift.

genau die Zeichen des englischen Alphabets, weist ihnen aber teilweise ungewohnte Lautwerte zu. Die fünf Töne des Hochchinesischen werden durch vier diakritische Zeichen oder durch eine nachgestellte Zahl markiert[21]. Selbst verglichen mit überkommenen alphabetischen Schriften wie dem Englischen ist die Beziehung zwischen Zeichen und Aussprache ungewöhnlich eng: Die Abbildung von Schriftzeichen auf Pinyin ist eindeutig[22]. Daher ist die Übertragung von Schriftzeichen auf Pinyin auch leicht zu automatisieren.

Begrüßenswerterweise wurde der LCMC-Korpus in zwei Varianten veröffentlicht: Einmal mit herkömmlichen Schriftzeichen geschrieben, einmal in Pinyin[23]. So kann man $V(T)$ für diese zwei Schriftsysteme des Chinesischen direkt anhand desselben Textes gegenüberstellen.

Der Vergleich ist in Abbildung 11 dargestellt. Die x-Achse ist normiert. Dies ist notwendig, da in Pinyin der Lautwert eines traditionellen Schriftzeichen durch ein bis sechs Zeichen repräsentiert wird. Im LCMC-Korpus ist es eines mehr, da die Töne nicht durch Diakrite, sondern durch nachgestellte Ziffern kodiert werden. Auf der x-Achse ist daher der Anteil am Gesamttext aufgetragen, nicht die absolute Zeichenzahl.

Ab einem Wert von etwa einem Tausendstel des Textes verläuft die Kurve für Pinyin zwischen 0.5105 und 0.5255, also gut innerhalb des Bereiches, den auch die anderen Sprachen einhalten.

3.4.2 Tamil

Betrachtet man die Abbildungen 8 und 9 so fällt schnell auf, dass die Kurve für Tamil im Mittel deutlich höher liegt als alle anderen. Sie verläuft auch relativ glatt, vor allem im Vergleich zu den übrigen indischen Korpora. Der tamilische Korpus weist also keine extrem langen Wiederholungen auf, der V nach oben verschieben würde[24]. Auch sonst gibt es keinen Grund, die Qualität ausgerechnet dieses Korpus anzuzweifeln.

Am Beispiel des Chinesischen haben wir eben [25] gesehen, dass das Schriftsystem Einfluss auf $V(T)$ haben kann. Was sagt die Literatur zur Tamilischen Schrift?

Laut [Daniels1996, 426] hat die Tamilische Schrift denselben Ursprung wie die meisten anderen indischen Skripten:

> The Tamil writing system, called *tamiz ezuttu* 'Tamil letter', derives from the southern branch of Ashokan Brahmi. Its immediate predecessor, Granhta script, serves as the basis for the Tamil and Malayalam writing systems

[21] Der fünfte, neutrale, Ton bleibt oft unmarkiert.

[22] Die seltenen Ausnahmen betreffen die Zeichen des Chinesischen, für die mehrere Aussprachen existieren.

[23] Alle fünf Töne werden durch nachgestellte Zahlen dargestellt.

[24] Kapitel 2.5

[25] Kapitel 3.4.1

Dennoch gibt es einen fundamentalen Unterschied (ibid.):

> Tamil is written alphasyllabically (...). Virtually uniqe within Indic scripts *tamiz ezuttu* has evolved toward an alphabet in one respect: it has eliminated most conjuncts, placing consonant clusters in a linear string.

Zum Begriff *alphasyllabary* siehe die Typologie der Schriftsysteme in Kapitel 2.6.2. Was aber ist mit *conjunct* gemeint? Dazu [Daniels1996, 376]:

> A graphic "syllable" consisting of a cluster of two or more consonants followed by a vowel (type CCV, CCCV, etc.) requires that the consonants be joined together in a *conjunct* character to indicate the cancellation of the inherent *a* vowel of the preceding consonant(s), ...

Die aufeinanderfolgenden Konsonanten werden übereinandergeschrieben und so zu einem einzigen Zeichen, dem *conjunct* zusammengefasst. Dieses System wurde im Tamilischen Schriftsystem aufgegeben, die einzelnen Konsonanten werden wie in einer alphabetischen Schrift einzeln hintereinander geschrieben.

Die Schrift des Tamilischen ist also eng verwandt mit den übrigen indischen Schriften, hat jedoch eine singuläre Entwicklung durchlaufen, die diese nicht mitvollzogen haben.

Folgendes **Szenario** auf der Grundlage der auf Seite 21 aufgestellten Hypothese ist möglich: Die Schriftsysteme natürlicher Sprachen streben ein ökonomisches Gleichgewicht an, damit weder zu viel, noch zu wenig wiederholt wird. In der Konvergenz von $V(T)$ gegen $1/2$ äußert sich dieses Gleichgewicht. Tamil hat sich durch den Übergang von einer Abugida zu einer weitgehend alphabetischen Schrift ein Stück weit von diesem Gleichgewicht entfernt. Die Erhöhung von $V(T)$, die wir beobachten, ist eine Folge dieses Ungleichgewichtes.

Wenn diese Hypothese stichhaltig ist, sollte die Tamilische Schrift sich im Laufe der Zeit wieder auf ein V von $1/2$ zubewegen.

Um diese Vermutungen zu überprüfen sind folgende Teilfragen zu klären: Wann ist diese Umstellung geschehen, wie plötzlich ging sie vonstatten? Gab es einen bekannten oder feststellbaren Grund? Eine diachronische Untersuchung des Tamilischen könnte zeigen, ob und wie sich der Konvergenzpunkt mit der Zeit verschoben hat.

Aber auch ohne solch weiterführende Untersuchungen, die den Rahmen dieser Arbeit sprengen würden, bleibt es eine bemerkenswerte Tatsache, dass die einzige Sprache, deren $V(T)$ deutlich von dem der anderen Sprachen abweicht, in einer Schrift geschrieben wird, die in der einschlägigen Literatur klar als Ausnahme bezeichnet wird.

Die Besonderheit der tamilischen Schrift ist auch in Abbildung 20 zu erkennen, in dem die Zeichenhäufigkeitsverteilungen aller hier betrachteten Sprachen zusammengefasst sind: Die Zeichenverteilung des Tamilischen weist mit dem alphabetisch geschriebenen Englisch wesentlich mehr Ähnlichkeit auf als mit den indischen Schriften, zum Beispiel dem (eine Abugidaschrift verwendenden) Sinhala.

3.5 Zentrale Aussagen

Ich fasse das Bisherige zusammen und formuliere die zentralen Beobachtungen, Hypothesen und Fragen dieser Arbeit:

Beobachtung B1: Für alle untersuchten Korpora konvergiert das Verhältnis V von Knotenzahl K zu Zeichenzahl T mit wachsendem T gegen eine Konstante.

Beobachtung B2: Die Konvergenz setzt in allen Fällen bereits bei sehr kleinen Textlängen T ein. Für noch kürzere Texte ist trotz des statistischen Rauschens eine Mittelwertkurve für $V(T)$ klar erkennbar.

Beobachtung B3: Diese Konstante ist für die überwiegende Mehrheit der Korpora mit $1/2$ verträglich. Die beiden Ausnahmen Chinesisch und – in geringerem Maße – Tamil sind auch in anderer Hinsicht Sonderfälle[26].

Hypothese H1: B1 und B2 gelten für alle natürlichen Sprachen in ihrer schriftlichen Form.

Hypothese H2: B3 gilt für alle natürlichen Sprachen, deren Schriftsystem ein Alphabet, ein Abjad oder eine Abugida ist.

Die Beobachtungen B1, B2 und B3 sowie die Bildung der Hypothesen H1 und H2 stellen die zentralen Ergebnisse der vorliegenden Untersuchung dar. Deren allgemeine Gültigkeit an vertrauenswürdigen Korpora aus einem breiteren Spektrum verschiedener Sprachen aus unterschiedlichen Sprachfamilien zu testen wird eine wichtige Aufgabe für weiterführende Untersuchungen darstellen.

Das Phänomen, das durch die Beobachtungen B1, B2 und B3 beschrieben wird, bezeichne ich im Folgenden mit dem zusammenfassenden Begriff der V-Kongruenz.

Im weiteren Verlauf der Arbeit vergleiche ich einerseits V-Kongruenz mit dem Zipfschen Gesetz und andererseits das Verhalten von $V(T)$ für natürlichsprachige Texte und für nicht natürlichsprachige Texte.

[26] Kapitel 3.4

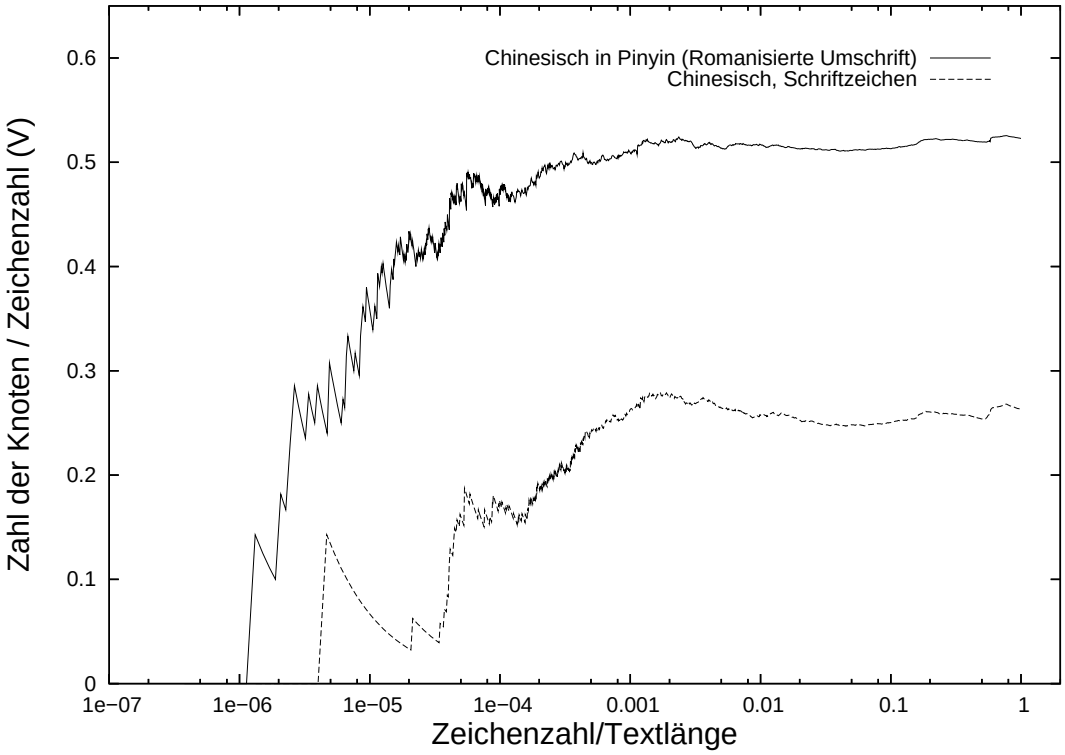


Abbildung 11: $V(T)$ für zwei Versionen des chinesischen Korpus. Die untere Kurve repräsentiert den Text in Schriftzeichen, die obere Kurve die Umschrift in Pinyin. Da durch die Transformation von Schriftzeichen zu Pinyin die Zahl der Zeichen im Text etwa um den Faktor 4 ansteigt, müssen für einen sinnvolen Vergleich gleiche Textstellen auf gleiche Punkte auf der x-Achse abgebildet werden. Daher ist auf der x-Achse nicht die absolute Zeichenzahl aufgetragen, sondern der Anteil am Gesamttext. Dass es sich beide Male um den selben Text handelt, kann man daran erkennen, dass sich vor allem im hinteren Teil der Kurve die Unregelmäßigkeiten in beiden Kurven synchron verhalten, abgesehen von der absoluten Größe der Schwankungen.

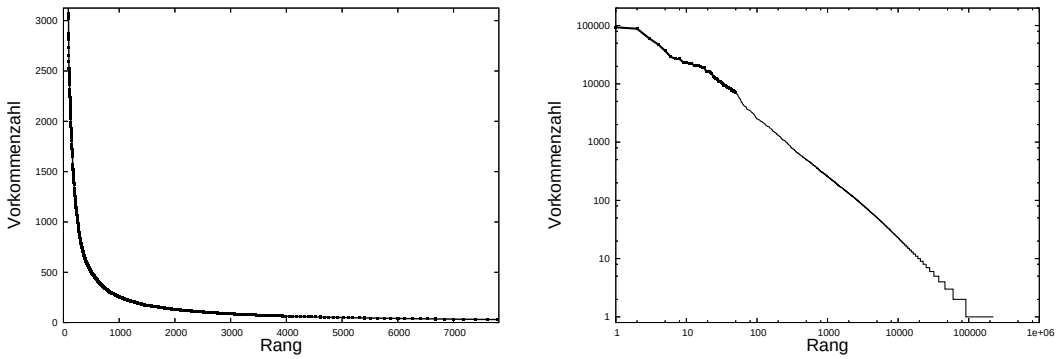
4 Vergleich mit dem Zipfschen Gesetz

In diesem Kapitel vergleiche ich zwei sprachstatistische Phänomene: Die Konvergenz von $V(T)$ gegen $1/2$ und das *Zipfsche Gesetz*. Kapitel 4.1 enthält eine kursorische Darstellung des Zipfschen Gesetzes. Kapitel 4.2 stellt den Zusammenhang zwischen der Konvergenz von V und dem Zipfschen Gesetz her und motiviert einen Vergleich. Das technische Vorgehen und die experimentellen Ergebnisse sind in Kapitel 4.3 dargestellt. Kapitel 4.4 faßt die Ergebnisse des Vergleichs zusammen.

4.1 Das Zipfsche Gesetz

Folgende empirische Tatsache ist als das Zipfsche Gesetz (auch: Zipf-Mandelbrot-Gesetz) bekannt ([Zipf1949], [Mandelbrot1954]): Man geht aus von der sortierten Frequenzliste der in einem Text vorkommenden Oberflächenformen: Das häufigste Wort im Korpus steht an erster Stelle, gefolgt vom zweithäufigsten, und so fort. Der Platz eines Wortes in dieser Liste wird oft als sein Rang bezeichnet.

Trägt man nun den Rang jedes Wortes auf der x-Achse und seine Häufigkeit auf der y-Achse auf, so ergibt sich (per definitionem) eine streng monoton fallende Kurve. In Abbildung 12(a) ist diese Konstruktion für den deutschen Korpus zu sehen.



(a) (b)

Abbildung 12: Ein Beispiel für das Zipfsche Gesetz. In beiden Teilbildern sind dieselben Daten dargestellt. Grundlage ist die Frequenzliste der Oberflächenformen des deutschen Korpus. Aufgetragen ist jeweils die Häufigkeit der Worte über ihrem Rang (siehe Text). Die Achsen in Teilbild (a) sind linear, während in Teilbild (b) beide Achsen logarithmischen Maßstab haben (Zur logarithmischen Darstellung vergleiche die Bemerkungen zu Abbildung 3 in Kapitel 2.4 auf Seite 8).

In doppellogarithmischer Darstellung (Abbildung 12(b)) liegen dieselben Da-

ten auf einer Geraden. Dies bedeutet[27], dass sich die Kurve durch eine Gleichung der Form

$$h(r) = cz^{-b} \text{ mit } b, c > 0 \tag{1}$$

beschreiben läßt. Die Tatsache, dass für natürliche Sprachen b immer sehr nahe bei 1 liegt, wird als das *Zipfsche Gesetz* bezeichnet. c ist eine Normierungskonstante.

Das Zipfsche Gesetz hat seinen Namen von Georg Kingsley Zipf, der es als Baustein einer weiterreichenden Theorie erstmals einer breiteren Öffentlichkeit zu Gehör brachte ([Zipf1949]). Wie so oft gab es auch hier Vorläufer.

4.2 Die Motivation für einen Vergleich

Das Zipfsche Gesetz ist wohl das bekannteste sprachstatistische Phänomen seiner Art. Daher sind Unterschiede und Gemeinsamkeiten zur Konvergenz von V gegen $1/2$ interessant, die vielleicht helfen, das neue Phänomen anhand des bekannten besser einzuschätzen.

Das Zipfsche Gesetz betrifft die Häufigkeitsverteilung von Worten, V ist ein relatives Maß der Wiederholungen im Text (siehe Kapitel 2.5 auf Seite 10). Insofern treten beide Phänomene in einem ähnlichen Bereich auf. Dies macht einen Vergleich sinnvoll.

Die wesentliche Motivation des Vergleiches war aber, dem Verdacht zu begegnen, dass sich in der V-Kongruenz lediglich das Zipfsche Gesetz neu manifestiert.

4.3 Vorgehen und Ergebnisse

Die eben angesprochene Ähnlichkeit von Zipfschem Gesetz und V-Kongruenz wird noch ergänzt durch die Tatsache, dass sich beide Beobachtungen letztlich auf eine einzige Zahl reduzieren lassen: $V(T)$ konvergiert gegen $1/2$, während die Steigung b in Gleichung 1 in der Nähe von 1 liegt.

Die These, die im Folgenden widerlegt werden soll lautet:

> Die Besonderheiten des Verlaufs von $V(T)$ und seine Uniformität in verschiedensprachlichen Texten ist nur eine andere Erscheinungsform des Zipfschen Gesetzes. Wenn die Worte eines Textes entsprechend dem Zipfschen Gesetz verteilt sind, dann zeigt auch $V(T)$ den in den Beobachtungen B1, B2 und B3 (Kapitel 3.5 auf Seite 27) zusammengefaßten Verlauf.

Wir gehen aus vom russischen Korpus[28]. Seine Worte sind nach dem Zipfschen Gesetz verteilt(Abbildung 12) und $V(T)$ zeigt V-Kongruenz (z.B. Abbildung 10). Ich habe nun schrittweise Umordnungen am Text vorgenommen, die die Häufigkeitsverteilung der Worte jeweils unberührt ließen.

[27] Der Beweis ist elementar.

[28] Kapitel 2.6.3

1. $V(T)$ für den unberührten Korpus ist in Abbildung 13 als Kurve (1) eingezeichnet.

2. In einem ersten Umordnungsschritt bleibt die innere Struktur der Worte intakt, aber ihre Reihenfolge wird randomisiert, d.h. durch eine zufällige Permutation ersetzt. Es ergibt sich Kurve (2) im selben Bild. Sie liegt bereits deutlich unter dem Originalkorpus.

3. Wenn man zwar die Reihenfolge der Worte untereinander unverändert läßt, aber die Buchstaben, aus denen sie sich zusammensetzen randomisiert, ergibt sich Kurve (3). Es wurden jeweils gleiche Oberflächenformen durch die immer gleiche Folge von Buchstaben ersetzt. Diese wurde für das erste Vorkommen zufällig erzeugt. Dabei war jeder Buchstabe gleich wahrscheinlich.[29]

4. In einem letzten Schritt wurden beide Randomisierungen gleichzeitig angewendet. Es ergibt sich Kurve (4).

5. (Kurve (5) entstand aus einem rein zufällig erstellten Text. Dieser ist nicht zipfverteilt. Die Kurve wurde als Kontrast und logische Fortführung in das Bild aufgenommen. Dies ist ein Vorgriff auf Kapitel 5. Dort werden wir auch eine Erklärung für die in den Kurven (3), (4) und (5) auftretenden Schwingungen finden.)

Der Vergleich zeigt:

Randomisierung führt immer zu einer Reduzierung von V. Dies ist verständlich, da sich bei einer zufälligen Reihenfolge von Worten oder Buchstaben auch nur zufällige Wiederholungen ergeben. Da Sprache das Einhalten gewisser wiederkehrender Strukturen vorschreibt, erwartet man, dass das willkürliche Zerstören dieser Strukturen das Maß an Wiederholungen im Text herabsetzt.

Solange die innere Struktur der Worte erhalten bleibt, bleibt auch die Konvergenz bestehen.

Sobald aber die Zeichen innerhalb der Worte durcheinandergemischt werden, ergeben sich Schwingungen im Kurvenverlauf. Auch ist die Herabsetzung von V in diesem Fall wesentlich größer. Dem gegenüber ist der Einfluß von Veränderungen der Reihenfolge der Worte untereinander ein untergeordneter Effekt. Den Schwingungen in $V(T)$ werden wir in Kapitel 5 wiederbegegnen.

4.4 Folgerungen

Es ergeben sich zwei wichtige Folgerungen:

Es ist ohne weiteres möglich, einen Text zu konstruieren, der zwar eine zipfverteilte Wortstatistik aufweist, für den sich $V(T)$ aber deutlich anders verhält

[29] Für mein Experiment habe ich eine Alphabetgröße von 62 Zeichen verwendet.

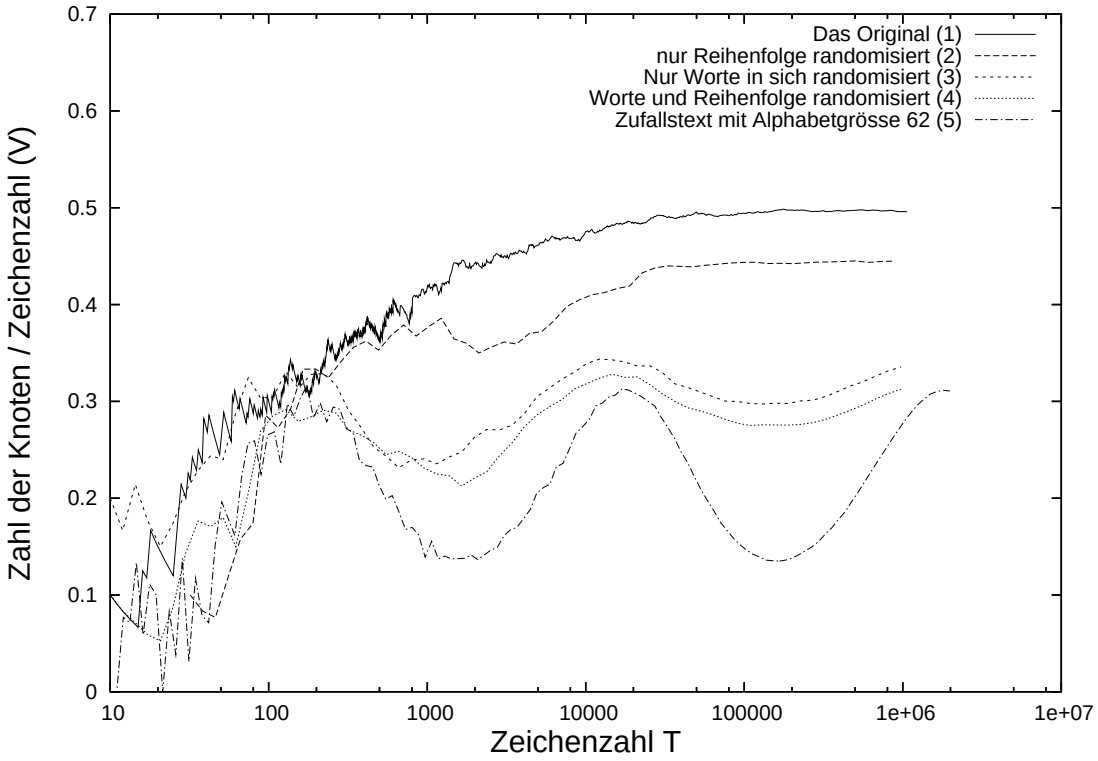


Abbildung 13: Vom natürlichsprachigen Text zum zufällig erstellten Text: die oberen vier Kurven reproduzieren exakt die Worthäufigeitsverteilung des russischen Korpus (Schuld und Sühne). (1) Der Originaltext. (2) Die Reihenfolge der Worte ist randomisiert, die Worte in sich sind intakt. (3) Die Reihenfolge der Worte bleibt unangetastet, aber die Worte sind in sich durch Zufallsketten derselben Länge ersetzt (Dieselbe Kette für jedes Vorkommen derselben Oberflächenform). (4) Sowohl die Reihenfolge der Worte, als auch die Worte in sich sind randomisiert wie in (2) und (3). (5) ist ein gleichverteilter Zufallstext der Alphabetgröße 62.

als für natürlichsprachige Texte (vergleiche B1, B2 und B3 in Kapitel 3.5 auf Seite 27). Die Hypothese zu Beginn von Kapitel 4.3 auf Seite 30 ist damit widerlegt. Die bei natürlichsprachigen Texten beobachtete V-Kongruenz ist nicht nur eine neue Erscheinungsform des Zipfschen Gesetzes, es handelt sich um ein eigenständiges Phänomen.

In Kapitel 3.3 auf Seite 22 wurde bereits aus dem Verlauf von $V(T)$ für natürlichsprachige Texte der Schluss gezogen, dass es Wechselwirkungen zwischen den Bestandteilen des Textes zu geben scheint, die über Entfernungen von einigen wenigen Zeichen wirken und in der Lage sind, $V(T)$ für unterschiedliche Sprachen auch bei sehr kurzen Texten uniform zu halten. In anderen Worten: V-Kongruenz ist ein lokales Phänomen. Dies wird auch im Vergleich zum Zipfschen Gesetz deutlich: Wie im vorhergehenden Abschnitt erwähnt, haben Veränderungen der

internen Struktur der Worte wesentlich mehr Einfluß auf $V(T)$ als Veränderungen auf der Ebene der Wortreihenfolge.

5 Zufallstexte

Als *Zufallstext* bezeichne ich eine zufällige Aneinanderreihung von Zeichen. D.h. jedes Zeichen kommt an jeder Stelle im Text mit der gleichen Wahrscheinlichkeit vor, unabhängig von den benachbarten Zeichen[30]. Ein Zufallstext dessen Zeichen alle dieselbe Vorkommenswahrscheinlichkeit haben, wird im Folgenden als *gleichverteilter Zufallstext* bezeichnet. Entsprechend heißen alle Zufallstexte, die diese Bedingung nicht erfüllen, *nicht-gleichverteilt*. Einen nicht-gleichverteilten Zufallstext, der so erzeugt wurde, dass er dieselbe Häufigkeitsverteilung aufweist wie eine natürliche Sprache, nenne ich einen *simulierenden Zufallstext*. Soll Bezug auf eine bestimmte Sprache X genommen werden, so spreche ich von einem *X-simulierenden Zufalltext*.

In diesem Kapitel untersuche ich Zufallstexte in Bezug auf den Verlauf von $V(T)$ und das Zipfsche Gesetz. Es wird gezeigt, dass $V(T)$ in Zufallstexten deutlich anders verläuft als in natürlichsprachgen Texten: Nur dort findet man die charakteristische schnelle Konvergenz gegen 1/2 (V-Kongruenz). Das Zipfsche Gesetz dagegen tritt nicht nur in natürlichsprachigen Texten auf, sondern auch in simulierenden Zufallstexten.

Dieser Befund ist ein Indiz dafür, dass V-Kongruenz natürlicher Sprache eigentümlich ist. Das Zipfsche Gesetz dient als illustrierendes Gegenbeispiel.

5.1 $V(T)$ für Zufallstexte

Im folgenden Abschnitt wird $V(T)$ für gleichverteilte und simulierende Zufallstexte untersucht. Anschließend vergleiche ich die Ergebnisse mit dem Verlauf von $V(T)$ für natürlichsprachige Texte.

5.1.1 $V(T)$ für gleichverteilte Zufallstexte verschiedener Alphabetgrößen

Das Alphabet eines gleichverteilten Zufallstextes sei die Menge der in ihm vorkommenden Zeichen. Seine Mächtigkeit sei mit α bezeichnet.

Abbildung 14 zeigt den Verlauf von $V(T)$ für gleichverteilte Zufallstexte mit α zwischen 3 und 100. Die Kurven nehmen insgesamt den Bereich zwischen 0,1 und 0,75 ein. Sie sinken mit steigender Alphabetgröße immer weiter ab. Dies veranschaulicht wiederum die Bedeutung von V als das relative Maß der Wiederholungen im Text (Kapitel 2.5): Je größer das Alphabet eines Zufallstextes, desto weniger Wiederholungen gibt es und desto kürzer sind die wiederholten Zeichenketten.

[30]Die verwendeten Zufallstexte wurden mit der in den üblichen c++ und Perlbibliotheken enthaltenen rand()-Funktion erstellt.

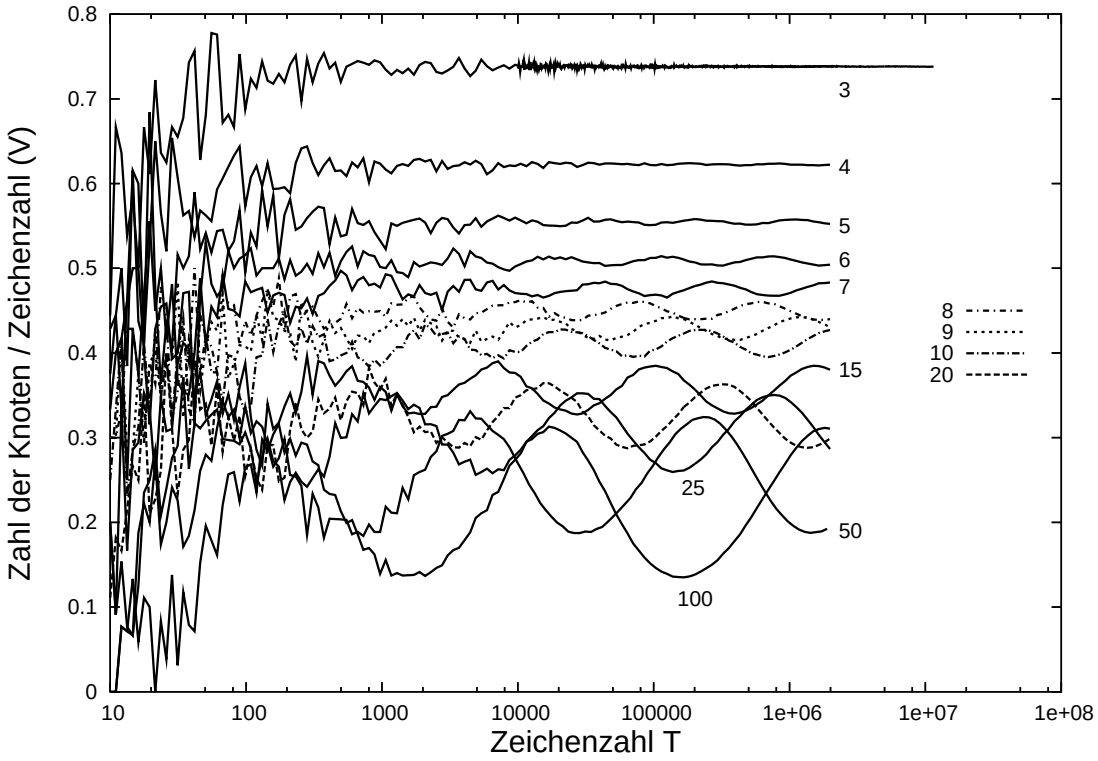


Abbildung 14: Der Verlauf von $V(T)$ für gleichverteilte Zufallstexte verschiedener Alphabetgrößen.

Die folgenden sechs Bemerkungen beschreiben die wesentlichen Charakteristika der Kurven. Ihre Gültigkeit ist meist mit bloßem Auge zu erkennen.[31]

1. Die Kurven gehen schnell in Sinusschwingungen über. Ab eine Alphabetgröße von 5 sind sie mit dem freien Auge erkennbar. Numerische Analyse zeigt ihre Existenz ab einer Alphabetgröße von 3. Diese Schwingungen in $V(T)$ werden in Anhang A.3 genauer begründet: Ihre Erklärung ist für diese Arbeit zwar von untergeordneter Bedeutung, aber sie sind auf den ersten Blick zu befremdlich, um sie unkommentiert zu lassen. Auch fördert die dortige Argumentation das allgemeine Verständnis für die Struktur von Suffixbäumen und für die Natur der Größe V.

2. Da auf der x-Achse der Logarithmus der Textgröße aufgetragen ist, gehorchen diese Schwingungen für ausreichend große T der allgemeinen Gleichung

$$V(T) = C + A\sin(b\ln(T) + \delta), \tag{2}$$

mit den reellwertigen Konstanten C, A, b und δ.

[31] Darüberhinaus habe ich zur Bestätigung numerische Analysen durchgeführt, auf die ich hier nicht weiter eingehen möchte, da Details für diese Arbeit nicht von Bedeutung sind.

3. C,A,b und δ in Gleichung 2 sind Funktionen von α.

4. $C(\alpha)$ ist eine streng monoton fallende Funktion.

5. $A(\alpha)$ ist eine streng monoton wachsende Funktion.[32]

6. $b(\alpha)$ ist eine streng monoton fallende Funktion. Meine numerische Analysen reichen nicht aus, um zu entscheiden, ob die Frequenz für $\alpha \to \infty$ gegen einen endlichen Wert oder gegen 0 strebt.

Insgesamt haben die V-Kurven für gleichverteilte Zufallstexte (Abbildung 14) wenig gemein mit $V(T)$ für natürliche Sprachen, vergleiche z.B. Abbildung 8.

5.1.2 $V(T)$ für simulierende Zufallstexte

Ein gleichverteilter Zufallstext ist ein sehr einfaches Sprachmodell. Daher war im Grunde nicht zu erwarten, dass sich V-Konvergenz auch dort findet.

Wenn wir die statistischen Eigenschaften des Modells an diejenigen natürlicher Sprachen anpassen, ist es realistischer, statistische Phänomene zu finden, die auch in natürlichsprachigen Texten auftreten.

Die auf Seite 34 eingeführten simulierenden Zufallstexte reproduzieren die Häufigkeitsverteilung natürlicher Sprachen. Ich beschränke mich darauf, beispielhaft $V(T)$ für Englisch- und Sinhala-simulierende Zufallstexte zu untersuchen. Die Zeichenstatistiken aller untersuchten Sprachen sind in Anhang A.1.2 graphisch zusammengestellt (Abbildung 20). Dort sind auch einige Bemerkungen zu den recht interessanten Charakteristika dieser Verteilungen zu finden.

Der Zeichenvorrat des Sinhala ist ungefähr doppelt so groß wie der des Englischen. Sinhala wird mit einer Abugida geschrieben[33]. Allgemein haben Abudidaschriften deutlich mehr Zeichen als Alphabetschriften, da jedes Zeichen ein Paar aus Konsonant und Vokal repräsentiert.

In Abbildung 15 ist $V(T)$ für Englisch- und Sinhala-simulierende Zufallstexte dargestellt. Zum Vergleich sind die Kurven für den englischen und den sinhalesischen Korpus eingezeichnet.

Die V-Kurven der simulierenden Zufallstexte sind den Kurven der natürlichsprachlichen Texte qualitativ viel ähnlicher als dies bei gleichverteilten Zufallstexten der Fall war. Insbesondere treten die charakteristischen Sinusschwingungen hier nicht auf. Dies ist intuitiv verständlich[34]: Man kann sich einen nicht-

[32] (4) zusammen mit (5) bedeutet, dass die Kurven für wachsende Alphabete immer weiter absinken, gleichzeitig aber die Amplitude der Schwingungen steigt. Da die Zahl der Knoten im Suffixbaum – und damit V – nicht negativ werden kann, impliziert dies, dass es eine untere Schranke für C geben sollte. C ist der Nullpunkt der Schwingung. Dies bedeutet, dass V im Mittel auch für sehr große α einen endlichen Wert behält. Damit dies für unendlich große Alphabete keine begrifflichen Probleme gibt, sollte die Amplitude b in diesem Fall gegen 0 gehen. Genau wäre dies nur mit analytischen Rechnungen für $V_\alpha(T)$ zu überprüfen.

[33] Kapitel 2.6.2

[34] siehe auch Anhang A.3, gegen Ende

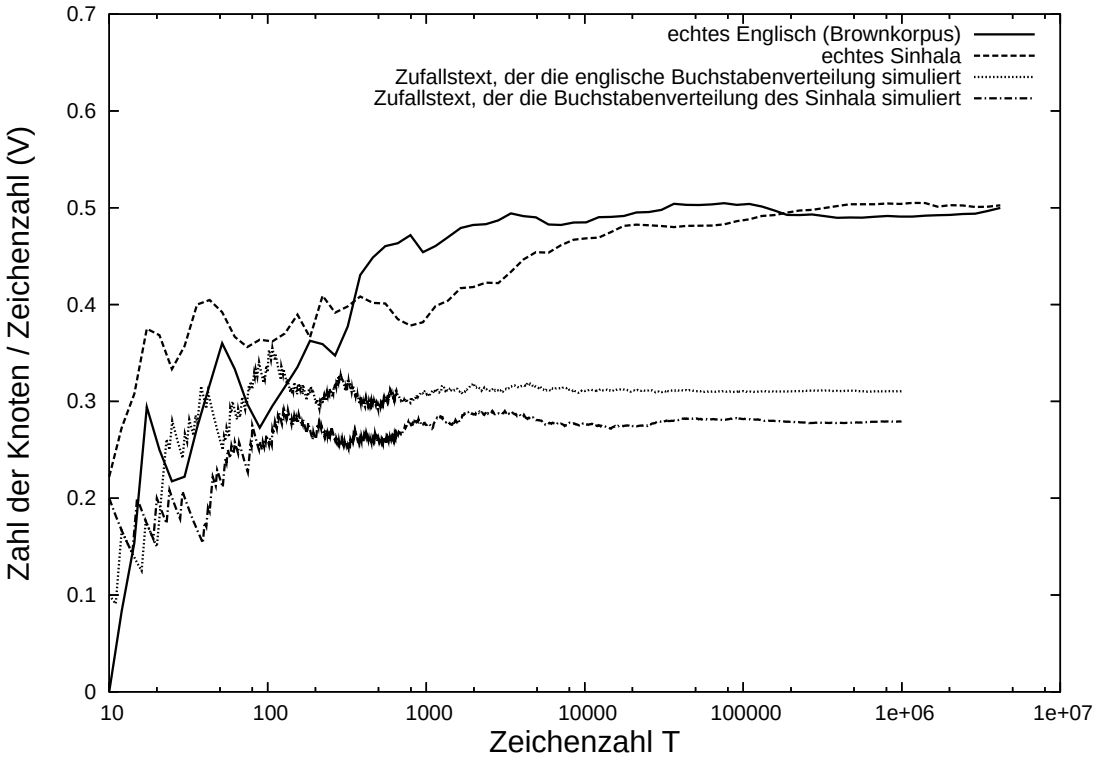


Abbildung 15: Die beiden oberen Kurven sind aus natürlichsprachigen Texten erstellt worden. Die beiden unteren Kurven gehören zu Zufallstexten, deren Zeichenstatistik diejenige der natürlichen Sprachen widerspiegelt.

gleichverteilten Zufalltext auch aus der Vermischung mehrerer gleichverteilter Zufalltexte verschiedener Alphabetgröße entstanden denken.

Nun scheint es plausibel, dass sich $V(T)$ als Überlagerung von Schwingungen verschiedener Frequenz, Wellenlänge und Amplitude darstellt. In der graphischen Darstellung ist nur der mehr oder weniger konstante Mittelwert dieser Überlagerung zu erkennen.

Wie für natürlichsprachigen Texte konvergiert $V(T)$ für simulierende Zufallstexte schnell gegen einen festen Wert (vergleiche B1 und B2 in Kapitel 3.5 auf Seite 27). Oben wurde begründet, warum die –verglichen mit gleichverteilten Zufallstexten– höhere Ähnlichkeit zum Verhalten natürlichsprachiger Texte im Einklang mit den Erwartungen ist.

Es gibt aber auch wesentliche Unterschiede zwischen den Kurven für simulierenden Zufallstexten und für natürlichsprachige Texte.

Ein wichtiger Unterschied ist der niedrigere Konvergenzpunkt. Für natürliche Sprachen liegt er bei 0.5, für die beiden untersuchten simulierenden Zufalltexte lediglich um 0.3. Dies bedeutet, dass das relative Maß der Wiederholungen[35] in diesem Fall niedriger ist. Dies ist bereits auf sehr niedrigem Niveau verständlich,

[35]Kapitel 2.5

da alle natürlichen Sprachen Worte bilden, die sich als feste Einheiten wiederholen. In diesem Text zum Beispiel wiederholt sich das Wort "Konvergenz" mit Sicherheit öfter (23 mal), als dies in einem Zufallstext vergleichbarer Länge (etwa 100.000 Zeichen) je der Fall sein könnte.

Ein weiterer Unterschied ist, dass die V-Kurven der beiden simulierenden Zufallstexte wesentlich weiter voneinander entfernt sind, als die der entsprechenden natürlichen Sprachen: Der englisch-simulierende Text liegt deutlich über dem Sinhala-simulierenden, während für natürliches Englisch und Sinhala kein klar erkennbarer Unterschied existiert.

Da es sich um Zufallstexte mit verschieden großem Zeichenvorrat handelt, ist der Unterschied in der Höhe der Kurven verständlich: Je mehr Zeichen, desto weniger Wiederholung und desto tiefer V (Kapitel 5.1.1 auf Seite 34). Deswegen liegt Sinhala (großer Zeichenvorrat) unter Englisch (kleiner Zeichenvorrat) Erstaunlich ist umgekehrt die Tatsache, dass die Differenz in der Größe des Zeichenvorrats im Falle natürlichsprachige Texte zu keinem signifikanten Unterschied im Verlauf von $V(T)$ führt.

5.2 Das Zipfsche Gesetz für Zufallstexte

Bereits 1992 wurde von Li Wentian [Li1992] das Zipfsche Gesetz für Zufalltexte untersucht. Hier werden seine Ergebnisse ergänzt und eine grundlegende Idee von damals zu Ende geführt.

Das Zipfsche Gesetz für Zufallstexte zu untersuchen, setzt voraus, ein spezielles Zeichen als Leerzeichen zu interpretieren und den entstehenden Text an diesem Zeichen in Stücke zu unterteilen. Diese Stücke werden dann als Worte interpretiert und ihre Verteilung untersucht. Zur besseren Unterscheidung von natürlicher Sprache spreche ich im Folgenden meist von *Pseudoworten*.

5.2.1 Gleichverteilte Zufallstexte

Normalerweise wird das Zipfsche Gesetz graphisch dargestellt wie in Kapitel 4.1 auf Seite 29 erläutert: Über dem Rang der Worte ist ihre Häufigkeit aufgetragen.

Um die Darstellung für unsere Zwecke klarer zu machen, habe ich für Abbildung 16 eine andere Auftragung gewählt: Über der Häufigkeit der Pseudoworte auf der x-Achse ist die Zahl der Pseudoworte mit dieser Häufigkeit auf der y-Achse aufgetragen. Beide Achsen haben logarithmischen Maßstab.

Für einen zipfverteilten Text ergibt sich wiederum eine Gerade, wie der Verlauf der Kurve für den deutschen Korpus zeigt.

wie in Abbildung 16 erkennbar, sieht die Kurve für gleichverteilte Zufallstexte vollkommen anders aus. Statt einer glatten Kurve treten einzelne hohe Maxima hervor, die mit wachsender Häufigkeit (weiter rechts auf der x-Achse) immer schärfer und niedriger werden.

Dies erklärt sich folgendermaßen: Für einen Zufallstext der Alphabetgröße

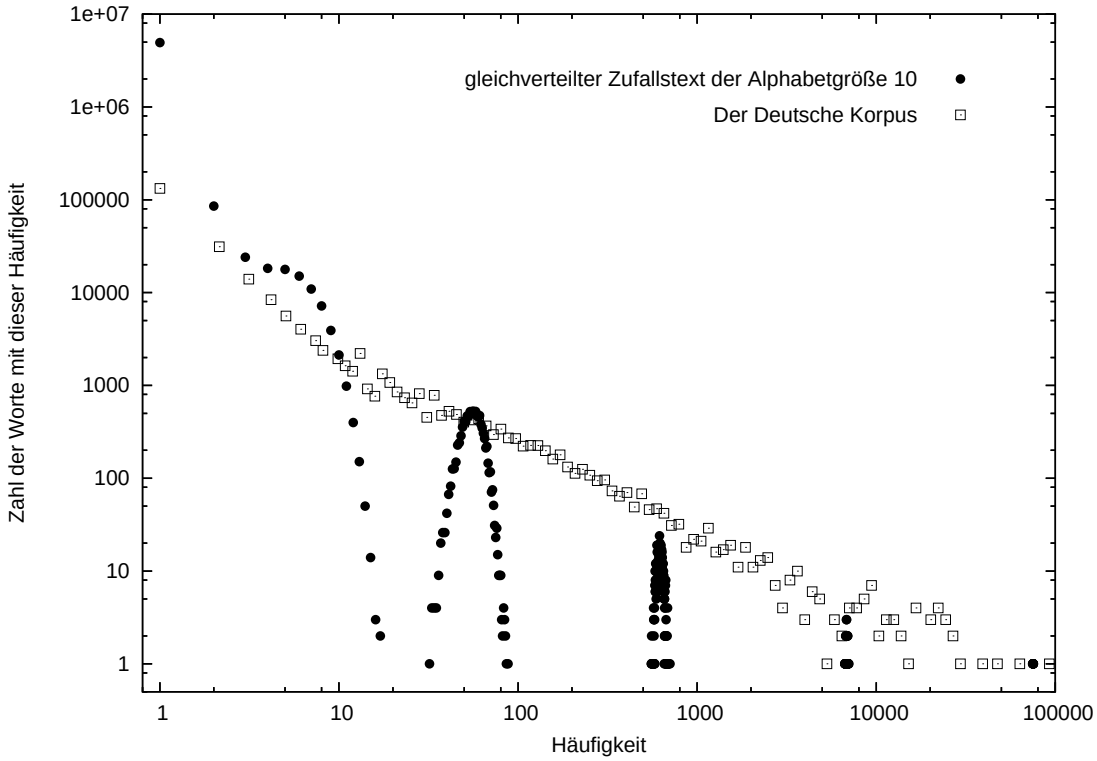


Abbildung 16: Das Zipfsche Gesetz für einen gleichverteilten Zufalltext der Alphabetgröße 10 im Vergleich zum Deutschen Korpus. Für andere Alphabetgrößen ergibt sich ein ähnliches Bild.

α gibt es genau $\alpha - 1$ Pseudoworte der Länge 1.[36] Damit das nächste Pseudowort im Text ein solches Einzeichenwort ist, muss auf ein Zeichen, das nicht das Leerzeichen ist, das Leerzeichen folgen. Die Wahrscheinlichkeit hierfür ist $P_1 = ((\alpha - 1)/\alpha)(1/\alpha) = (\alpha - 1)/\alpha^2$: Pseudoworte der Länge 2 gibt es $(\alpha - 1)^2$. Die Wahrscheinlichkeit für ihr Auftreten ist $P_2 = ((\alpha-1)/\alpha)^2(1/\alpha) = (\alpha-1)^2/\alpha^3$. Allgemein ist die Wahrscheinlichkeit P_n für das Auftreten eines Pseudowortes der Länge n nur von n und α abhängig und gleich $((\alpha - 1)/\alpha)^n(1/\alpha)$.

Ich verwendete eine Alpabetgröße von 10. Der letzte ausgefüllte Punkt ganz rechts bei einem Häufigkeitswert von etwa 100.000 in Abbildung 16 besteht in Wirklichkeit aus 9 verschiedenen Punkten, die ununterscheidbar nah beieinander liegen. Sie repräsentieren die 9 verschiedenen Einzeichenworte. Der nächste Peak bei etwa 7000 wird von den Zweizeichenworten gebildet, und so fort. da zwischen den Bildungswahrscheinlichkeiten für Pseudoworte der Länge n und $n+1$ immer der feste Faktor $(\alpha - 1)/\alpha$ besteht und es immer um den Faktor α mehr Pseudoworte der Länge $n + 1$ gibt, sind die Peaks auf der x-Achse jeweils um den selben Betrag gegeneinander verschoben, während sie in y-Richtung ebenfalls um

[36] Nicht α, da wir ein Zeichen als Trennzeichen definieren.

denselben Betrag wachsen.[37]

Da die relativen Häufigkeiten für seltene Ereignisse weiter streuen als für häufige, werden die Peaks immer breiter, wenn man auf der x-Achse nach links geht.

Das völlige Auflösen der Peakstruktur für Häufigkeiten kleiner als ca 9 kann man sich nach dem bisher gesagten leicht erklären: jenseits einer Häufigkeit von 1 gibt es noch unendlich viele weitere Peaks, die nicht direkt in Erscheinung treten können, da keine Häufigkeiten kleiner als 1 auftreten können. Die Ausläufer dieser immer breiter werdenden Peaks links der y-Achse erscheinen als die gleichmässige Häufung an Meßpunkten bei sehr kleinen Häufigkeiten.

5.2.2 Simulierende Zufallstexte

In [Li1992] wird zunächst auch das Zipfsche Gesetz für gleichverteilte Zufalltexte untersucht. Dort wird die übliche Auftragung (Häufigkeit über Rang) gewählt. So ergibt sich eine Kurve, die treppenartig in Stufen abfällt und sehr regelmäßig mal über und mal unter der Kurve für natürliche Texte liegt. In diesem eingeschränkten Sinne gilt das Zipfsche Gesetz tatsächlich auch für gleichverteilte Zufallstexte wie in [Li1992] behauptet. Dies hängt direkt mit der oben beschriebenen Tatsache zusammen, dass es α mal mehr Pseudoworte der Länge $n+1$ gibt als solche der Länge n, deren Vorkommen aber um den Faktor $\alpha/(\alpha-1)$ wahrscheinlicher ist.

Bereits in [Li1992] wird angeregt, nicht-gleichverteilte Zufallstexte zu untersuchen, da die beobachteten Stufen in diesem Falle verschwinden würden, da nun die Wahrscheinlichkeit eines Wortes nicht mehr lediglich von seiner Länge abhängt.

Der Gedanke wird in [Li1992] selbst aber nur rudimentär durchgeführt. Dort werden lediglich Zufallstexte mit Alphabetgrößen von 2 und 4 untersucht, deren Zeichenhäufigkeiten ungleich aber willkürlich verteilt sind. Bereits dieses Modell unterstützt aber deutlich die Vermutung, dass die Häufigkeitsverteilung der entstehenden Pseudoworte einer echten Zipfverteilung näher kommen.

Wir entwickeln die zugrundeliegende Idee hier weiter und stellen die Frage: Wieweit gehorchen Zufallstexte dem Zipfschen Gesetz, die die Zeichenverteilung natürlicher Sprache möglichst genau simulieren? Als Beispiel betrachten wir das Deutsche. Die Daten werden in Abbildung 17 mit der Zipfkurve des deutschen Korpus verglichen. Für diese Darstellung wird die für das Zipfsche Gesetz übliche Auftragung verwendet.

Die Ergebnisse lassen sich folgendermaßen zusammenfassen:

1. Über weite Strecken verlaufen beide Kurven nahezu parallel.

2. Wesentliche Abweichungen gibt es nur im Bereich sehr kleiner Häufigkeiten (rechts): Die deutsch-simulierende Kurve verläuft dort flacher als die natürliche Kurve. Diese macht zwischen Rang 1000 und 10000 eine kleine Biegung nach unten.

[37] Wir erinnern uns, dass durch den logarithmischen Maßstab gleiche Faktoren zwischen verschieden Werten jeweils in gleiche Abstände umgewandelt werden.

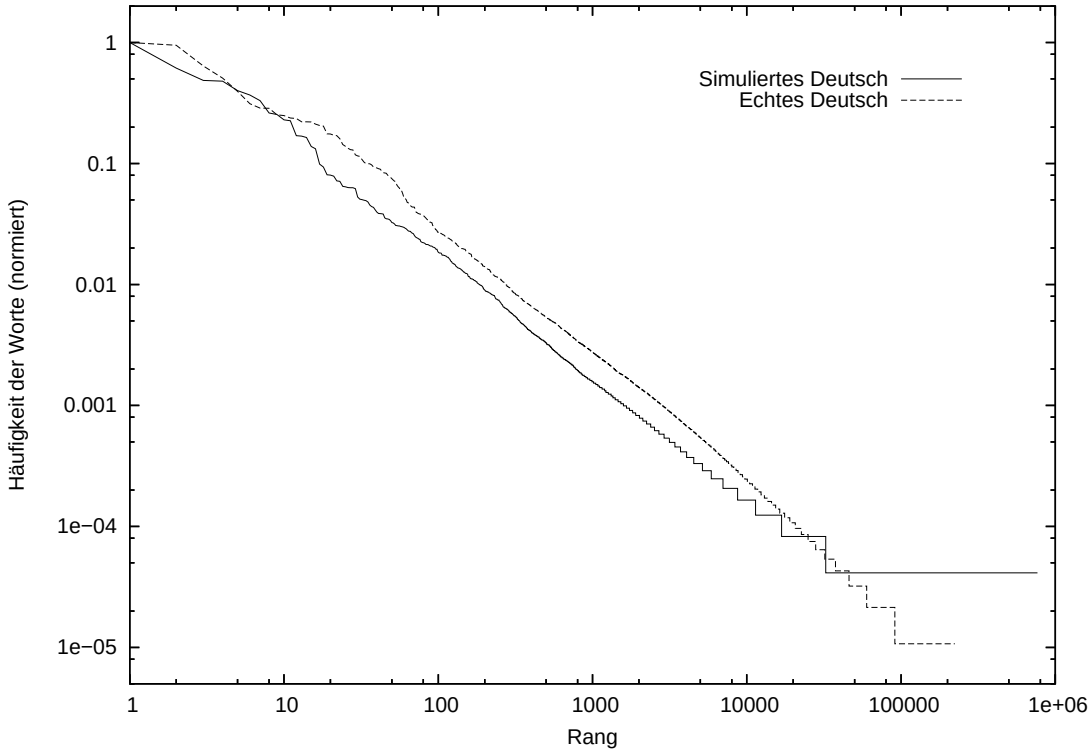


Abbildung 17: Das Zipfsche Gesetz für einen Deutsch-simulierenden Zufallstext, verglichen mit dem deutschen Korpus. Mit "Rang" ist die Nummer in der nach Häufigkeit sortierten Liste aller vorkommenden Oberflächenformen bezeichnet, bzw. das Äquivalent für einen Zufallstext. Die Häufigkeit ist normiert auf das jeweils häufigste Wort.

3. Bemerkenswert ist, dass in beiden Kurven etwa zwischen Rang 10 und 100 sehr ähnliche Ausbuchtungen nach oben finden. Ihr Schwerpunkt liegt im Falle des Zufallstextes bei etwas kleineren Rängen.

4. Die ersten (beiden) Worte treten im Deutschen Texte fast gleich häufig auf. Dies trift auf den simulierten Text nicht zu. Durch die Vergrößerung des Maßstabes für kleine Ränge infolge der logarithmischen Auftragung tritt dieser unbedeutende Unterschied übermäßig deutlich hervor.

Insgesamt kann man sagen: Das Zipfsche Gesetz lässt sich in beinahe unveränderter Form auch in simulierenden Zufallstexten beobachten. Damit ist die Vermutung von Li [Li1992] bestätigt und präzisiert.

Diese Beobachtung stellt die tiefere Bedeutung des Zipfschen Gesetzes insgesamt stark in Frage. Ein Phänomen, das sich mit einem so einfachen Sprachmodell reproduzieren läßt, scheint nicht besonders interessant. Untersuchenswert sind möglicherweise der in Punkt 2 beschriebene Knick und die Ausbuchtungen der Kurve im Bereich kleiner Ränge. (Punkt 3). Beides tritt in erstaunlich unver-

änderter Form in den Zipfkurven aller natürlichsprachigen Texte auf. Allgemein gibt es charakteristische Abweichungen der Zipfkurven natürlicher Sprachen von der idealen $1/r$-Form [38], die sich ebenfalls von Sprache zu Sprache wiederholen. Diese Beobachtung scheint mir interessanter als das Zipfsche Gesetz selbst.

Ich fasse die letzten beiden Kapitel in einem Resümee zusammen:

- V-Kongruenz und das Zipfsche Gesetz sind zwei klar unterscheidbare sprachstatistische Phänomene.
- Im Gegensatz zur V-Kongruenz läßt sich das Zipfsche Gesetz auch in simulierenden Zufallstexten beobachten.

[38] r ist der Rang

6 $V(T)$ für Programmcode

Bisher haben wir $V(T)$ für konstruierte Beispiele, für natürliche Sprache und für automatisch und zufällig erstellte Texte untersucht. Es bietet sich an, den Verlauf von $V(T)$ auch für formale Sprachen zu untersuchen. Programmcode ist mit Abstand der größte Einsatzbereich formaler Sprachen.

Ich untersuche drei Beispiele: Meinen eigenen c++ Code, der das Programm zum Erstellen von Suffixbäumen beschreibt, Perl-Code eines Internetprojektes und Teile der Linuxquellen[39]. In Analogie zum Vorgehen für natürlichsprachige Texte (Kapitel 3.1.2) wurden auch hier sämtliche Formatierungszeichen aus dem Text entfernt.[40] Die (größtenteils natürlichsprachigen) Kommentare dagegen habe ich beibehalten, um keine zu weitgehenden Veränderungen an den Originaldateien vorzunehmen. Die Ergebnisse werden in Abbildung 18 mit dem russischen Korpus

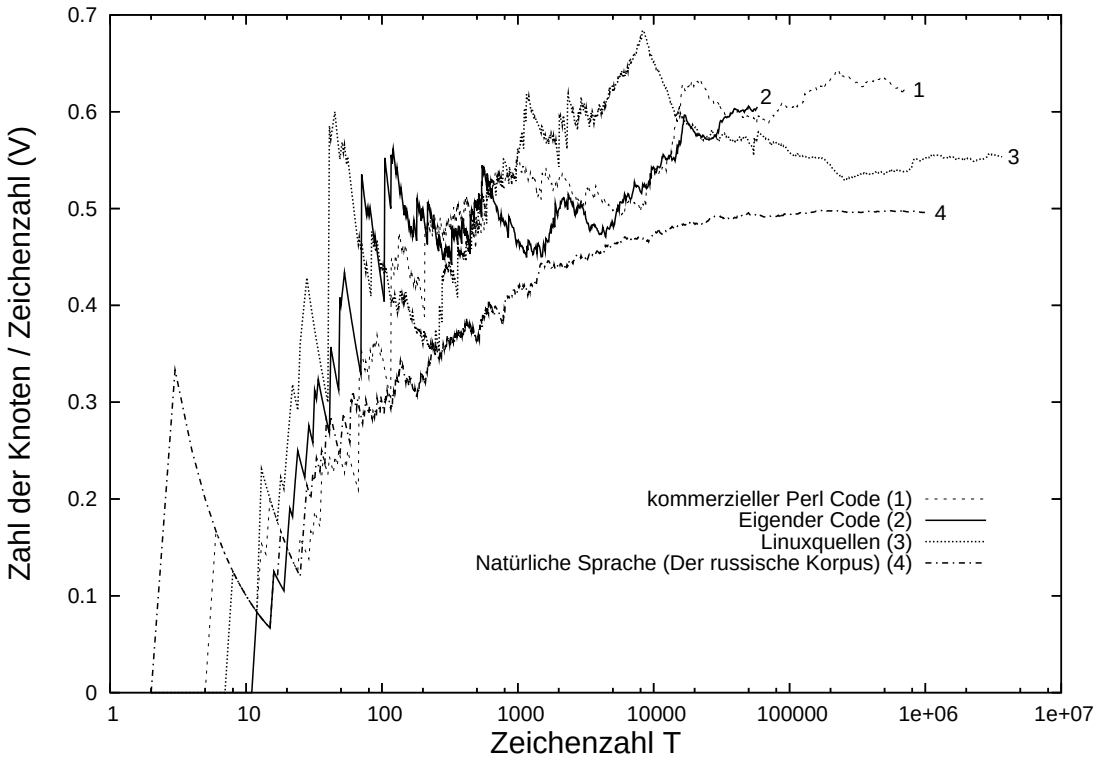


Abbildung 18: $V(T)$ für Programmcode im Vergleich mit natürlichsprachigem Text. Dargestellt sind drei Beispiele: Mein eigener c++ Code, Teile der Linuxquellen (auch c++) und der Code eines komerziellen Internetprojektes (Perl). Zum Vergleich ist $V(T)$ für den russischen Korpus ebenfalls eingetragen.

[39] der Inhalt aller .h-Dateien aus dem Verzeichnis /usr/src/linux-2.6.4-52/include/linux meiner SuSe 9.1 Distribution.

[40] Graphischer Vergleich zeigte, dass diese Veränderung des Originaltextes auch hier kaum einen Unterschied im Kurvenverlauf bedeutet.

verglichen. Zwei wichtige Unterschiede zwischen $V(T)$ für natürliche Sprache und für Programmtext sind festzustellen:

1. Alle drei Quellcodekurven liegen im Mittel deutlich über der Kurve des natürlichsprachigen Textes. Bereits nach etwa 90 Zeichen liegen sie vollständig darüber.

2. Im Gegensatz zur stetigen Konvergenz des russischen Textes gegen $1/2$ strebt keine der Quellcodekurven gegen einen eindeutigen Wert. Die Schwankungen werden mit wachsendem T nicht einmal eindeutig flacher.

$V(T)$ verhält sich also für Programmcodetext deutlich anders als für natürlichsprachige Texte.

Dass V im Mittel für Programmcode höher liegt als für natürliche Sprache ist leicht einzusehen: V ist ein relatives Maß für die Wiederholungen im Text. Von Variablennamen (und natürlichsprachigen Kommentaren) abgesehen ist der Wortschatz einer Programmiersprache klein und vorgegeben. Auch die Syntax ist starr. Daher ist mit einem höheren Maß an Wiederholungen zu rechnen.

Der Grund für die fehlende Konvergenz von V ist unklar. Es scheint eine bemerkenswerte Tatsache, dass es selbst innerhalb eines Projektes keinen eindeutigen Wert für das Ausmaß an Wiederholungen im Code zu geben scheint, in scharfem Gegensatz zu natürlicher Sprache, wo solch ein eindeutiger Wert bereits für sehr kurze Texte erkennbar wird.

7 Zusammenfassung

Im einleitenden Kapitel 2 wird der Begriff des Suffixbaumes eingeführt und die Größe V als das Verhältnis von Knotenzahl zu Textlänge definiert. Die 21 untersuchten Sprachen werden nach ihrer genetischen Verwandtschaft und nach den für sie verwendeten Schriftsystemen klassifiziert.

In Kapitel 3 werden die grundlegenden Korpusdaten zum Verhalten von $V(T)$ für natürlichsprachige Texte vorgestellt und die schnelle Konvergenz von $V(T)$ gegen $1/2$ als experimentelle Tatsache etabliert. Dieses Phänomen bezeichne ich als V-Kongruenz.

Kapitel 4 bestätigt V-Kongruenz als eine gegenüber dem Zipfschen Gesetz eigenständige Eigenschaft natürlichsprachiger Texte.

In Kapitel 5 weise ich nach, dass im Gegensatz zur V-Kongruenz das Zifpsche Gesetz nicht nur in natülichsprachigen Texten, sondern auch in bestimmten Arten von Zufallstexten auftritt. Damit kann die Aussagekraft des Zipfschen Gesetzes stark in Zweifel gezogen werden. Dieser Einwand gilt bezüglich der V-Kongruenz nicht.

Kapitel 6 zeigt, dass die $V(T)$-Kurve für Programmtexte wiederum einen fundamental unterschiedlichen Verlauf besitzt, im Vergleich mit natürlicher Sprache, aber auch im Vergleich mit Zufallstexten.

Damit erweist sich V-Kongruenz als ein für natürlichsprachige Texte eigentümliches Phänomen: Es ist im Rahmen meiner Untersuchungen unabhängig von Sprache und Schriftsystem[41] und tritt ausschließlich in natürlichsprachigen Texten auf.

Weiterführende Untersuchungen bieten sich an. Offene Fragen sind beispielsweise:

Findet sich V-Kongruenz auch in weiteren Sprachen aus unterschiedlichen Sprachfamilien, und in Sprachen, die mit bisher noch nicht behandelten Schriftsystemen geschrieben werden wie z.B. Koreanisch?

Was für eine Bedeutung hat die hier vorgestellte Gesetzmäßigkeit? Ist die Konvergenz von V die Manifestation eines inneren Gleichgewichts natürlicher Sprache und stellt der Konvergenzwert von $1/2$ ein Optimum dar, z.B. unter informationstheoretischen Gesichtspunkten?

Was für Eigenschaften muss ein Sprachmodell haben, das in der Lage ist, V-Kongruenz zu reproduzieren?

Darüberhinaus stellt sich die spannende Frage, ob Suffixbäume natürlichsprachiger Texte noch weitere interessante statistische und strukturelle Eigenschaften besitzen, die möglicherweise neue Zugänge zum Verständnis des Systems natürlicher Sprache insgesamt eröffnen.

[41] In gut verstandenen Ausnahmefällen ergeben sich Abweichungen im Konvergenzwert, Kapitel 3.4

A Anhänge

A.1 Detaillierte Statistiken und Ergebnisse für die einzelnen Sprachen

A.1.1 Der Verlauf von V

Der genaue Verlauf von $V(T)$ für die einzelnen Sprachen ist in Abbildung 8 und Abbildung 9 schwer zu erkennen. In Abbildung 19 auf Seite 48f sind die Einzelkurven übersichtlich zusammengestellt.

A.1.2 Zeichenstatistiken

In dieser Arbeit werden Zeichen (nicht etwa Worte) als die Grundeinheit eines Textes behandelt. Daher ist es interessant, die Häufigkeitsverteilung der Zeichen in den verschiedenen Sprachen zu betrachten. Sie sind für alle untersuchten Sprachen mit Ausnahme des Chinesischen in Abbildung 20 zusammengefaßt. Wie bei der üblichen Darstellungsweise des Zipfschen Gesetzes (Kapitel 4.1) ist auf der x-Achse der Rang[42] aufgetragen, auf der y-Achse die Häufigkeit der Zeichen. Die x-Achse ist in Abbildung 20 allerdings nicht logarithmisch, wie dies bei Darstellungen des Zipfschen Gesetzes gewöhnlich der Fall ist.

Allen Kurven ist der steile Abschnitt im Bereich sehr häufiger Zeichen gemeinsam. Numerische Analyse zeigt, dass sich die Kurven für Ränge $r < 10$ durch die Funktion $h(r) = const/x^b$ beschreiben lassen, mit einem b von etwa $-1/2$. Es handelt sich also um ein Potenzgesetz. Der in der einfachlogarithmischen Darstellung von Abbildung 20 lineare Verlauf zeigt an, dass der Abfall weiter rechts im Wesentlichen exponentiell verläuft.

Bemwerkenswert ist, dass die Menge der untersuchten Sprachen in zwei ziemlich scharf getrennte Klassen zerfällt:

1. Die Häufigkeitskurven der Sprachen der erste Klasse fallen ab Rang 10 in guter Näherung exponentiell ab. Sie besitzen etwa 100 bis 170 Zeichen.

2. Die Zeichenstatistiken der zweiten Gruppe sind im allgemeinen weniger glatt als die der ersten Gruppe. Auffällig ist der scharfe Abbruch der Kurven am unteren Ende. Sie haben nur 70 bis 100 Zeichen.

Zur ersten Gruppe gehören alle indischen Sprachen, bis auf Tamil, das eine klare Ausnahme bildet. Urdu ist ein Sonderfall: Vom Kurvenverlauf her eher Mitglied der ersten Gruppe, liegt sein Zeichenvorrat mit 102 Zeichen auf der Grenze zwischen den beiden Bereichen.

Zur zweiten Gruppe gehören alle europäischen Sprachen, einschließlich Russisch, Ungarisch und Finnisch. Auch Tamil mit seinem Zeichenvorrat von 69 Zei-

[42] Posistion in der sortierten Frequenzliste, siehe ebd.

chen und dem scharfen Abbruch der Statistik am unteren Ende gehört in diese Gruppe.

Diese Trennung ist insofern im Einklang mit der Typologie der Schriftsysteme wie in Kapitel 2.6.2 eingeführt, als alle Abugidas in die erste Gruppe fallen und alle Alphabetschriften in die zweite Gruppe.

Urdu, Kaschmiri und Punjabi werden jedoch nach [Daniels1996] weder mit einer Abugida, noch mit einem Alphabet geschrieben, sondern mit einem Abjad. Kaschmiri und Punjabi sind von den Abugidaschriften in Gruppe 2 kaum unterscheidbar, während Urdu keiner der beiden Gruppen voll angehört.

Dass die Zeichenstatistik der Tamilischen Schrift denen der europäischen Alphabetschriften ähnlicher ist, als den indischen Abugidas, ist gut vereinbar mit den in Kapitel 3.4.2 zitierten Bemerkungen aus [Daniels1996], denen zufolge das Tamilische sich von einer Abugida zu einem Alphabet entwickelt hat. Vergleiche auch 3.3 und 3.4.2.

Betrachtet man die Kurve für Englisch in einem vergrößerten Maßstab (siehe Abbildung 21), erkennt man eine weitere Besonderheit der Kurven aus Klasse 2: Exponentielle Abschnitte einer bestimmten Steilheit wechseln mit wesentlich steileren Abschnitten ab. Es ergibt sich ein Abfall in Stufen.

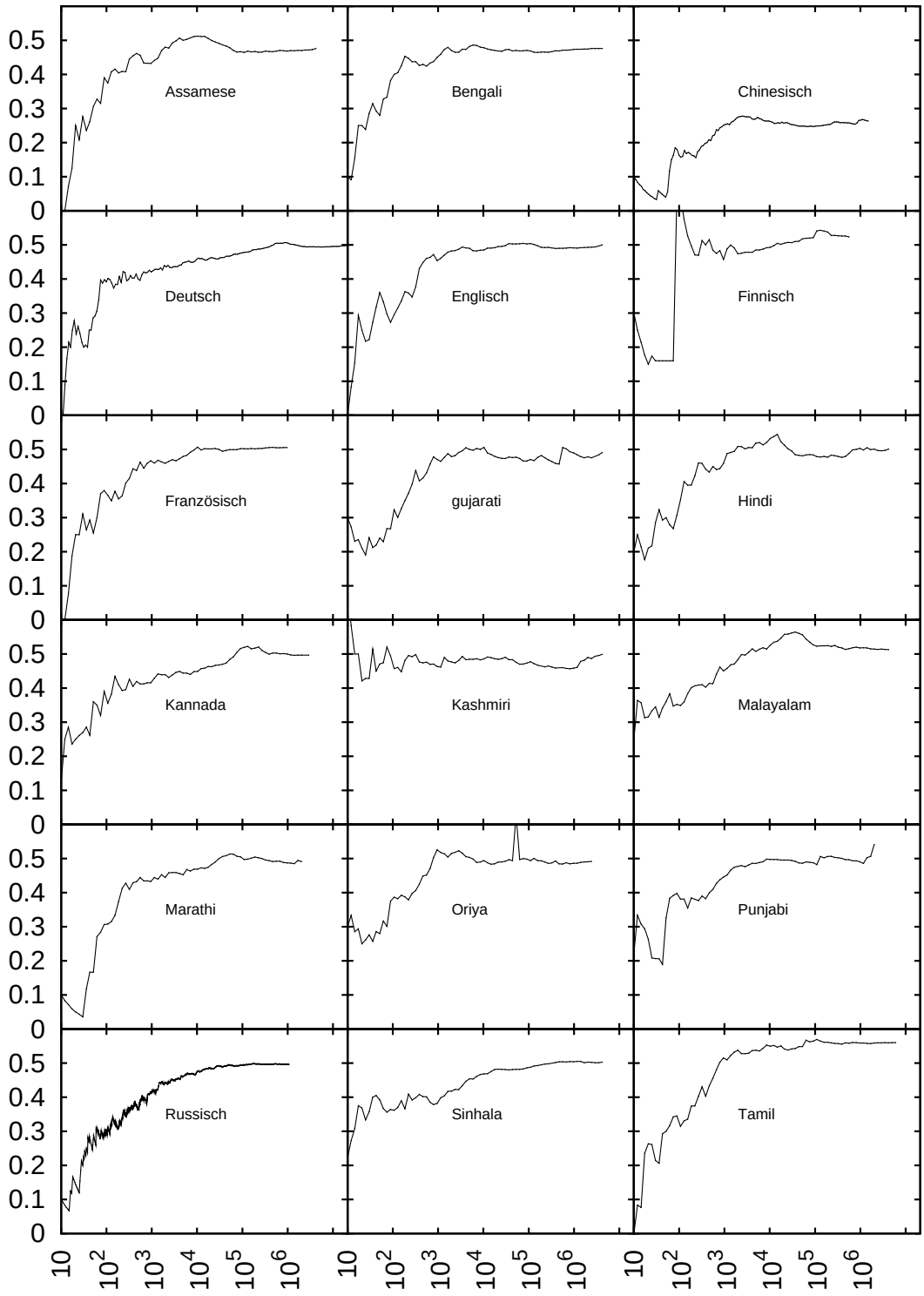
Wegen seiner einzigartigen Schrift betrachten wir die Zeichenverteilung des Chinesischen gesondert. In Abbildung 21 werden die Zeichenstatistiken der zwei Schriften, in denen uns der chinesische Korpus vorliegt (traditionell und Pinyin), mit zwei typischen Vertretern der Klassen 1 und 2 verglichen: Englisch und Sinhala.

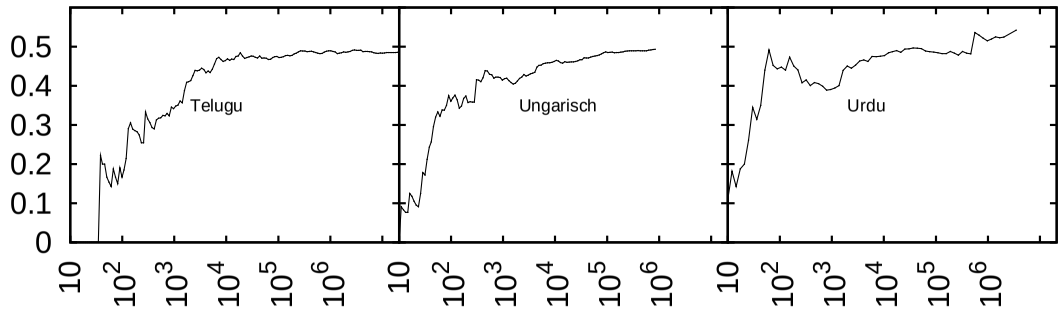
Auch die Kurve der traditionellen chinesischen Schriftzeichen besitzt den steilen Abschnitt im Bereich sehr kleiner Ränge. Der Übergang ins exponentielle ist hier aber weniger scharf ausgeprägt.

Die Verteilung der Zeichenhäufigkeiten für Pinyin teilt so gut wie keine Charakteristika mit den anderen Sprachen und mit den traditionellen Schriftzeichen. Weder gibt es den deutlichen exponentiellen Zerfall der Kurve im Bereich der seltenen Zeichen, noch den sehr steilen Bereich für hohe Frequenzen. Um so erstaunlicher, dass $V(T)$ für Pinyin sich doch so gut ins Bild der anderen Sprachen einpasst (siehe Kapitel 3.4.1).

Es ist eine naheliegende Vermutung, dass sich in der abweichenden Zeichenstatistik des Pinyin die Tatsache widerspiegelt, dass es sich um ein konstruiertes Schriftsystem handelt, im Gegensatz zu den gewachsenen Schriften des Englischen, des Sinhala oder der chinesischen Schriftzeichen.

Abbildung 19: Der Verlauf von $V(T)$ für alle untersuchten Sprachen im Überblick




0.5
0.4
0.3
0.2
0.1
0
Telugu
Ungarisch
Urdu
10
10^2
10^3
10^4
10^5
10^6

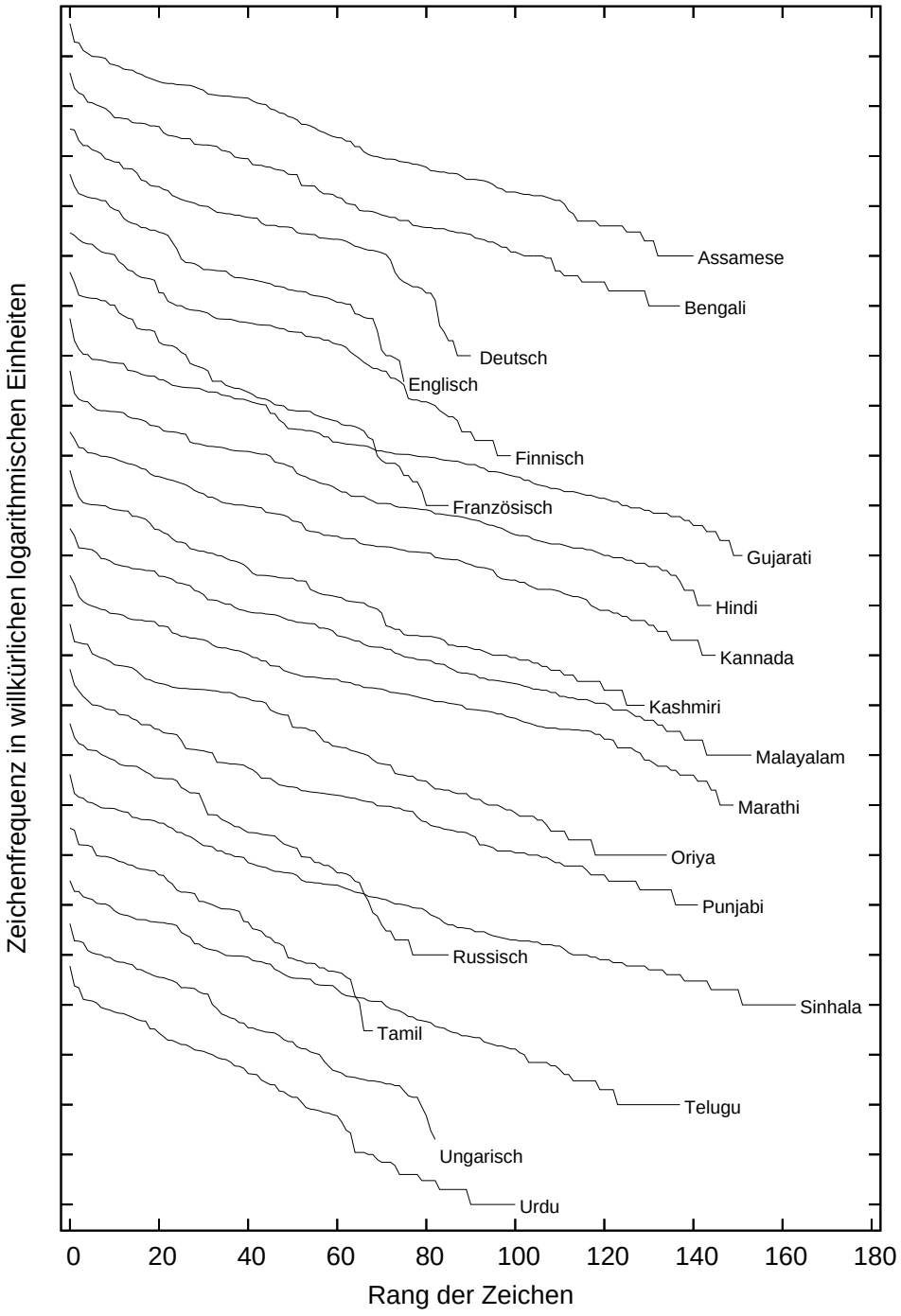


Abbildung 20: Die Zeichenstatistiken für alle betrachteten Sprachen mit Ausnahme des Chinesischen. Auf der x-Achse ist der Rang, auf der y-Achse die Häufigkeit aufgetragen. Die Kurven sind vertikal jeweils um den gleichen Abstand gegeneinander abgesetzt. Die Einheiten auf der y-Achse sind infolgedessen nicht absolut. Jeder Strich entspricht einem Abfallen der Häufigkeit um den Faktor 10.

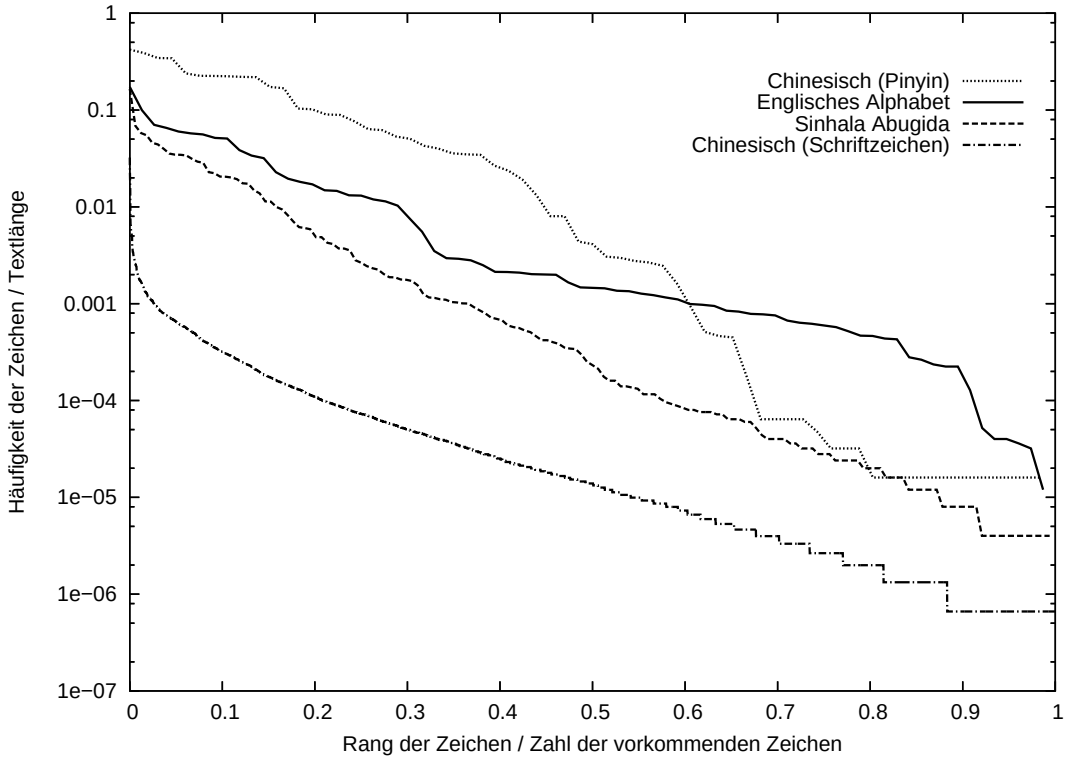


Abbildung 21: Die Zeichenstatistik für die beiden chinesischen Schriften (traditionell und die romanisierte Umschrift Pinyin), Englisch und Sinhala. Beide Achsen sind normiert: Auf der x-Achse ist der Rang der Zeichen dargestellt, geteilt durch die Zahl der Zeichen, die in den Korpora vorkommen. Die y-Achse zeigt die relative Häufigkeit der Zeichen.

A.2 $V(T)$ für extrem heterogene Texte

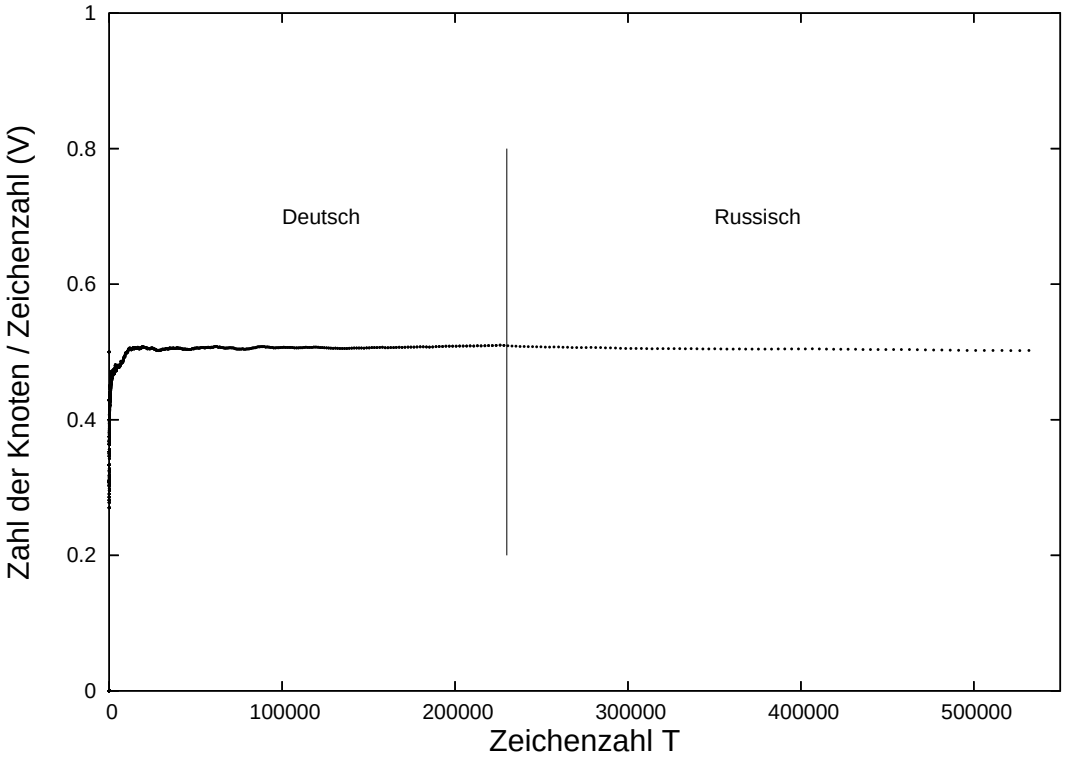


Abbildung 22: Wie verhält sich V, wenn sich der eingelesene Text aus zwei verschiedensprachigen Stücken zusammensetzt? Die x-Achse ist hier nicht logarithmisch, damit die zwei (fast) gleichgroßen Hälften gleichen Raum einnehmen.

Wie verhält sich $V(T)$, wenn ein natürlichsprachiger Text extrem heterogen ist, d.h. wenn etwa in einem größtenteils deutschen Text ein längeres französisches Zitat enthalten ist? Bei automatisch erstellten Korpora können solche "Verunreinigungen" schnell entstehen. Intuitiv würde man vielleicht erwarten, dass $V(T)$ an solchen Bruchstellen einen Sprung macht.

Abbildung 22 zeigt, dass das Gegenteil der Fall ist: $V(T)$ verläuft völlig glatt für einen Text, dessen erste Hälfte dem deutschen Korpus entstammt, die zweite Hälfte aber dem russischen. Damit beide Hälften in der Graphik den selben Raum einnehmen, ist der Maßstab der x-Achse nicht wie gewöhnlich logarithmisch.

Dieses Verhalten kann man sich leicht erklären: Da sich Zeichenketten aus dem deutschen Teil im russischen nicht wiederfinden und umgekehrt, zerfällt der aus dem gesamten Text erstellte Suffixbaum bereits an der Wurzel in zwei getrennte Unterbäume, einen russischen und einen deutschen. Für beide Teilbäume ist das Verhältnis V von Knotenzahl zu Zeichenzahl (des russischen bzw. des deutschen Teils) $1/2$. Dies gilt also auch für den gesamten Suffixbaum.

A.3 Informelle Begründung der Schwingungen in $V(T)$ für gleichverteilte Zufallstexte

Knoten werden während dem zeichenweisen Einlesen des Textes genau dann in den Suffixbaum eingebaut, wenn sich eine Zeichenkette wiederholt und mit dem aktuellen Zeichen auf neue Art fortgesetzt wird.

Ein Beispiel: Die Alphabetgröße α sei 25. Nehmen wir weiter an, dass wir gerade ein Zeichen eingelesen haben und nun an einer Stelle im Text sind, in der die Kette der letzten fünf Zeichen bereits früher vorkam. Diese fünf Zeichen lange Kette heiße s. Wir müssen genau dann einen Knoten einbauen, wenn das folgende Zeichen im Text noch niemals als Fortsetzung von s auftrat.

s kann bisher gar nicht, 1 mal oder öfter vorgekommen sein. Entsprechend viele Fortsetzungen von s kann es bisher gegeben haben. Die Wahrscheinlichkeit dafür, dass im folgenden Schritt ein neuer Knoten entsteht, ist umso größer, je kleiner die Zahl der bisherigen Fortsetzungen von s ist. Sie ist daher maximal, wenn bisher nur eine einzige Fortsetzung für s auftrat. Dies ist dann am wahrscheinlichsten, wenn s selbst bisher genau einmal vorkam.

Für ausreichend kurze Texte es aber viel wahrscheinlicher, dass s noch gar nicht auftrat. Mit größer werdender Textlänge wächst die Wahrscheinlichkeit, dass s genau einmal vorkam. Für extrem große Texte wird dieser Fall dann wieder sehr unwahrscheinlich, da nun fast jede Zeichenkette der Länge 5 mehr als einmal vorkommt.

Die Wahrscheinlichkeit, dass s oder jede andere Kette der Länge 5 genau einmal vorkommt, hat demnach ein Maximum bei einer bestimmten Textlänge T_5. Zu diesem Zeitpunkt müssen sehr viele Knoten in den Suffixbaum eingebaut werden, bei denen genau fünf Zeichen an den Kanten auf dem Weg zur Wurzel stehen. Sie haben also die Zeichentiefe 5, siehe Seite 8.

Wenn die Wahrscheinlichkeit maximal ist, eine Zeichenkette der Länge 5 genau einmal im Text zu finden, wird dieselbe Wahrscheinlichkeit für Zeichenketten der Länge 6 sehr klein sein: Jedes Vorkommen einer Zeichenkette der Länge 5 kann nur mit einem einzigen Buchstaben fortgesetzt werden, also nur zu einer Zeichenkette der Länge 6 führen. Es existieren aber α^5 Zeichenketten der Länge 5 und α^6 Zeichenketten der Länge 6. Die Wahrscheinlichkeit, eine willkürlich ausgewählte Kette der Länge 6 genau einmal zu finden ist also von der Größenordnung α (in unserem Beispiel 25) kleiner als dieselbe Wahrscheinlichkeit für Zeichenketten der Länge 5.

Es entstehen zu diesem Zeitpunkt also sehr wenig Knoten der Zeichentiefe 6 im Baum. Die maximale Entstehungsrate für Knoten dieser Art ist erst dann erreicht, wenn der Text α mal länger ist ($T_6 = \alpha T_5$). Dann sind aber beinahe alle Knoten der Zeichentiefe 5 bereits im Baum vorhanden: Wenn eine Zeichenkette in einem Text der Länge T_5 mit maximaler Wahrscheinlichkeit 1 mal auftritt, wird sie in einem 25 mal längeren Text normalerweise mehr als einmal enthalten sein.

In Texten der Länge T mit $T_5 < T < T_6$ dagegen dagegen entstehen weder

besonders viele Knoten der Zeichentiefe 5, noch solche der Zeichentiefe 6: Die einen gibt es bereits fast alle, für die anderen fehlt noch die Voraussetzung, dass die entsprechenden Zeichenketten bereits einmal im Text vorkamen. Dasselbe gilt erst recht für Knoten der Zeichentiefe Z mit $Z < 5$ oder $Z > 6$.

Diese Argumentation ist nicht nur auf Zeichenketten der Länge 5 und 6 beschränkt, sondern gilt allgemein.

Die Tatsache, dass die Maxima in der Entstehungsrate neuer Knoten der Zeichentiefen Z und $Z + 1$ jeweils für Textlängen T_Z und T_{Z+1} auftreten mit $T_{Z+1} = \alpha T_Z$, führt zu den beobachteten $sin(ln(T))$-Schwingungen in $V(T)$ für gleichverteilte Zufallstexte (siehe Abbildung 14).

Die Sinusform der Kurven lässt sich nur mit analytischen Rechnungen exakt begründen. Intuitiv ist aber einsehbar, dass es sich um eine streng periodische Funktion handeln sollte: Die Verhältnisse sollten bei ϵT_{Z+1} sein wie bei ϵT_Z, wobei ϵ eine Zahl zwischen 0 und α sei. Der Sinus ist aber die einfachste periodische Funktion ohne Sprünge.

In nicht-gleichverteilten Zufallstexten treten die Schwingungen nicht auf (siehe Bild Abbildung 15), da die Wahrscheinlichkeit, dass eine Zeichenketten der Länge n genau m mal im Text vorkommt, nicht für alle Zeichenketten der Länge n gleich ist, da die Häufigkeit der Zeichen unterschiedlich ist.

A.4 Pseudocode zu den wichtigsten Teilen des Algorithmus von Ukkonen

Der Implementierung liegt Gusfields [Gusfield1997] Zusammenfassung des in [Ukkonen1995] dargestellten Algorithmus zugrunde. Ich habe mich für rekursive Grundstruktur entschieden: Jeweils während ein Zeichen in den Suffixbaum eingebaut wird, wird die Schlüsselfunktion `phase()` rekursiv durchlaufen.

```
root_node is a node object
suffixlink of root_node = root_node
last_seen_node = root_node
pending_node = False
distance_from_last_seen_node = 0

for all characters in linear order do:
  init_phase(position_in_file)
  phase()
end for

subfunction init_phase(integer position_in_file):
    if(distance_from_last_seen_node < 0)
        distance_from_last_seen_node = 0
    endif
    # Blätter zeigen immer auf das Textende
    head_marker = position_in_file
    # Neu eingebaute Knoten enden eine Stelle vor der aktuellen
    # Textstelle:
    phase_head_marker = position_in_file - 1
    current_char = get_current_char(position_in_file)

subfunction get_current_char(pos):
    # Hole das Zeichen Nummer pos aus der Datei, in der der
    # einzulesende Text gespeichert ist.

subfunction phase():
    # wir sind direkt am letzten eingebauten knoten:
    if distance_from_last_seen_node < 0 then: # (1)
        set_suffix_link(None)
        if (last_seen_node has a transition starting with
            current_char) then:
            distance_from_last_seen_node ++
            return
        else # neuer uebergang
            add_new_leaf_to_last_seen_node()
```

```
            return follow_suffix_link()
        endif
    else: # da wir nicht am knoten sind, muss es bereits einen
          # Übergang geben, der mit dem richtigen Buchstaben
          # anfängt.
        transition = find_transition()

        # der Übergang ist kürzer, als der Weg, den wir vom in
        # den Baum einzusetzenden Zeichen entfernt sind: wir
        # können Skippen
        if (distance_from_last_seen_node
            - length(transition) >= 0) then: # (2)
            last_seen_node = endpoint of transition
            distance_from_last_seen_node -= length(transition)
            # Recursiver Aufruf:
            return phase()
        # wir müssen mitten im Übergang feststellen, ob das in
        # den Baum einzusetzende Zeichen bereits existiert:
        else: # (2)
            if match(transition): # (3)
                # wir müssen gar nichts tun, außer uns merken,
                # dass wir uns vom zuletzt gesehenen Knoten um
                # eine Stelle weiter entfernt haben.
                distance_from_last_seen_node ++
                return
            else: # wir müssen einen neuen Knoten in den Übergang
                  # einbauen
                new_transition = split(transition)
                set_suffix_link(new_transition)
                return follow_suffix_link()
            endif # (3)
        endif # (2)
    endif # (1)

subfunction find_transition:
    1) Gehe im Text distance_from_last_seen_node zurück.
    2) Merke dir das dort gesehene Zeichen.
    3) Suche am last_seen_node nach dem Übergang, der mit diesem
       Buchstaben beginnt.
    4) Gib diesen Buchstaben zurück.

subfunction match(transition):
    steht an der distance_from_last_seen_node'ten Stelle des
    Übergangs transition der aktuell in den Suffixbaum
    einzubauende Buchstabe?
```

```
    *Ja: return True
    *Nein: return False

subfunction split(transition):
    1) Baue einen Übergang U, der
       - an dem Knoten startet, an dem auch transition startet
       - der distance_from_last_seen_node Zeichen lang ist
    2) Füge U den alten Übergang transition als Verzweigung hinzu.
    3) Kürze das wurzelseitige Ende von Transition um
       distance_from_last_seen_node Zeichen
    4) Füge U ein Blatt als Verzweigung hinzu (Alle Blätter zeigen
       immer auf das aktuelle Textende)
    5) Gib U zurück

subfunction set_suffix_link(new_trans):
    if new_trans is not defined then:
        if pending_node is defined then:
            suffix_link of pending_node = last_seen_node
        endif
        undefine pending_node
    else: # new_trans ist definiert
        if pending_node is defined then:
            suffix_link of pending_node = endpoint of new_trans
        endif
        pending_node = endpoint of new_trans
    endif

subfunction follow_suffix_link():
    tmp = last_seen_node
    last_seen_node = suffix_link of last_seen_node
    if last_seen_node is identical with tmp then:
        # wir sind an der Wurzel im Kreis gegangen
        set_distance -= 1
    endif
    if distance_from_last_seen_node < 0 then: # wir haben nichts
                                              # mehr einzubauen
        return
    # sonst rekursiver Aufruf
    else:
        return phase()
    endif

subfunction add_new_leaf_to_last_seen_node:
    Add a new leaf to last_seen_node.
```

A.5 Handelt das Zipfsche Gesetz von Worten?

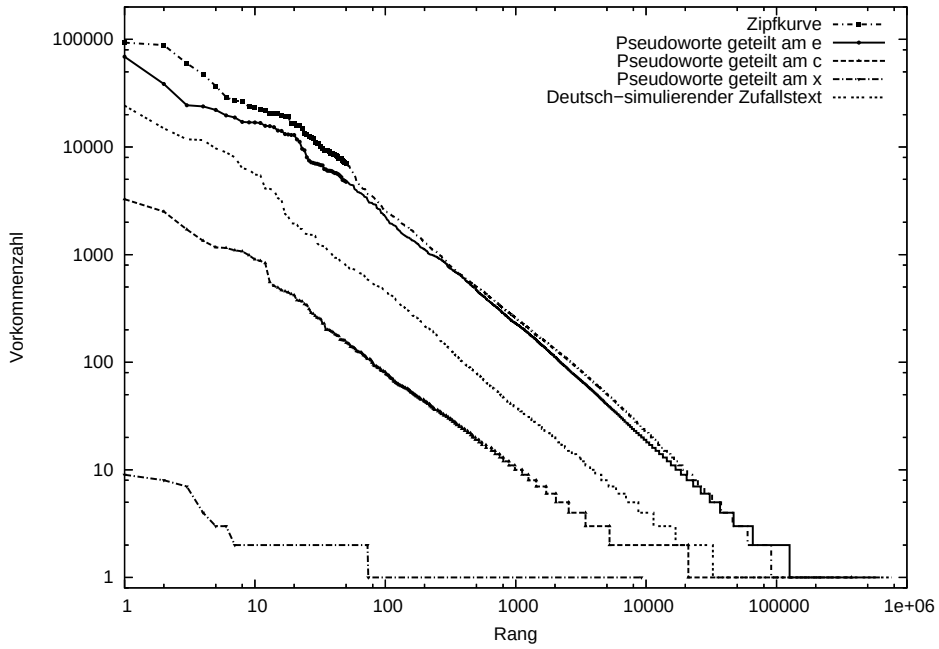


Abbildung 23: Das Zipfsche Gesetzt gilt unabhänig davon, wie Wortgrenzen definiert werden. Trennt man Worte nicht am Leerzeichen, sondern am häufigsten Buchstaben, dem 'e', ergibt sich kaum ein wesentlicher Unterschied. Definiert man das 'c' als Wortgrenze, so verschiebt sich die Kurve bei gleicher Steigung nach. Erst bei sehr seltenen Buchstaben wie dem 'x' bricht die Zipfkurve in sich zusammen. Den Daten liegt der deutsche Korpus zugrunde (Kapitel 2.6.3).

Die Beobachtung aus Kapitel 5.1.2, dass das Zipfsche Gesetz auch für simulierende Zufallstexte (siehe Definition auf Seite 34) gilt, ist ein wenig irritierend: Wir nehmen natürliche Sprache als etwas völlig anderes wahr als eine zufällig erstellte Zeichenkette. Entsprechend überrascht es, dass die Worte eines Textes als bedeutungstragende Einheiten so grundlegende statistische Eigenschaften mit den Pseudoworten simulierender Zufallstexte teilen.

Eine Möglichkeit, diesen Widerspruch aufzulösen ist die Hypothese, dass der zugrundegelegte Wortbegriff derart simplifizierend ist, dass er nicht wirklich bedeutungstragende Einheiten des Textes identifiziert, sondern die Textstruktur soweit zerreißt, dass die entstehende Worthäufigkeitsverteilung der eines Zufallstextes entspricht. In der Tat ist das verwendete Verfahren, Leerzeichen[43] als Wortgrenzen zu definieren, vereinfachend. Beispielsweise werden Strukturen zerteilt, die fest zusammengehören, wenn z.b. aus "New York" die zwei Worte "New" und

[43] Sämtliche Satzzeichen einschließlich des Bindestrichs wurden vor der Untersuchung durch Leerzeichen ersetzt.

"York" werden.[44]

Wenn es wirklich unser übereinfacher Wortbegriff ist, der dafür sorgt, dass sich natürlichsprachige Texte wie Zufallstexte verhalten, sollte sich daran kaum etwas ändern, wenn wir gar nicht mehr versuchen, bedeutungstragende Teile zu identifizieren, sondern den Text nach beliebigen anderen Kriterien unterteilen: Wenn schon ein auf den ersten Blick vernünftiger Wortbegriff das Verhalten eines Zufallstextes reproduziert, sollte dies für einen von vornherein willkürlichen noch eher gelten.

Ich habe für meine Untersuchung den deutschen Korpus[45] herangezogen. Der Text wurde jeweils an verschiedenen Zeichen in Teilstücke unterteilt. Diese Teilstücke sind im folgenden ebenfalls als Pseudoworte bezeichnet. in Abbildung 23 sind die Ergebnisse für das Leerzeichen, für 'e', 'c' und 'x' dargestellt. Zum Vergleich ist das bereits bekannte Ergebnis für einen Deutsch-simulierenden Zufallstext ebenfalls eingetragen. Es wurde die für Auftragungen des Zipfschen Gesetzes übliche Darstellungsweise gewählt: Die Häufigkeit der Worte (bzw. der Teilstücke) ist aufgetragen über ihrem Rang (siehe auch Kapitel 4.1 auf Seite 29).

Ich führe folgende Bezeichnungen ein: *Zipfkurve* sei die Kurve, die die Häufigkeitsverteilung der am Leerzeichen getrennten Worte zeigt. Als e-Kurve, c-Kurve und x-Kurve seien die Kurven bezeichnet, die die Verteilung der an den entsprechenden Buchstaben getrennten Pseudoworte bezeichnen.

Alle Kurven außer der x-Kurve sind sich sehr ähnlich:

- Sie zeigen einen über weite Strecken linearen Verlauf mit gleicher Steigung. Diese liegt nahe bei -1 wie man sich leicht überlegt: die Zipfkurve und die e-Kurve starten bei $x = 1$, $y = 100.000$ und enden bei $x = 100.000$, $y = 1$.
- Sie haben etwa im Rangbereich zwischen 5 und 100 sehr ähnlich ausgeprägte Ausbuchtungen nach oben.

Es gibt zwei Unterschiede zwischen der Zipfkurve auf der einen Seite und der e- und c-Kurve und der simulierenden auf der anderen:

- Die Zipfkurve verläuft für die ersten beiden Ränge fast waagrecht, im Gegensatz zu den anderen. Es ist bekannt, dass das Zipfsche Gesetz in den meisten Sprachen für sehr niedrige Ränge nur eine Näherung darstellt.
- Ab einem Rang von etwa 10.000 ist die Zipfkurve erkennbar steiler als die anderen. Das heisst, dass diese noch mehr extrem seltene Worte bzw Pseudoworte aufweisen als die Zipfkurve.

Worte am Leerzeichen voneinander zu trennen macht intuitiv mehr Sinn, als an irgendeinem anderen Zeichen. Möglicherweise manifestiert sich dieser Unterschied in den beiden sichtbaren Unterschieden zwischen der Zipfkurve und allen anderen.

[44] Auch ist dieses Verfahren auf viele (Schrift)-Sprachen gar nicht anwendbar, die kein Leerzeichen verwenden. In diesen Sprachen werden aber im Normalfall ebenfalls Worte existieren.

[45] Kapitel 2.6.3 auf Seite 15

Bis auf die Zipfkurve und die e-Kurve liegen alle auf sehr unterschiedlicher Höhe. Dies ist verständlich: Je seltener das Zeichen ist, das als Wortgrenze definiert wird[46], desto weniger und längere (Pseudo-)Worte wird es geben. Diese sind dann auch seltener.

Die x-Kurve stellt erwartungsgemäß kein auswertbares Ergebnis mehr da. Das 'x' ist im Deutschen ein so seltener Buchstabe, dass die Pseudoworte, die durch Trennung am x entstehen, fast immer so lange Zeichenketten sind, dass sie einmal oder maximal ein paar mal vorkommen.

[46] 14% der Zeichen sind e's, nur 2% sind c's.

Literatur

[Asher1994] Asher, R. E. und J.M.Simpson (1994): *The Encyclpedia of Language and Linguistics*. Oxford, New York, Seoul, Tokyo: Pergamon Press.

[Bak1997] Bak, Per (1997): *How Nature Works*. New York: Springer.

[Brown1998] *The Brown Corpus*. 1998
http://clwww.essex.ac.uk/w3c/corpus_ling/content/corpora/
Organisation(en): Brown University, Providence, RI

[Bußmann2002] Bußmann, Hadumod (2002): *Lexikon der Sprachwissenschaft*. Stuttgart: Kröner.

[Daniels1996] Daniels, Peter T. und William Bright (1996): *The world's writing systems*. Oxford: Oxford University Press.

[Emille2004] *The Emille Corpus*. 23.09.2004 - 29.10.2004
http://bowland-files.lancs.ac.uk/corplang/emille/
Organisation(en): EMILLE project; Lancaster University, GB; the Central Institute of Indian Languages (CIIL), Mysore, India

[Gusfield1997] Gusfield, Dan (1997): *Algorithms on Strings, Trees, and Sequences*. Cambridge: Cambridge University Press.

[LCMC2004] *The Lancaster Corpus of Mandarin Chinese*. 23.09.2004 - 29.10.2004
http://bowland-files.lancs.ac.uk/corplang/lcmc/
Organisation(en): Lancaster University, GB

[Li1989] Li, Wentian (1989): *Mutual information functions of natural language texts*. Sante Fe Inst. preprint 89-009. [unveröffentlicht]

[Li1992] Li, Wentian (1992): Random texts exhibit Zipf's-law-like word frequency distributions. *IEEE Transactions on Information Theory* 38: 1842 - 1845.

[Mandelbrot1954] Mandelbrot, Benoit (1954): Structure formelle des textes et communication. *Word* 10: 1 - 27.

[Miller1965] Miller, G (1965): Introduction. In: *The Psycho-Biology of Language: An Introduction to Dynamic Philology*. Cambridge, MA: MIT Press

[Proust2001] *Marcel Proust - a la Recherche du Temps Perdu.* 30.07.2001 - ?
http://www.gutenberg.org/dirs/etext01/swann11h.htm
Organisation(en): Project Gutenberg

[Ukkonen1995] Ukkonen, Esko (1995): Online Construction of Suffix Trees. *Algorithmica* 14(3): 249 - 260.

[Zipf1949] Zipf, George Kingsley (1949): *Human Behavior Behavior and The Principle of Least Effort.* New York, London: Hafner Publishing Company.