Tastaturbelegung optimieren

Andreas Wettstein
wettstae@gmail.com
September 2021

1 Vorbereitung: Programm übersetzen

Um den Optimierer benutzen zu können, muss er übersetzt werden. Dabei wird der für Menschen lesbare Programmtext in eine Form überführt, in der sie der Computer direkt ausführen kann. Zum Übersetzen braucht man einen C++-Compiler, zum Beispiel den kostenlosen GNU C++-Compiler (ab Version 4.8.1). Damit geht das Übersetzen so:

```
g++ -std=c++11 -O2 -DNDEBUG opt.cc -o opt
```

Resultat ist das ausführbare Programm `opt`.

Der Optimierer kann für eine bestimmte Zahl von Tasten übersetzt werden, zum Beispiel mit

```
g++ -std=c++11 -O2 -DNDEBUG -DTASTENZAHL=32 opt.cc -o opt
```

für 32 Tasten. Die Voreinstellung ist 35 Tasten, dabei sind zwei Shifttasten mitgezählt. Wenn man die Tastenzahl ändert muss man auch die Konfiguration ändern, siehe Abschnitt 6.

Um die bestmögliche Geschwindigkeit zu erzielen, kann man mit den Optionen des Compilers experimentieren, zum Beispiel

```
g++ -std=c++11 -Wall -Ofast -DNDEBUG -DOHNE2SHIFT \
    -DMIT_THREADS -pthread opt.cc -o opt
```

Hier bewirkt die Option `-DOHNE2SHIFT`, dass Grossbuchstaben, die als zweites in einer Folge zweier Zeichen auftreten, nicht berücksichtigt werden. Da Grossbuchstaben fast nur am Wortanfang (also als erstes Zeichen) vorkommen, hat diese Vereinfachung meist keinen Einfluss auf das Resultat.

Wenn der Optimierer Multithreading (die Verwendung mehrerer Prozessorkerne gleichzeitig) unterstützen soll, muss man beim Übersetzen die Option `-DMIT_THREADS` angeben. Auf vielen Systemen sind weitere Optionen nötig. Zum Beispiel brauche ich auf meinem System die Option `-pthread`.

Normalerweise benutzt der Optimierer die Zeichencodierung UTF-8. Mit der `-DAUSGABE_8BIT` Compileroption kann man die Terminalausgabe auf ISO-8859-1 umstellen.

2 Was man tippen will: Korpus und Häufigkeitsfiles

Um mit Computerhilfe eine gute Tastaturbelegung zu finden, muss man angeben, was man tippen will. Dazu benutzt man ein *Korpus*, eine Textsammlung, die die zu tippenden Texte repräsentiert.

Der Optimierer benutzt nur einen Teil der Information im Korpus: die Zeichenhäufigkeiten, die Häufigkeiten von Zeichenpaaren (*Bigrammen*) und Zeichentripeln (*Trigrammen*). Diese Häufigkeiten sind in Files mit gleichem Namensstamm und den Endungen `.1`, `.2` und `.3` abgelegt. Bigramme und Trigramme überlappen. Zum Beispiel enthält das Wort «Velo» die Bigramme «Ve», «el» und «lo» sowie die Trigramme «Vel» und «elo».

Beim Optimierer sind Häufigkeitsfiles für Deutsch und für Englisch dabei. Bei den Files mit `-t` im Namen wurden nur Bi- und Trigramme mitgezählt, die im Innern von Worten liegen und keine Trennstelle enthalten.

2.1 Eigene Häufigkeitsfiles erstellen

Um eine eigene Textsammlung zu benutzen, geht man so vor:

1. Man packt die Sammlung in ein File, zum Beispiel `meinkorpus.txt`.
2. Man stellt sicher, dass das File UTF-8-codiert ist.
3. Man benutzt `opt`, um die Häufigkeitsfiles zu erzeugen. Zum Beispiel erzeugt

```
./opt meinkorpus.txt
```

   aus der Textsammlung im File `meinkorpus.txt` die drei Häufigkeitsfiles `meinkorpus.txt.1`, `meinkorpus.txt.2` und `meinkorpus.txt.3` sowie eine Wortliste in `meinkorpus.txt.wl`.

Wenn man `opt` mit zwei Argumenten aufruft wird angenommen, dass das erste Argument ein File mit UTF-8-codierten TEX-Trennmustern (`*.pat.txt`) ist, das zweite die Textsammlung. Es werden Häufigkeitsfiles erzeugt, bei denen nur Bigramme und Trigramme mitgezählt sind, die im Innern von Worten liegen und keine Trennstelle enthalten, und statt der Wortliste wird eine Silbenliste gebildet. Zum Beispiel:

```
./opt hyph-de-1996.pat.txt meinkorpus.txt
```

`opt` kann auch Häufigkeitsfiles zusammenzählen. Dazu ruft man es mit mehr als zwei Argumenten auf. Zum Beispiel erzeugt

```
./opt deutsch.txt deutsch-t.txt deutsch-t.txt gemischt.txt
```

aus den mitgelieferten Häufigkeitsfiles für Deutsch neue Files, bei denen Bi- und Trigramme an Wortgrenzen und mit Trennstellen nur zu einem Drittel gezählt werden. Eine andere Möglichkeit, Häufigkeitsfiles zu kombinieren und zu gewichten, ist die Option `-G`, siehe unten.

2.2 Häufigkeitsfiles benutzen

Beim Start des Optimierers gibt man den Namensstamm der Häufigkeitsfiles an. Dazu benutzt man die Optionen `-2` oder `-3`. Mit `-2` werden nur die Zeichen- und Bigrammhäufigkeiten benutzt, mit `-3` auch die Trigrammhäufigkeiten. Zum Beispiel startet

```
./opt -2 deutsch.txt
```

eine Optimierung mit den Häufigkeitsfiles für Deutsch, bei der Trigramme ignoriert werden.

Falls das Häufigkeitsfile mit den Zeichenhäufigkeiten (im Beispiel oben `deutsch.txt.1`) nicht existiert, versucht der Optimierer, ein File mit dem angegebenen Namen (also `deutsch.txt`) zu öffnen. Gelingt das, wird das File als Korpus behandelt. Man kann so den Zwischenschritt über Häufigkeitsfiles überspringen, jedoch dauert das Einlesen des Korpus länger.

Man kann `-2` oder `-3` mehrfach benutzen. Die so angegeben Häufigkeitsfiles werden gleich gewichtet, unabhängig von der Grösse der tabellierten Korpora. Mit der Option `-G` (Voreinstellung 1) kann man die Gewichtung nachfolgend angegebener Häufigkeitsfiles ändern, siehe Gleichung (1). Zum Beispiel führt

```
./opt -2 deutsch.txt -G 3 -2 englisch.txt
```

eine Optimierung durch, in der Englisch dreimal so stark gewichtet wird wie Deutsch.

Die separate Angabe verschiedener Häufigkeitsfiles durch mehrfaches `-2` oder `-3` ist etwas rechenaufwändiger, als ein einziges Summenkorpus zu verwenden. Dem steht der Vorteil gegenüber, dass das Gesamt-Optimum (wenn man es denn findet) Pareto-optimal bezüglich der einzelnen Korpora ist: Man kann die Belegung für kein Korpus verbessern, ohne sie für einen

andern zu verschlechtern. Wenn man ein Summenkorpus verwendet, ist das nicht garantiert (siehe Abschnitt 7.3).

2.3 Korpusgrösse, systematischer und statistischer Fehler

Wenn man statt Häufigkeitsfiles direkt ein volles Korpus einliest, wird bei der Bewertung von Belegungen aus einem File (siehe Option `-r` in Abschnitt 5) die Standardabweichung des Aufwands und die Standardabweichung der Differenz zum Aufwand der ersten Belegung im File geschätzt und ausgegeben. Dafür wird angenommen, dass die Häufigkeiten von verschiedenen Wörtern in einem Text unkorreliert sind. Ein «Wort» ist hier eine Zeichenfolge, die durch ein Leerzeichen oder ein Zeichen, das in der Belegung nicht vorkommt (siehe Abschnitt 6.1) begrenzt wird. Für die Häufigkeiten von Wörtern wird eine Binomialverteilung angenommen, deren Mittelwert und Varianz aus den relativen Häufigkeiten im Korpus geschätzt werden. Bei der Berechnung der Standardabweichung werden Trigramme und fehlende Zeichen nicht berücksichtigt.

Aufwandsdifferenzen von einer Standardabweichung oder weniger sind unerheblich, das heisst, sie könnten ohne weiteres allein durch die zufällige Textauswahl zustande kommen. Unterschiede von drei Standardabweichungen oder mehr hingegen darf man als real ansehen. Um die Standardabweichung zu reduzieren, muss man das Korpus vergrössern. Die Standardabweichung sinkt mit der Quadratwurzel der Korpusgrösse. Um zum Beispiel die Standardabweichung zu halbieren, muss man ein viermal grösseres Korpus benutzen.

Der statistische Fehler ist nicht die einzige Unsicherheit, die vom Korpus herkommt. Zum Beispiel hat die Art der Texte (Texte mit langen oder kurzen Sätzen, mit vielen oder wenigen Fremdwörtern) einen systematischen Einfluss, der sich in der Standardabweichung nicht zeigt. Nach meiner Erfahrung ist ein Korpus von wenigen Megabyte ausreichend gross.

2.4 Zeichen durch andere ersetzen

Will man ein einzelnes Zeichen durch Tippen einer Folge von einem oder mehreren anderen Zeichen eingeben, kann man das in der Konfiguration mit `Ersatz` erreichen, siehe Abschnitt 6.1. Es ist nicht notwendig, das Korpus oder die Häufigkeitsfiles zu verändern.

Will man hingegen eine Folge von mehreren Zeichen durch eine andere Folge von Zeichen eingeben, funktioniert dieses einfache Verfahren nicht, da in den Häufigkeitsfiles nicht genug Information steckt, um allgemeine Ersetzungen zu machen. In diesem Fall muss man im Korpus die zu erzeugende Zeichenfolge durch die zu tippende ersetzen. Aus diesem veränderten Korpus erzeugt man neue Häufigkeitsfiles.

2.5 Weitere Operationen

Mit der Option `-T` kann man aus einem Korpus und einem File mit Trennmustern ein neues Korpus erzeugen, in dem alle möglichen Trennstellen durch weiche Trennzeichen (U+00AD) markiert sind. Zum Beispiel:

```
./opt -T hyph-de-1996.pat.txt eingabe.txt ergebnis.txt
```

3 Das Bewertungsschema

Um eine Belegung zu bewerten, betrachtet man die Tastenanschläge, die nötig sind, um das Korpus damit einzugeben. Wie beim Korpus beschränkt man sich bei der Bewertung auf einige Kriterien.

Bigramm	Tasten	Tastenbi- und trigramme
xy	$t_x t_y$	$t_x t_y$
Xy	$s_x t_x t_y$	$s_x t_x t_y$, $t_x t_y$, $s_x t_x$
xY	$t_x s_y t_y$	$t_x s_y t_y$, $s_y t_y$, $t_x s_y$
XY	$s_x t_x s_y t_y$	$s_x t_x s_y$, $t_x s_y t_y$, $s_x t_x$, $t_x s_y$, $s_y t_y$

Tabelle 1: Zerlegung von Bigrammen in Tastenbi- und trigramme. t_X ist die Taste für x, s_X die zugehörige Shifttaste, t_Y und s_Y analog. $s_X t_X$ und $s_Y t_Y$ zählen als Einzeltasten, da sie Grossbuchstaben entsprechen. $t_X t_Y$ und $t_X s_Y$ und sind Tastenbigramme, $s_X t_X t_Y$ und $s_X t_X s_Y$ sind Shift-Bigramme, $t_X s_Y t_Y$ ein Tastentrigamm.

Jedes dieser Kriterien steuert einen Beitrag zum *Aufwand* bei. Die Summe dieser Beiträge ist der Gesamtaufwand, siehe Gleichung (2).

3.1 Mechanische Kriterien

Einzeltastenaufwände. Jede Taste hat einen Aufwand. Dieser wird mit der Häufigkeit multipliziert, mit der sie angeschlagen wird, also mit der Häufigkeit des Zeichens, mit dem die Taste belegt ist. Die Summe dieser Produkte ist der *Lageaufwand*, siehe Gleichung (3).

Bigrammaufwände. Jede Folge zweier Tasten hat einen Aufwand, den Bigrammaufwand. Wir nennen hier die Folge zweier Tastenanschläge *Bigramm*, genauso wie ein Zeichenpaar. Der Klarheit wegen werden wir zum Teil *Tastenbigramm* schreiben. Der Unterschied wird in Tabelle 1 durch Zerlegung des Bigramms «xy» mit allen Kombinationen von Gross- und Kleinschreibung gezeigt. Dabei treten Tastenbigramme der Form Shift+Zeichentaste auf, deren Häufigkeit aus Zeichenhäufigkeit bestimmt wird und deren Aufwände zum Lageaufwand zählen. Ferner treten normale Tastenbigramme aus zwei Zeichentasten, Shift-Bigramme und Trigramme auf, so dass der Beitrag durch Bigramme recht komplex wird, siehe Gleichung (6).

Shift-Bigramme bestehen aus einer Shifttaste, der zugehörigen Zeichentaste und der darauf folgenden Taste. Zu ihrer Bewertung wird der Bigrammaufwand für die Shifttaste und die letzte Taste mit einem Faktor multipliziert, siehe Gleichung (6).

Trigrammaufwände. Einer Folge von drei Tastendrücken wird ein Aufwand zugewiesen, der Trigrammaufwand. Nur wenn man Option `-3` verwendet werden alle Tastentrigramme bewertet, ansonsten nur solche, deren Häufigkeit bereits durch Bigrammhäufigkeiten festgelegt ist, siehe Tabelle 2 und Gleichung (7).

Fingerbelastung. Für jeden Finger wird eine *Zielhäufigkeit* festgelegt, also der Anteil an den Anschlägen, die er tippen soll. Wenn die tatsächliche Häufigkeit die Zielhäufigkeit überschreitet, wird der Überschuss quadriert und mit einem Gewicht multipliziert, um einen Aufwand zu erhalten, siehe Gleichung (8). Ausgenommen sind *fixe Finger*: Finger, deren Anschläge nicht von der Belegung abhängen, sondern durch Konfiguration und Korpus festgelegt sind.

Trigramm	Tasten	Tastentrigramme
xyz	$t_x t_y t_z$	$t_x t_y t_z$
Xyz	$s_x t_x t_y t_z$	$s_x t_x t_y$, $t_x t_y t_z$
xYz	$t_x s_y t_y t_z$	$t_x s_y t_y$, $s_y t_y t_z$
XYz	$s_x t_x s_y t_y t_z$	$s_x t_x s_y$, $t_x s_y t_y$, $s_y t_y t_z$
xyZ	$t_x t_y s_z t_z$	$t_x t_y s_z$, $t_y s_z t_z$
XyZ	$s_x t_x t_y s_z t_z$	$s_x t_x t_y$, $t_x t_y s_z$, $t_y s_z t_z$
xYZ	$t_x s_y t_y s_z t_z$	$t_x s_y t_y$, $s_y t_y s_z$, $t_y s_z t_z$
XYZ	$s_x t_x s_y t_y s_z t_z$	$s_x t_x s_y$, $t_x s_y t_y$, $s_y t_y s_z$, $t_y s_z t_z$

Tabelle 2: Zerlegung von Trigrammen in Tastentrigramme. t_X ist die Taste für x, s_X die zugehörige Shifttaste, t_Y, s_Y, t_Z und s_Z analog. Nur für die Häufigkeiten von $t_X t_Y t_Z$ und $t_X t_Y s_Z$ brauchen wir Trigrammenhäufigkeiten. Für die anderen Tastentrigramme und Shift-Bigramme genügen Bigrammhäufigkeiten.

3.2 Nichtmechanische Kriterien

Die bisher beschriebene Bewertung beschäftigt sich mit der Mechanik des Tippens. Möglicherweise hängt die Häufigkeit, mit der man ähnliche Zeichen verwechselt, davon ab, wie die entsprechenden Tasten zueinander liegen. Im Konfigurationsfile kann man den Grad der Ähnlichkeit von Zeichen als Zahl angeben. Diese wird mit dem *Verwechslungspotenzial* multipliziert, um einen Aufwand zu bekommen, siehe Gleichung (9). Das Verwechslungspotenzial hängt von den Tasten ab, denen die Zeichen zugeordnet sind und lässt sich im Konfigurationsfile einstellen. In der Voreinstellung sind alle Zeichen unähnlich.

Manche Anwender wollen bestimmte Zeichen in einem bestimmten Teil der Belegung haben. Diese *Vorlieben* für die Zuordnung von Zeichen zu Tasten kann man im Konfigurationsfile mit einer Zahl festlegen, die das Ausmass der Vorliebe angibt. Jede erfüllte Vorliebe reduziert den Aufwand um diese Zahl, siehe Gleichung (9).

Auch durch Manipulation von Korpus und Häufigkeitsfiles kann man die Bewertung zu beeinflussen. Zum Beispiel wurde in einem Beitrag auf der Neo-Mailingliste behauptet, dass guter Schreibfluss innerhalb von Silben wichtiger ist als über Silbengrenzen hinweg. Verwendet man zur Optimierung Häufigkeitsfiles, bei denen nur Bi- und Trigramme mitgezählt sind die keinen Trennstelle enthalten (siehe Abschnitt 2), kann man dem näherungsweise Rechnung tragen (näherungsweise, weil Silbengrenzen und Trennstellen nicht ganz dasselbe sind).

Zum Vergleich von Belegungen mit unterschiedlicher Zeichenausstattung kann man einen Aufwand für die Zeichen veranschlagen, die sich mit einer Belegung nicht eingeben lassen.

3.3 Bewertungskriterien ausgeben

Mit der Option `-A` werden alle von Null verschiedenen Bigrammaufwände, Verwechslungspotenziale und Vorlieben ausgegeben. Mit `-3` werden auch die Trigrammaufwände ausgegeben.

4 Der Ablauf einer Optimierung

Der Optimierer benutzt ein einfaches Verfahren, das von einer zufälligen Belegung ausgeht. Diese wird durch wiederholtes Vertauschen von Zeichen schrittweise verbessert. Falls keine Vertauschung mehr eine Verbesserung bringt, ist man am Ende. Das Resultat ist eine *lokal optimale* Belegung.

Eine lokal optimale Belegung kann weit davon entfernt sein, optimal im Sinne der Bewertungskriterien zu sein. Daher wird das obige Verfahren vielfach (mit immer neuen Ausgangsbelegungen) durchgeführt. Die Zahl der Wiederholungen kann mit der Option `-i` angegeben werden. Ohne diese Angabe läuft der Optimierer bis er abgebrochen wird. Wie viele Wiederholungen man durchführen muss, hängt von den Bewertungskriterien, vielleicht auch vom Korpus ab. Nach meiner Erfahrung ist 10000 für die vorgegebenen Kriterien ein vernünftiger Wert, wenn man die Option `-2` verwendet. Mit Option `-3` braucht man eher mehr Wiederholungen.

Normalerweise wird jede lokal optimale Belegung ausgegeben, die besser als alle bisher gefundenen ist. Wenn man mit der Option `-m` eine Schwelle angibt, wird jede lokal optimale Belegung angezeigt, deren Aufwand unterhalb dieser Schwelle liegt.

Mit der Option `-s` kann man eine positive, ganze Zahl für den Saatwert des Zufallsgenerators angeben. Das kann nützlich sein, um reproduzierbare Optimierungsläufe zu erhalten. Ohne `-s` wird der Saatwert zufällig gewählt.

Die Zahl der Threads, die verwendet werden sollen, kann man mit `-t` angeben, sonst wird ein Thread verwendet. Die mit `-i` angegebene Zahl der Wiederholungen versteht sich pro Thread. Wenn man mehr Threads verwendet kann man diese Zahl also entsprechend senken.

5 Die Ausgabe des Optimierers

Zunächst einige Begriffe: *Handwechsel* sind Bigramme, deren Tasten von verschiedenen Händen angeschlagen werden. *Doppeltanschläge* sind Bigramme, bei eine Taste zweimal angeschlagen wird. *Kollisionen* sind Bigramme, bei denen verschiedene Tasten vom selben Finger angeschlagen werden. *Nachbaranschläge* sind Bigramme, deren Tasten von benachbarten Fingern derselben Hand angeschlagen werden. *Einwärtsbewegungen* sind Bigramme, bei denen die erste Taste von einem Finger weiter aussen an derselben Hand als die zweite angeschlagen wird (Der äusserste Finger ist der kleine Finger, der innerste der Zeigefinger). *Auswärtsbewegungen* verlaufen umgekehrt. Der Vorsatz «Shift-» kennzeichnet besondere Bigramme, deren erste Taste Shift und deren zweite Taste eine Zeichentaste ist.

5.1 Textausgabe

Hier die Ausgabe für «Aus der Neo-Welt», bewertet mit einem deutsch-englisch gemischten Korpus:

```
 Aus der Neo-Welt 382.859 Gesamtaufwand  187.075 Lageaufwand        links rechts
                    1.029 Kollisionen      6.976 Shift-Kollisionen  ob  5.7 11.8
   kuü.ä vgcljf    71.404 Handwechsel     24.118 Shift-Handwechsel  mi 36.4 32.1
   hieao dtrnsß     1.796 Ein-/Auswärts   25.117 Ein- oder auswärts un  5.2  8.9
   xyö,q bpwmz      9.262 benachbart      22.116 Shift-benachbart  sum 47.2 52.8
                   8.4 11.2 14.0 13.7 --.- --.- 17.6 10.8 14.3 10.1 Sh  2.9  1.2
```

Links oben steht der Name der Belegung oder eine Laufnummer. Daneben stehen Gesamt- und Lageaufwand. Links ist die Belegung gezeigt, daneben Bigrammhäufigkeiten: der Prozentsatz aller normalen Tastenbigramme (ohne solche, die ein Leerzeichen enthalten), die Kollisionen, Handwechsel oder Nachbaranschläge sind, sowie das Verhältnis von Ein- zu Auswärtsbewegungen. In der Spalte rechts daneben steht der Prozentsatz von Shift-Bigrammen, die Kollisionen, Handwechsel oder Nachbaranschläge sind. Rechts stehen die Anteile, mit denen sich die Anschläge auf die drei Tastenzeilen und die Hände verteilen. Die unterste Zeile zeigt die Verteilung der Anschläge auf die Finger (von linkem zu rechtem Kleinfinger) und die Anschlagshäufigkeiten der Shifttasten. Das `--.-` für die Daumen kommt daher, dass die Leertaste einem Daumen zugeordnet wurde, aber keinem bestimmten. Daher lassen sich für die einzelnen Daumen keine Häufigkeiten angeben. Anschläge die auf eine unbestimmte Daumentaste fallen werden zwar im Gesamt- und Lageaufwand berücksichtigt, aber für die restliche Ausgabe nicht mitgezählt.

Verwendet man Option `-3`, bekommt man zwei zusätzliche Zeilen:

```
    4.765 kein Handwechs. 44.851 zwei Handwechsel
    3.582 Wippe            6.403 IndirKollision
```

Die erste gibt die Häufigkeiten von Trigrammen ohne und mit zwei Handwechseln an, die zweite die Häufigkeit von Trigrammen, die aus einer Ein- und einer Auswärtsbewegung bestehen (Reihenfolge egal), und die von Trigrammen mit zwei Handwechseln bei denen die erste und letzte Taste verschieden sind, aber vom selben Finger getippt werden (*indirekte Kollisionen*).

Mit der Option `-b`, gefolgt von einer Zahl zwischen 0 und 100, erhält man eine ausführliche Beschreibung der häufigsten Bigramme ohne Handwechsel. Zudem wird die Häufigkeit von Kollisionen und Nachbaranschlägen pro Finger beziehungsweise Fingerpaar aufgeschlüsselt. Verwendet man `-b` ein zweites Mal (gefolgt von einer Zahl) bekommt man zusätzlich Shift-Bigramme, verwendet man sie ein drittes Mal auch noch Trigramme.

Option `-k` verkürzt die Ausgabe auf eine Zeile pro Belegung. Mit der Option `-m` kann man einen Gesamtaufwand angeben, unterhalb dem alle lokale optimalen Belegung ausgegeben werden. Sonst

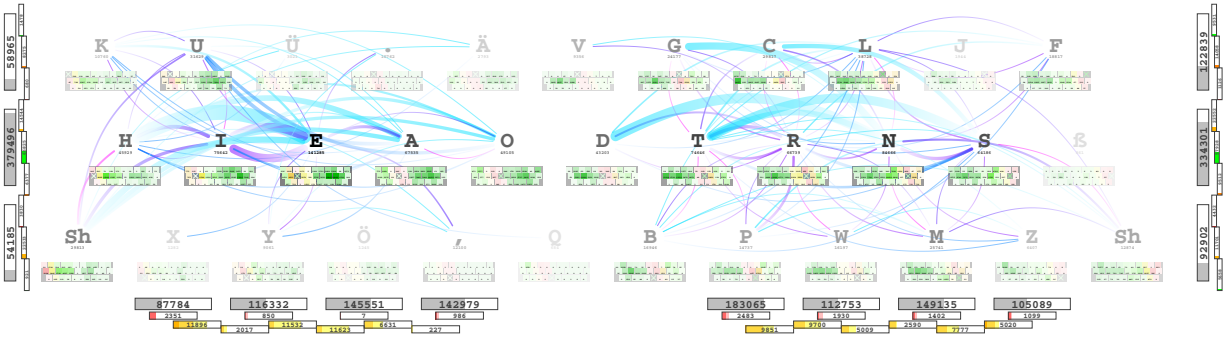


Abbildung 1: Beispiel für eine Belegungsgrafik

wird immer nur die aktuell beste Belegung angezeigt.

Mit der Option `-w` kann man ein File mit einer Wortliste angeben. Das File muss UTF-8-codiert sein. In jeder Zeile muss eine Worthäufigkeit und, durch Leerzeichen getrennt, das entsprechende Wort stehen. Für eine gegebene Belegung werden alle Wörter in Abschnitte unterteilt, die mit derselben Hand getippt werden. Die Häufigkeiten dieser *Handeinsätze* werden gesammelt, aufsummiert und nach Häufigkeit sortiert ausgegeben. Normalerweise werden alle Handeinsätze aufgeführt, deren Häufigkeit insgesamt 95 % aller Handeinsätze ausmacht. Diese Grenze kann man mit der Option `-H` ändern.

Mit der Option `-r` kann man ein File angeben, in dem Belegungen stehen, die bewertet ausgegeben werden sollen, ohne dass eine Optimierung durchgeführt wird. Zusätzlich kann man mit Option `-V` diese Belegungen variieren. Von jeder Belegung werden alle Variationen erzeugt, die für bis zu der mit `-V` angegeben Zahl von Tasten von der gegebenen abweichen. Ist im File ein Zeichen einer Belegung als Grossbuchstabe angegeben, wird seine Position nicht variiert. Man sollte die Ausgabe mit `-m` auf die besseren Variationen einschränken, da sonst die Zahl der Belegungen schnell unhandhabbar gross wird.

5.2 Grafische Ausgabe

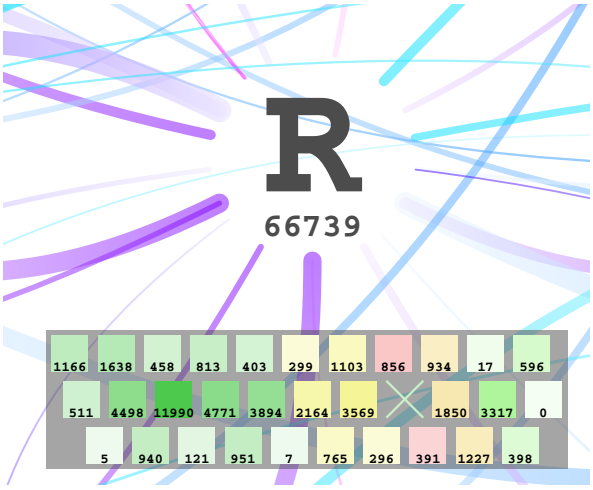
Mit der Option `-g` kann man den Namen eines PostScript-Files angeben, in das Grafiken für alle Belegungen geschrieben werden, die auch als Text ausgegeben werden. Zum Beispiel erzeugt

```
./opt -2 deutsch.txt -r bsptast.txt -g bsptast.ps
```

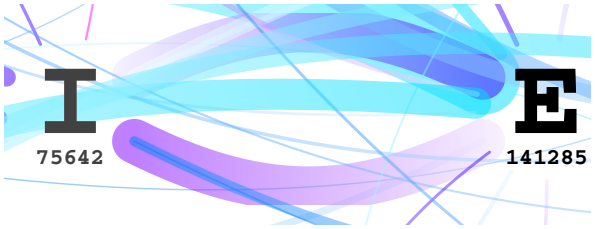
Grafiken in `bsptast.ps`, die die Belegungen in `bsptast.txt` für deutsche Texte auswerten. Abbildung 1 zeigt ein Beispiel.[1]

Die Grafiken haben zwei Ebenen. Der Vordergrund zeigt die Zeichen auf den Tasten, die Schwärze entspricht ihrer Häufigkeit. Diese steht als Zahlenwert unter den Zeichen (10000 entspricht 1 %). Darunter zeigt eine Minitastatur die Verteilung der folgenden Anschläge. Die Farbe ist je nach Bigrammtyp verschieden, die Intensität und Zahlen auf den Minitasten geben die Häufigkeit der Bigramme an. Abbildung 2a zeigt ein Beispiel. Man erkennt, dass auf «r» und «R» etwa 6,7 % der Anschläge entfallen und dass nach einem «r» oder «R» am häufigsten die Taste in der Grundposition des linken Mittelfingers angeschlagen wird. Sie ist im Beispiel mit «e» belegt, und etwa 1,2 % aller Bigramme sind r-e.

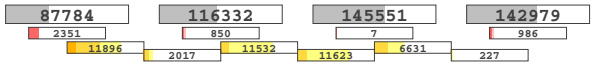
[1] Für die Abbildungen wurde das PostScript-File in PDF umgewandelt. Dadurch steigt die Filegrösse, dafür beherrscht PDF im Gegensatz zu PostScript Transparenz. Für die Umwandlung mit Ghostscript verwenden Sie die Option `-dALLOWPSTRANSPARENCY`.



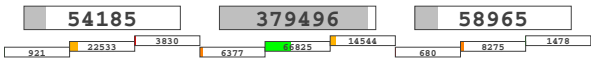
(a) Vordergrund: Einzelzeichen und folgende Anschläge



(b) Hintergrund: Bögen für Bigramme



(c) Unterer Rand: Fingerbelastung



(d) Seitenrand (gedreht): Gruppen und Zeilensprünge

Abbildung 2: Details der Belegungsgrafik

Der Hintergrund zeigt Tastenbigramme durch Bögen. Die Bogendicke entspricht der Häufigkeit, die Farbe ist je nach Bigrammtyp verschieden. Die Reihenfolge, in der die Tasten eines Bigramms angeschlagen werden, wird durch den Intensitätsverlauf und die Krümmung angezeigt. Für die rechte Hand läuft die Bewegung gegen den Uhrzeigersinn, für die linke im Uhrzeigersinn. Seltene Bigramme und Handwechsel sind der Übersichtlichkeit wegen weggelassen. Abbildung 2b zeigt dicke violette Bögen, die «I» und «E» verbinden. Im Beispiel werden «I» und «E» mit links getippt, also ist der obere Bogen für i-e, der untere für e-i.

Die grauen Balken unter der Tastatur zeigen die Anschlagshäufigkeit pro Finger. Der Kasten um den Balken entspricht 25 % der Anschläge, die Zahl darin ist die relative Häufigkeit bezogen auf die Zeichen. Der Kasten rechts in Abbildung 2c zeigt, dass der linke Zeigefinger etwa 14,3 % der Zeichen tippt. Dieser Wert ist etwas höher als in der Textausgabe, da dort der Grundwert die Zahl aller Anschläge, nicht, wie hier, die Zahl aller Zeichen ist.

Ganz unten sind Kästchen mit summierten Bigrammhäufigkeiten. Der Finger, für den ein Kästchen gezeigt wird, tippt das erste Zeichen des Bigramms. Kollisionen sind rot, die Intensitäten stehen für drei Bereiche der Sprungdistanz. Die Nachbaranschläge, in Ein- und Auswärtsbewegungen aufgeteilt, sind gelb-orange, die Farbintensität steigt mit den übersprungenen Zeilen (0, 1 oder 2). Die Kästchengrösse entspricht 2 % der Gesamtbigrammzahl, die Zahl im Kästchen ist die relative Häufigkeit bezogen auf die Gesamtzeichenzahl. In Abbildung 2c links unten sieht man, dass links der kleine Finger die meisten Kollisionen bewältigen muss, nämlich etwa 0,24 auf 100 Zeichen.

Jede Tastenzeile jeder Hand bildet eine *Gruppe*. Die grauen Balken links und rechts neben der Tastatur zeigen die Anschlagshäufigkeit pro Gruppe. Der Kasten um den Balken entspricht 40 % der Anschläge, die Zahlen darin sind auf die Zahl der Zeichen bezogen. Wie im mittleren grauen Kasten in Abbildung 2d zu sehen ist, entfallen etwa 37,9 % aller Zeichen auf die mittlere Zeile der linken Hand.

Die Kästchen seitlich zeigen die Bigrammhäufigkeiten ohne Handwechsel, aufgeteilt nach der Gruppe, in die der erste Anschlag fällt. Für jede Gruppe wird weiter aufgeteilt nach der Zeile, in die der zweite Anschlag fällt. Die Kästchengrösse entspricht 20 % der Gesamtbigrammzahl, die Zahl im Kästchen ist die relative Häufigkeit bezogen auf die Gesamtzeichenzahl. Die Zahl im grünen Kästchen unten in der Mitte von Abbildung 2d besagt also, dass auf 100 Zeichen etwa 6,7

Bigramme kommen, deren beide Anschläge in der mittleren Zeile der linken Tastaturhälfte liegen.

Wenn während einer Optimierung Grafiken erzeugt werden, sollte man mit Option `-i` die Zahl der Durchläufe festlegen. Bricht man die Optimierung manuell (mit Ctrl+C) ab, kann die Grafik unvollständig sein. Das PostScriptfile, das der Optimierer erzeugt, ist für Menschen les- und veränderbar. Am Anfang gibt es einige Schalter, die die Anzeige beeinflussen. Am Ende befindet sich die Liste der Belegungen, die dargestellt werden sollen.

5.3 Tippvorgang im Text markieren

Mit Option `-M` kann man den Namen eines UTF-8-codierten Textfiles angeben, für das für jede der Belegungen im mit Option `-r` angegebenen Belegungsfile gezeigt wird, wie die Eingabe erfolgt: welche Hand verwendet wird, ob das Zeichen in der Grundposition liegt, ob es ein Nachbaranschlag oder eine Kollision mit dem vorherigen Zeichen ist. Das Ergebnis steht in einem HTML-File, dessen Namen durch Anhängen von `.html` aus dem Namen des Textfiles entsteht. Am Anfang dieses Files befinden sich Definitionen, mit dem man den Stil der Darstellung einstellen kann.

6 Die Konfiguration ändern

Normalerweise holt der Optimierer seine Einstellungen aus `standard.cfg`, einem *Konfigurationsfile*, das beim Optimierer dabei ist. Die Option `-K` erlaubt es, ein anderes Konfigurationsfile anzugeben. Die Option darf mehrfach verwendet werden, was denselben Effekt hat, als würde man die so angegebenen Konfigurationsfiles aneinanderhängen.

Konfigurationsfiles sind UTF-8-codierte Textfiles mit zeilenorientierter Struktur. Leerzeilen werden ignoriert. Am Anfang jeder Zeile dürfen beliebig viele Leerzeichen und Tabulatoren stehen, die ignoriert werden. Ist das erste darauf folgende Zeichen ein #, wird der Rest der Zeile ignoriert. Andernfalls kommt ein Schlüsselwort, gefolgt von einer vom Schlüsselwort abhängigen Zahl von Argumenten, getrennt durch Leerzeichen oder Tabulatoren. Hinter den Argumenten darf ein mit # eingeleiteter Kommentar stehen. Argumente können *Namen* (Zeichenfolgen, die weder Leerzeichen noch Tabulatoren enthalten), *Strings* (in Anführungszeichen gesetzte Zeichenfolgen, wobei die Anführungszeichen am Anfang und Ende übereinstimmen müssen, sonst aber beliebig sind), *Flags* (+ oder −) oder Zahlen sein. Zahlen dürfen Dezimalkomma oder Dezimalpunkt enthalten.

Im Folgenden erscheinen einige mathematische Symbole in Klammern, auf die in Abschnitt 7 Bezug genommen wird.

6.1 Den Zeichenumfang festlegen

In der Regel wird man bei der Optimierung nur die wichtigsten Zeichen für die Sprachen, für die man eine Belegung sucht, berücksichtigen. Zeichen werden meist als Paare angegeben. Das erste Zeichen liegt auf der Grundebene, das zweite auf der geshifteten Ebene derselben Taste. Gibt man mehr als zwei Zeichen an, werden die zusätzlichen als Aliase des zweiten behandelt. Man kann auch ein einzelnes Zeichen angeben, das allein auf einer Taste liegt. Zum Beispiel bedeutet

```
Zeichen 'aA'
Zeichen ' '
```

dass das kleine und grosse «A» zusammen auf eine Taste kommen und das Leerzeichen eine eigene Taste bekommt. Man kann Zeichen auch einer bestimmten Taste zuordnen. Zum Beispiel wird mit

```
FixesZeichen AD11 'ß'
```

das «ß» auf die Taste mit dem Namen AD11 gelegt (näheres zur Festlegung von Tasten und ihren Namen siehe Abschnitt 6.2). Die Position des «ß» wird dann nicht mitoptimiert, sein Vorhandensein in der Belegung auf AD11 kann aber beeinflussen, welchen Tasten andere Zeichen zugeordnet werden.

Die Zahl von Zeichen und FixesZeichen im Konfigurationsfile muss der Zahl der Tasten weniger zwei (Shifttasten) sein, siehe –DTASTENZAHL in Abschnitt 1. Dasselbe Zeichen darf nicht in verschiedenen Zeichen- oder FixesZeichen-Festlegungen vorkommen.

Mit Zeichen und FixesZeichen kann man optional einen PostScript-Glyphnamen angeben, der das Symbol benennt, mit dem diese Zeichen in Grafiken dargestellt wird. Zum Beispiel legt

```
FixesZeichen SPCE ' ' underscore
```

fest, dass Leerzeichen in den Grafiken durch Unterstriche dargestellt werden. Die Angabe von Glyphnamen kann für Zeichen ausserhalb von ISO-8859-1 nötig sein. Zu deren Anzeige muss zudem die verwendete Schriftart diese Zeichen unterstützen. Die Schriftart lässt sich durch Angabe von PostScript-Fontnamen mit Zeichenfont und Beschreibungsfont für die Zeichen der Belegung und die Beschreibung (die Zahlen) getrennt angeben, zum Beispiel:

```
Zeichenfont       FiraMono-Medium
Beschreibungsfont FiraSans-Book
```

Manchmal will man die Grundebene zu einem Zeichen in der Shift-Ebene freilassen. Das erreicht man, indem man ein Platzhalterzeichen festlegt und damit die Grundebene belegt. Zum Beispiel kommt mit

```
Platzhalter '¦'
Zeichen '¦ß'
```

das «ß» auf die geshiftete Ebene einer Taste, deren Grundebene leer ist. Man kann nur einen Platzhalter festlegen, vor allen Zeichen. In der Auswertung wird der Platzhalter behandelt als käme er im Korpus nicht vor.

Man kann auch einzelne Zeichen durch eine Folge anderer ersetzen. Wenn man zum Beispiel statt «ß» lieber «ss» tippt, kann man das im Konfigurationsfile mit

```
Ersatz 'ßss'
```

ausdrücken. Das erste Zeichen in einem Ersatz-String ist das zu ersetzende Zeichen, der Rest der Ersatz. Ersetzt wird jedoch nur, wenn das zu ersetzende Zeichen nicht unter den mit Zeichen und FixesZeichen angegeben ist. Der Ersatz darf auch Zeichen enthalten, für die mit einem vorher angegebenen Ersatz-String ein Ersatz festgelegt wurde. Besonders interessant ist Ersatz, um tote-Taste-Folgen bei der Optimierung zu berücksichtigen.[2]

6.2 Das Tastaturlayout festlegen

Die Zahl der im Konfigurationsfile festgelegten Tasten muss der beim Übersetzen bestimmten entsprechen, siehe –DTASTENZAHL in Abschnitt 1. Ein Beispiel für die Festlegung einer Taste ist

```
Taste AD03   4 1   3.25 1   –3   –   4   –
```

Hier ist AD03 der Name der Taste.[3] Tastennamen dürfen nur einmal vergeben werden.

[2] totetasten.cfg zeigt ein Beispiel.

[3] Die Tastennamen in den mitgelieferten Konfigurationsfiles sind von xkeyboard-config übernommen und lehnen sich an ISO 9995-1 an. Wer will kann sie durch eingängigere Namen ersetzen, zum Beispiel AD01 durch Q.

Auf den Tastennamen folgen die Spalte und die Zeile (ξ und ρ) der Taste, im Beispiel `4` und `1`. Beides sind ganze Zahlen, Zeilen zwischen 0 und 4 (wobei 0 oben ist) und Spalten zwischen 0 und 15 (0 ist links). Dann kommen horizontale und vertikale Koordinaten (x und y) der Taste, im Beispiel `3.25` und `1`. Das sind beliebige Zahlen, die in der Regel jedoch nahe bei den Werten für Spalte und Zeile liegen.

Danach kommt eine Zahl, die den Finger (f) bezeichnet, der die Taste anschlägt. Negative Zahlen sind links, positive rechts. `-5` und `5` sind die kleinen Finger, `-4` und `4` die Ringfinger, `-3` und `3` die Mittelfinger, `-2` und `2` die Zeigefinger und `-1` und `1` die Daumen. `0` ist ein Daumen, wobei nicht festgelegt ist, welcher.

Dann kommt ein Flag. Ist es `+`, ist die Taste die Grundposition des entsprechenden Fingers. Die folgende Zahl gibt den Einzeltastenaufwand (α) der Taste an.

Schliesslich kommt optional noch ein Flag, das `+` oder `-` ist, wenn für diese Taste zur Ebenenumschaltung die linke oder rechte Shifttaste benutzt wird. Fehlt das Flag, wird für Tasten auf einem linken Finger die recht Shifttaste genommen und umgekehrt.

Für die Festlegung der linken und rechten Shifttaste gibt es `ShiftL` und `ShiftR`, die dieselben Argumente wie `Taste` erwarten. Die besonderen Schlüsselwörter werden verwendet, um sicherzustellen, dass beide Shifttasten tatsächlich im Konfigurationsfile festgelegt werden.

6.3 Die Bewertung festlegen

Einzelne Tasten

Die Einzeltastenaufwände werden mit den Tasten festgelegt, siehe Abschnitt 6.2. Die Zielhäufigkeit wird mit `Zielhäufigkeit` angegeben. `Zielhäufigkeit` hat zehn Argumente ($\tilde{\zeta}_{-5} \dots \tilde{\zeta}_{-1}, \tilde{\zeta}_1 \dots \tilde{\zeta}_5$), die unnormierten Zielhäufigkeiten für die Finger, beginnend mit dem linken kleinen Finger über linken und rechten Daumen bis zum rechten kleinen Finger. Mit `Fingerbelastung` werden entsprechend die zehn Gewichtsfaktoren ($\tilde{\phi}_{-5} \dots \tilde{\phi}_{-1}, \tilde{\phi}_1 \dots \tilde{\phi}_5$) angegeben, die bei der Überschreitung der Zielhäufigkeit benutzt werden, um eine Aufwand daraus zu bestimmen.

Bigramme

Für Bigramme gibt es viele Parameter. Am allgemeinsten ist `Bigramm`, das zwei Tastennamen (in der Tippreihenfolge) und eine Zahl für den Aufwand ($\beta_{tt'}^{\mathrm{ex}}$) erwartet. Dieser Aufwand wird zu dem mit den anderen Parametern ermittelten addiert.[4] Beispiel:

```
Bigramm AD03 AB03 -2,5
```

Wahlweise kann man am Ende einer `Bigramm`-Zeile einen String angeben. Bigramme mit demselben String gehören zum selben Bigrammtyp, und ihre summierte Häufigkeit erscheint in der Textausgabe. Beispiel:

```
Bigramm AC04 AC07 0 'Symmetrisch'
```

Dasselbe Bigramm kann in mehreren `Bigramm`-Zeilen vorkommen und somit zu verschiedenen selbstbestimmten Bigrammtypen gehören.

Die anderen Parameter für Bigramme drücken Aufwände in den Begriffen aus, die im Abschnitt 3 und 5 eingeführt wurden. Die Bigrammenklassen überlappen teilweise, zum Beispiel ist jede Auswärtsbewegung auch eine Handwiederholung. Die Aufwände, die für die verschiedenen Klassen veranschlagt werden, werden addiert.

[4] Bigramme einzeln anzugeben wird schnell unhandlich. Das mitgelieferte `gencfg.awk` zeigt, wie man sich mit einem einfachen Skript behelfen kann.

`Handwechsel`, `Handwiederholung` und `Auswärts` erwarten jeweils eine Zahl, den Aufwand für einen Handwechsel (ω_1), eine Handwiederholung (ω_0), und eine Auswärtsbewegung ($\omega_{\leftrightarrow}$; ein Bigramm mit Daumenbeteiligung zählt nie als solche).

Mit `DoppeltRabatt` wird ein Aufwand für Doppeltanschläge berechnet, indem das Argument (ϱ_0) mit der Differenz aus dem Einzeltastenaufwand der Taste in der Grundposition des zugehörigen Fingers und dem Einzeltastenaufwand der Taste selbst multipliziert wird. Diese Differenz ist normalerweise negativ, daher bekommt man mit `DoppeltRabatt` üblicherweise eine Aufwandsreduktion. Von der Idee her ähnlich ist `ZeilenwiederholungRabatt`, das fünf Zahlen ($\varrho_1 \dots \varrho_5$) erwartet. Wenn beide Anschläge mit verschiedenen Fingern (ohne Daumen) derselben Hand erfolgen und auf derselben Zeile liegen, diese nicht die mittlere Zeile (2) ist, die erste Taste im Bigramm nicht sowohl in der unteren Zeile liegt als auch mit dem kleinen Finger bedient wird, dann wird für die zweite Taste im Bigramm eine Differenz wie oben ermittelt und mit einem der Argumente von `ZeilenwiederholungRabatt` multipliziert. Das Argument wird gemäss der Spaltendifferenz der Tasten gewählt: Differenz 1 bedeutet erstes Argument, Differenz 2 zweites und so weiter. Zum Beispiel ist mit

```
ZeilenwiederholungRabatt 0.5 0.25 0.16666667 0.125 0.1
```

die Aufwandsreduktion bei Zeilenwiederholung umgekehrt proportional zur Spaltendifferenz.

Mit `KollisionKonstant` und `KollisionDistanz` kann man die Aufwände für Kollisionen festlegen. Beide Schlüsselwörter haben fünf Argumente ($\kappa_1 \dots \kappa_5$ beziehungsweise $\varkappa_1 \dots \varkappa_5$), eins pro Finger, beginnend mit dem Daumen auswärts. Der Aufwand ist der Wert aus `KollisionKonstant` plus das Produkt aus dem Wert aus `KollisionDistanz` und der Distanz, die der Finger bei der Kollision springen muss. Die Distanz ergibt sich aus den Koordinaten (x, y) der Tasten, siehe Abschnitt 6.2. Zum Beispiel legt

```
KollisionKonstant  10 10 10 10 10
KollisionDistanz   10 10 10 10 10
```

unter anderem fest, dass der Aufwand für Kollisionen unabhängig vom Finger ist.

Mit `Nachbar` werden die Aufwände für Nachbaranschläge festgelegt. Es gibt vier Argumente ($\nu_{3/2}, \nu_{5/2}, \nu_{7/2}, \nu_{9/2}$), für die Paarungen Daumen/Zeigefinger, Zeige/Mittelfinger, Mittel/Ringfinger und Ringfinger/Kleinfinger. Zum Beispiel werden mit

```
Nachbar  0 1.3333333 2 4
```

Nachbaranschläge als desto aufwändiger bewertet, je weiter aussen die beteiligten Finger liegen.

Für Handwiederholungen ohne Daumenbeteiligung gibt es zudem Aufwände für schräge Griffe, die mit `SchrägZS` und `SchrägYX` angegeben werden. Damit werden je zwei Zahlen angegeben (θ_{-1} und θ_1 beziehungsweise ϑ_{-1} und ϑ_1), die für die linke und rechte Hand gelten. Der Wert aus `SchrägZS` wird mit dem Verhältnis von Zeilendifferenz und Spaltendifferenz, der aus `SchrägYX` mit der Verhältnis von vertikalem zu horizontalem Abstand der Tasten multipliziert, um einen Aufwand zu erhalten. Zum Beispiel bewirkt

```
SchrägZS         0   1
SchrägYX         1   0
SchrägNenner0 0.1 0.1
```

dass schräge Griffe mit der linken Hand aufgrund der Koordinaten, solche mit der rechten Hand aufgrund der Spalten und Zeilen der Tasten berechnet werden. Mit `SchrägNenner0` werden hier zudem zwei Zahlen (Δ_{-1} und Δ_1, für die linke und rechte Hand) angegeben, die in dieser Berechnung sowohl zur Spaltendifferenz als auch zum horizontalen Abstand addiert werden.

Shift-Bigramme

Der Aufwand für ein Shift-Bigramm ergibt sich aus dem Aufwand des zugrunde liegenden Tastenbigramms durch Multiplikation mit einem der beiden Argumente (ς_+ und ς_-) von `Shiftbigramm`. Das erste wird benutzt, wenn der Aufwand des Tastenbigramms positiv ist, das zweite, wenn er negativ ist.

Trigramme

Die allgemeinste Festlegung von Trigrammaufwänden ist mit `Trigramm` möglich. Das funktioniert wie `Bigramm`, nur mit drei statt zwei Tastennamen.

Für Trigramme, deren erste und dritte Tasten mit derselben Hand angeschlagen werden, gibt es zusätzliche Beiträge. Zunächst wird der Aufwand für das Tastenbigramm aus erster und dritter Taste ermittelt. Er wird mit einem der beiden Werte (τ_+ und τ_-) multipliziert, die mit `Indirekt` angegeben werden: dem ersten, wenn der Aufwand positiv und dem zweiten, wenn er negativ ist. Wenn die mittlere Taste mit einer anderen Hand getippt wird, wird das Produkt als Zusatzaufwand genommen und dazu der mit `Doppelwechsel` angegebene Wert (ω_{11}) addiert. Wenn alle drei Tasten mit derselben Hand getippt werden, wird das Produkt mit der Summe der Bigrammaufwände von erster/zweiter und zweiter/dritter Taste verglichen; ist das Produkt grösser, ist der Zusatzaufwand das Produkt weniger der Summe, plus dem mit `Doppelwiederholung` angegeben Wert (ω_{00}). Sonst ist ist der Zusatzaufwand der mit `Doppelwiederholung` angegebene Wert.

Schliesslich wird für Trigramme, die aus einer Ein- und einer Auswärtsbewegung bestehen, der mit `Wippe` angegebene Wert ($\omega_{\rightleftarrows}$) zum Aufwand addiert.

Ähnlichkeit und Verwechslungspotenzial

Die allgemeinste Festlegung des Verwechslungspotenzials geht mit `Verwechslungspotenzial`. Das funktioniert wie `Bigramm`. Zusätzlich kann man Beiträge zum Verwechslungspotenzial von Tasten, die auf demselben (`VPKollision`, η) oder benachbarten Fingern (`VPNachbar`, υ) liegen angeben. `VPSymmetrisch` legt Beiträge (Σ) für Tasten fest, die mit denselben Finger an verschiedenen Händen bedient werden. Mit `VPSymmetrischGleicheZeile` gibt man Beiträge ($\Sigma_=$) für Tasten an, die zusätzlich noch in derselben Zeile liegen. `VPHandwechsel` (ϖ) ist für Tasten, die mit verschiedenen Händen bedient werden.

Mit `Ähnlich` wird die Ähnlichkeit (ε) zweier oder mehrerer Zeichen festlegt. Zum Beispiel kann man mit

```
Ähnlich 'bp' 0.1
Ähnlich 'sz' 0.05
```

sagen, dass sich «b» und «p» doppelt so ähnlich sind wie «s» und «z».

Vorlieben

Mit kann man bevorzugte Zuordnungen von Zeichen zu Tasten angeben. `Vorliebe` erwartet einen String mit Zeichen, die man derselben Tastenmenge zuordnen will, eine Zahl (λ), die die Stärke des Wunsches angibt, und die Menge Tasten, als Liste von Tastennamen. Um zu sagen, dass man x, c und v gerne auf den vier Tasten links unten hätte, kann man zum Beispiel angeben:

```
Vorliebe 'xcv' 0.1  AB01 AB02 AB03 AB04
```

Fehlende Zeichen

Mit `Fehlt` kann man einen Aufwand (Φ) für die Eingabe von Zeichen angeben, die in der Belegung nicht vorkommen. Zum Beispiel:

```
Fehlt 100
```

bewirkt, für dass jedes Zeichen, das im Korpus vorkommt und in der Belegung fehlt, ein Aufwand von 100 veranschlagt wird.

7 Die Aufwandsberechnung in Formeln

Wir gruppieren die Zeichen in der Belegung $\mathbb{B}$ in n Paare, ein Zeichen für die Grundebene und eins für die geshiftete Ebene. Jedes Zeichen $z \in \mathbb{B}$ ist durch ein Indexpaar $z = (p_z, e_z)$, mit $p_z \in \{0 \dots n-1\}$ und $e_z \in \{0, 1\}$, dargestellt. Für die n Zeichenpaare brauchen wir n Tasten, dazu zwei Shifttasten zur Ebenenwahl. Die Zeichentasten sind mit Null beginnend nummeriert, $t \in \{0 \dots n-1\}$. Die Shifttasten haben die Nummern n und $n+1$.

Eine Tasteneingabe i ist die Kombination von Zeichentaste t_i und Ebenenwahl e_i, dargestellt durch das Indexpaar $i = (t_i, e_i)$. Im Folgenden sind i, j und k solche Indexpaare, Summen über diese Indices laufen über alle Zeichentasten und Ebenen. t_i ist die erste Komponente (also die Taste) im Indexpaar i und e_i die zweite (die Ebene). Schliesslich ist s_i die zur Taste t_i gehörende Shifttaste.

Eine Belegung π ist eine Permutation von $(0 \dots n-1)$. Sie ordnet einer Tasteneingabe i ein Zeichen $z = \pi_i = (\pi_{t_i}, e_i)$ zu. Zeichen werden so immer paarweise Tasten zugeordnet. Die Ebene, auf der ein bestimmtes Zeichen liegt, bleibt fest.

Im Folgenden ist δ das Kronecker-Delta, $\bar{\delta}_{ij} = 1 - \delta_{ij}$ sein Komplement, Θ die Stufenfunktion

$$\Theta(x) = \begin{cases} 1 & \text{wenn } x > 0 \\ 0 & \text{sonst} \end{cases},$$

$\bar{\Theta}(x) = \Theta(-x)$ und $(x)_{\pm} = \pm x \Theta(\pm x)$ der Positiv- beziehungsweise der Negativteil von x.

7.1 Häufigkeiten

Die zu tippenden Texte sind durch Korpora gegeben, die wir mit einem Index $c \in K$ nummerieren. Korpus c hat ein Gewicht g_c, das sich durch Normierung aus den auf der Kommandozeile (mit `-G`) angegeben Gewichten ergibt, so dass $\sum_c g_c = 1$. Ein Korpus ist durch Tabellen mit Zeichenhäufigkeiten $H_i^{(c)}$, Bigrammhäufigkeiten $H_{ij}^{(c)}$ und Trigrammhäufigkeiten $H_{ijk}^{(c)}$ gegeben.

Sei $\mathbb{F}$ die Menge aller nicht-fixen Finger, f_t der Finger, der t bedient. Die Zahl von mit nicht-fixen Fingern getippter Zeichen in c ist $N^{(c)} = \sum_{f' \in \mathbb{F}} \sum_i \delta_{f' f_{t_i}} H_i^{(c)}$. Die Zahl der Anschläge mit nicht-fixen Fingern für c ist $T^{(c)} = \sum_{f' \in \mathbb{F}} \sum_i (\delta_{f' f_{t_i}} + e_i \delta_{f' f_{s_i}}) H_i^{(c)}$. Die relativen, gewichteten Gesamthäufigkeiten sind

$$h_i = \sum_{c \in K} g_c H_i^{(c)} / N^{(c)}, \tag{1a}$$

$$h_{ij} = \sum_{c \in K} g_c H_{ij}^{(c)} / N^{(c)} \quad \text{und} \tag{1b}$$

$$h_{ijk} = \sum_{c \in K} g_c H_{ijk}^{(c)} / N^{(c)}. \tag{1c}$$

Die relative Häufigkeit aller Zeichen, die im Korpus, aber nicht in der Belegung sind, ist

$$\bar{h} = \frac{\sum_{c\in K}\sum_{z\notin\mathbb{B}} g_c H_z^{(c)}}{\sum_{c\in K}\sum_{i\in\mathbb{B}} g_c H_i^{(c)}}\,.$$

7.2 Aufwände

Der gesamte Aufwand ist die Summe mehrerer Beiträge,

$$A(\pi) = A_0(\pi) + A_1(\pi) + A_2(\pi) + A_3(\pi) + A_\mathrm{f}(\pi) + A_\mathrm{fehlt}\,. \tag{2}$$

In der Textausgabe wird $100A$ als «Gesamtaufwand» angezeigt.

Der Beitrag zum Gesamtaufwand von Einzeltasten ist

$$A_1(\pi) = \sum_i a_i h_{\pi_i}\,, \tag{3}$$

wobei $a_i = \alpha_{t_i} + e_i(\alpha_{s_i} + \beta_{s_i t_i})$ und die α die im Konfigurationsfile angegebenen Aufwände pro Taste sind. In der Textausgabe wird $100A_1$ als «Lageaufwand» angezeigt.

Die Beiträge zum Gesamtaufwand von den Zeichenbigrammen und Zeichentrigrammen sind

$$A_2(\pi) = \sum_{ij} a_{ij} h_{\pi_i\pi_j}\,, \tag{4}$$

$$A_3(\pi) = \sum_{ijk} a_{ijk} h_{\pi_i\pi_j\pi_k}\,, \tag{5}$$

mit

$$a_{ij} = (1-e_j)\Big[\beta_{t_i t_j} + e_i\varsigma(\beta_{s_i t_j})\Big] + e_j\Big[\beta_{t_i s_j} + \gamma_{t_i s_j t_j} + e_i\varsigma(\beta_{s_i s_j})\Big]\,, \tag{6}$$

$$a_{ijk} = (1-e_j)\Big[(1-e_k)\gamma_{t_i t_j t_k} + e_k\gamma_{t_i t_j s_k}\Big]\,, \tag{7}$$

siehe auch Tabelle 1 und 2. Die β und γ sind die Aufwände für Tastenbigramme und Tastentrigramme,

$$\begin{aligned}
\beta_{tt'} = {}& \beta^\mathrm{ex}_{tt'} + \delta_{1|f_t-f_{t'}|}\nu_{|f_t+f_{t'}|/2} + \delta_{tt'}\varrho_0(\alpha_{\Gamma_t} - \alpha_t) + \\
&+ \Theta(f_t f_{t'})\,\bar\delta_{tt'}\Big[\omega_0 + \Theta(|f_{t'}| - |f_t|)\,\bar\delta_{1|f_t|}\omega_{\leftrightarrow}\Big] + \bar\Theta(f_t f_{t'})\,\omega_1 + \\
&+ \Theta(f_t f_{t'})\,\bar\delta_{f_t f_{t'}}\bar\delta_{1|f_t|}\bar\delta_{1|f_{t'}|}\delta_{\rho_t\rho_{t'}}\bar\delta_{\rho_t 2}\left(1 - \delta_{3\rho_t}\delta_{5|f_t|}\right)\varrho_{|\xi_t-\xi_{t'}|}(\alpha_{\Gamma_{t'}} - \alpha_{t'}) + \\
&+ \lim_{\delta\to\Delta_{\chi_t}} \Theta(f_t f_{t'})\,\bar\delta_{f_t f_{t'}}\bar\delta_{1|f_t|}\bar\delta_{1|f_{t'}|}\left(\frac{\theta_{\chi_t}|\rho_t - \rho_{t'}|}{|\xi_t - \xi_{t'}| + \delta} + \frac{\vartheta_{\chi_t}|\gamma_t - \gamma_{t'}|}{|x_t - x_{t'}| + \delta}\right) + \\
&+ \bar\delta_{tt'}\delta_{f_t f_{t'}}\left(\kappa_{|f_t|} + \varkappa_{|f_t|}\sqrt{(x_t - x_{t'})^2 + (y_t - y_{t'})^2}\right)
\end{aligned}$$

und

$$\begin{aligned}
\gamma_{tt't''} = {}& \gamma^\mathrm{ex}_{tt't''} + \Theta(f_t f_{t''})\Big\{\bar\Theta(f_t f_{t'})\left(\omega_{11} + \beta^\mathrm{ind}_{tt''}\right) + \\
&+ \Theta(f_t f_{t'})\Big[\omega_{00} + \Theta\left(\beta^\mathrm{ind}_{tt''}\right)\left(\beta^\mathrm{ind}_{tt''} - \beta_{tt'} - \beta_{t't''}\right)_+ + \Theta\Big((f_t - f_{t'})(f_{t''} - f_{t'})\Big)\omega_{\rightleftarrows}\Big]\Big\}\,,
\end{aligned}$$

wobei $\chi_t = \Theta(f_t) - \bar\Theta(f_t)$, $\beta^\mathrm{ind}_{tt'} = \tau_+(\beta_{tt'})_+ - \tau_-(\beta_{tt'})_-$ und $\varsigma(x) = \varsigma_+(x)_+ - \varsigma_-(x)_-$ ist. Γ_t ist die Taste in der Grundposition von f_t. Tasten mit gleicher Zeile und Spalte gelten als gleich, $\delta_{tt'} = \delta_{\rho_t\rho_{t'}}\delta_{\xi_t\xi_{t'}}$.

Ist $\zeta_f = \tilde{\zeta}_f / \sum_{f'} \tilde{\zeta}_{f'}$ die normierte Zielhäufigkeit für Finger f, die sich aus den $\tilde{\zeta}_f$ des Konfigurationsfiles ergibt, ist der Beitrag der Fingerbelastung zum Gesamtaufwand

$$A_{\mathrm{f}}(\pi) = \sum_{c \in K} g_c \sum_{f \in \mathbb{F}} \phi_f \left(b_f^{(c)}(\pi) - \zeta_f \right)_+^2 , \tag{8}$$

wobei $b_{f'}^{(c)}(\pi) = \sum_i (\delta_{f' f_{t_i}} + e_i \delta_{f' f_{s_i}}) H_{\pi_i}^{(c)} / T^{(c)}$ die Belastungen des Finger f' für Korpus c ist und $\phi_f = \tilde{\phi}_f$ falls ohne und $\phi_f = (1 + \tau_+)\, \tilde{\phi}_f$ falls mit Trigrammen bewertet wird.

Die nichtmechanischen Kriterien setzen sich aus Verwechselbarkeit und Vorlieben zusammen,

$$A_0(\pi) = \sum_{t=0}^{n-1} \sum_{t'=0}^{n-1} \varphi_{tt'} \varepsilon_{\pi_t \pi_{t'}} - \sum_{t=0}^{n-1} \lambda_{\pi_t t} . \tag{9}$$

φ ist das Verwechslungspotenzial zweier Tasten,

$$\varphi_{tt'} = \varphi_{tt'}^{\mathrm{ex}} + \bar{\delta}_{tt'} \delta_{f_t f_{t'}} \eta + \delta_{1|f_t - f_{t'}|} \upsilon + \left[1 - \Theta(f_t f_{t'}) \right] \varpi + \\ + \bar{\Theta}(f_t f_{t'})\, \bar{\delta}_{1|f_t|} \bar{\delta}_{1|f_{t'}|} \delta_{|f_t||f_{t'}|} \left(\bar{\delta}_{\rho_t \rho_{t'}} \Sigma + \delta_{\rho_t \rho_{t'}} \Sigma_= \right) ,$$

ε die Ähnlichkeit zweier Zeichen und λ die Vorliebe dafür, ein bestimmtes Zeichenpaar auf eine bestimmte Taste zu legen.

Der Aufwand für in der Belegung fehlende Zeichen ist $A_{\mathrm{fehlt}} = \bar{h} \Phi$.

7.3 Pareto-Eigenschaft

Dass die Belegung, die A minimiert, Pareto-optimal bezüglich der einzelnen Korpora ist, liegt daran, dass die Aufwände $A^{(c)}$ der einzelnen Korpora sich zum Gesamtaufwand summieren,

$$A = \sum_{c \in K} g_c A^{(c)} . \tag{10}$$

Wäre das Optimum π_0 nicht Pareto-optimal, gäbe es eine Belegung π_1 und ein $c_0 \in K$ mit $A^{(c_0)}(\pi_1) < A^{(c_0)}(\pi_0)$ und $A^{(c)}(\pi_1) \leq A^{(c)}(\pi_0)$ für $c \neq c_0$. Damit würde jedoch die Summe kleiner, was der Annahme widerspricht, dass π_0 optimal ist.

Die Summeneigenschaft (10) für A_{f} ist explizit in Gleichung (8). A_1, A_2 und A_3 (Gleichung (3), (4) und (5)) sind linear in den h, daher kann man die c-Summation aus der Definition (1) der h herausziehen, zum Beispiel $A_1(\pi) = \sum_i a_i \sum_c g_c H_{\pi_i}^{(c)} / N^{(c)} = \sum_c g_c \sum_i a_i H_{\pi_i}^{(c)} / N^{(c)} = \sum_c g_c A_1^{(c)}(\pi)$. A_0 in Gleichung (9) schliesslich ist korpusunabhängig, somit $A_0(\pi) = A_0^{(c)}(\pi)$, und die Summeneigenschaft gilt wegen $\sum_c g_c = 1$.

Verwendet man hingegen ein einzelnes Korpus, den man extern aus verschieden Teilkorpora zusammengesetzt hat, verhindert die Nichtlinearität von Gleichung (8) in h, dass man garantieren kann, dass das Gesamtoptimum bezüglich der Teilkorpora Pareto-optimal ist.

7.4 Statistischer Fehler

Wir betrachten zunächst ein einziges Korpus. Wir stellen uns vor, dieser sei eine Stichprobe aus einem unendlich grossen, idealen Korpus und dadurch gebildet, dass wir wiederholt zufällig und voneinander unabhängig ein Wort aus dem idealen Korpus wählen. Diese Annahme ist die entscheidende Vereinfachung. In Wirklichkeit sind die unkorrelierten Einheiten wohl grössere als einzelne Wörter.

Die Wahrscheinlichkeit, bei einer Wortwahl das Wort w zu wählen, sei p_w. Die Häufigkeit, mit der das Wort w ausgewählt wird, ist binominalverteilt mit Mittelwert $\mu_w = p_w H_{\text{wort}}$ und Varianz $\sigma_w^2 = H_{\text{wort}} p_w (1 - p_w)$. Wir können diese Parameter aus den beobachteten Häufigkeiten als $\mu_w = H_w$ und $\sigma_w^2 = H_w (1 - H_w / H_{\text{wort}})$ schätzen, wobei H_w die Häufigkeit des Worts w und H_{wort} die Grösse der Stichprobe sind.

Die Häufigkeit, mit der ein Zeichen i und das Zeichenbigramm ij in w vorkommen, bezeichnen wir mit w_i und w_{ij}. Die Anzahl der Zeichen in einem Wort ist $|w| = \sum_i w_i$, die Anzahl der Anschläge, die man zur Eingabe benötigt, $\|w\| = \sum_i (1 + e_i)\, w_i$. Damit kann man die Zahl der Zeichen und der Anschläge in der Stichprobe als $N = \sum_w |w| H_w$ und $T = \sum_w \|w\| H_w$ schreiben, die relative Häufigkeit des Zeichens i als $h_i = \sum_w w_i H_w / N$ und so weiter.

Die Worthäufigkeiten in einer Stichprobe werden von ihrem Erwartungswert abweichen. Die Abweichung $\delta A(\pi^{-1})$ vom Erwartungswert des Aufwands $A(\pi^{-1})$ der Belegung π^{-1} aufgrund dieser Abweichungen δH_w ist

$$\delta A(\pi^{-1}) = \sum_w \left[\sum_i a_{\pi_i} \frac{w_i - |w|\, h_i}{N} + \sum_{ij} a_{\pi_i \pi_j} \frac{w_{ij} - |w|\, h_{ij}}{N} + \right.$$
$$\left. + \sum_{f' \in \mathbb{F}} 2 \phi_{f'} \left(b_{f'} - \zeta_{f'}\right)_+ \sum_i (\delta_{f' f_{t_{\pi_i}}} + e_i \delta_{f' f_{s_{\pi_i}}}) \frac{w_i - \|w\| \bar{h}_i}{T} \right] \delta H_w \,,$$

wobei $\bar{h}_i = H_i / T$ und die Fingerbelastung und die Gesamtzeichenzahl um die Erwartungswerte der Worthäufigkeiten linearisiert wurden. Da wir die Häufigkeiten verschiedener Wörter als unabhängig betrachten ist $\langle \delta H_w \delta H_{w'} \rangle = \sigma_w^2 \delta_{ww'}$, wobei $\langle \cdot \rangle$ der Erwartungswert ist. Damit wird

$$\left\langle \delta A^2(\pi^{-1}) \right\rangle = \sum_w \left[\sum_i a_{\pi_i} \frac{w_i - |w|\, h_i}{N} + \sum_{ij} a_{\pi_i \pi_j} \frac{w_{ij} - |w|\, h_{ij}}{N} + \right.$$
$$\left. + \sum_{f' \in \mathbb{F}} 2 \phi_{f'} \left(b_{f'} - \zeta_{f'}\right)_+ \sum_i (\delta_{f' f_{t_{\pi_i}}} + e_i \delta_{f' f_{s_{\pi_i}}}) \frac{w_i - \|w\| \bar{h}_i}{T} \right]^2 H_w \left(1 - \frac{H_w}{H_{\text{wort}}}\right) .$$

Zur Auswertung multipliziert man die eckigen Klammern aus. Aus den Produkten der Terme $(w_i - |w|\, h_i)/N$, $(w_{ij} - |w|\, h_{ij})/N$ und $\left(w_i - \|w\| \bar{h}_i\right)/T$ entstehen sechs Korrelationen, zum Beispiel

$$\sigma_{ij;k}^{(2\text{f})} = \sum_w \frac{w_{ij} - |w|\, h_{ij}}{N} \frac{w_k - \|w\| \bar{h}_k}{T} H_w \left(1 - \frac{H_w}{H_{\text{wort}}}\right) .$$

Man kann sie direkt aus dem Korpus gewinnen und daraus $\langle \delta A^2 \rangle$ leicht berechnen,

$$\left\langle \delta A^2(\pi^{-1}) \right\rangle = \ldots + 2 \sum_{ijk} a_{\pi_i \pi_j} \sum_{f' \in \mathbb{F}} \phi_{f'} \left(b_{f'} - \zeta_{f'}\right)_+ (\delta_{f' f_{t_{\pi_k}}} + e_i \delta_{f' f_{s_{\pi_k}}}) \sigma_{ij;k}^{(2\text{f})} + \ldots$$

Um die Varianz einer Aufwandsdifferenz zu berechnen, ersetzt man die π-abhängigen Vorfaktoren durch Differenzen für die beiden Belegungen π und π', zum Beispiel $a_{\pi_i \pi_j}$ durch $a_{\pi_i \pi_j} - a_{\pi'_i \pi'_j}$. Die Gesamtvarianz für mehrere Korpora ergibt sich aus den mit g_c^2 gewichtetet Einzelvarianzen.

8 Anhang

8.1 Die mitgelieferten Häufigkeitsfiles

Das deutsche Korpus, der den Files `deutsch.txt.*` zugrunde liegt, setzt sich so zusammen:

- 636 kB aus zufällig gewählten Zeilen des von Karl Köckemann nachbearbeiteten deutschsprachigen Leipziger Korpus. Das Orginalkorpus enthält 3 Millionen Sätze aus verschiedenen Zeitungen. Karls Bearbeitung passt ihn an die neue Rechtschreibung an.
- 642 kB aus den Essays, die in der Zeitschrift c't zwischen 22/1999 und 12/2006 publiziert wurden. Diese Essays benutzen die neue Rechtschreibung und liegen als HTML vor. Die HTML-Formatierung wurde entfernt, ebenso Überschriften und Paragrafen, die `<a>`, `<code>` und ähnliches enthalten.
- 150 kB Sachtexte und 200 kB literarische Texte von Project Gutenberg:
  Sigmund Freud, Massenpsychologie und Ich-Analyse, Kapitel I und II; Arnold Sommerfeld, Relativitätstheorie, aus Deutsches Leben in der Gegenwart; Mihai Nadin, Jenseits der Schriftkultur, 40 kB aus Buch I, Kapitel I; Alexander Lipschütz, Warum wir sterben (ca. 40 kB); Gerhart Hauptmann, Bahnwärter Thiel (ca. 20 kB vom Anfang); Stefan Zweig, Brennendes Geheimnis (ca. 20 kB vom Anfang); Robert Walser, Der Gehülfe (ca. 20 kB vom Anfang); Gottfried Keller, Die Leute von Seldwyla, Vol I (ca. 20 kB vom Anfang); Franz Kafka, Die Verwandlung (ca. 20 kB vom Anfang); Alfred Döblin, Die Ermordung einer Butterblume und andere Erzählungen (ca. 20 kB vom Anfang); Hermann Hesse, Siddhartha (ca. 20 kB vom Anfang); Thomas Mann, Der Tod in Venedig (ca. 20 kB vom Anfang); Rainer Maria Rilke, Zwei Prager Geschichten (ca. 20 kB vom Anfang).
  Diese Texte verwenden die alte Rechtschreibung, lediglich «daß» wurde durch «dass» ersetzt und Paragrafen zu Zeilen zusammengezogen.

Für die `deutsch-t.txt`-Files wurden die Trennmuster `hyph-de-1996.pat.txt` (Stand Mai 2014) und dasselbe Korpus verwendet.

Das englische Korpus, der den Files `englisch.txt.*` zugrunde liegt, setzt sich so zusammen:

- 920 kB aus zufällig gewählten Zeilen des englischen Leipziger Korpus. Amerikanische Schreibweisen wurden teilweise in englische umgewandelt (zum Beispiel «center» in «centre»). Einige Einheiten wurden durch international gültiges ersetzt (zum Beispiel «miles» durch «kilometres»).
- 417 kB Sachtexte:
  - Aus O'Reilley open books: Stephen L. Talbott, The future does not compute, Kapitel 23, gefiltert wie die c't-Essays.
  - Von Project Gutenberg:
    Charles Darwin, On the Origin of Species, Kapitel I; James Clerk Maxwell, Five of Maxwell's Papers.
  - Aus dem Open American National Corpus, Verzeichnis `written_2/non-fiction/OUP`:
    Abernathy, Kapitel 3; Berk, Kapitel 7; Fletcher, Kapitel 10; Kauffmann, Kapitel 5; Rybczynski, Kapitel 1.
- 339 kB Literatur von Project Gutenberg:
  Lewis Carroll, Alice in Wonderland; Joseph Conrad, Heart of Darkness, Kapitel; W. Somerset Maugham, Of Human Bondage, Kapitel 1–16; James Joyce, Ulysses, ca. 2500 Zeilen.

Für die `englisch-t.txt`-Files wurden die Trennmuster `hyph-en-gb.pat.txt` (Stand Mai 2010) und dasselbe Korpus verwendet.

8.2 Beispielbelegungen

Näheres zu den Tastaturbelegungen in `bsptast.txt`: Arensito, Aus der Neo-Welt, Colemak, Klausler, KOY, Neo 2. Zum Teil habe ich die Umlaute selbst ergänzt.

8.3 Weitere Informationen

Der Optimierer hat eine Homepage, über die die aktuelle Version und weitere Informationen verfügbar sind. Ein guter Startpunkt, um mehr über den Hintergrund, auf dem Beurteilungen von Belegungen beruhen, zu lernen, ist `http://adnw.de`.