Bretter, die die Welt bedeuten

Einige Fragen zum Beitrag »Hyphenation Exception Log für deutsche Trennmuster, Version 1«

Stephan Hennig

Dieser Beitrag versucht einige Fragen aufzuwerfen, die die Beiträge von Werner Lemberg zum Thema Trennmuster und Ausnahmelisten in »Die TEXnische Komödie« 2/2003 und 2/2005 bisher unbeantwortet ließen.

Rückblick

In »Die TEXnische Komödie« 2/2003 rief Werner Lemberg dazu auf, Fehler in den Trennmustern für die deutschen Rechtschreibung zu sammeln und diese in einer öffentlichen Ausnahmeliste zusammenzufassen.[4] In der Ausgabe 2/2005 beschrieb er Methoden und Skripte, mit denen die Suche nach Fehlern in den Trennmustern für die traditionelle und neue deutsche Rechtschreibung vereinfacht werden kann und stellte eine erste Version der Ausnahmeliste für die traditionelle Rechtschreibung vor.[5]

Beide Beiträge sind interessant, jedoch bleiben meiner Meinung nach eine Reihe von Problemen unbeantwortet. Ich stelle meine Fragen hier in der Hoffnung, dass sie von allgemeinem Interesse sind und die eine oder andere Seite Antworten darauf geben kann. Auch wenn Werner Lembergs Beitrag als Aufhänger dient, richten sich die folgenden Fragen nicht nur an ihn, sondern an alle Mitglieder.

Fragenkatalog

1. Aus welchem Grund wurde eine Ausnahmeliste für die traditionelle Rechtschreibung vorgestellt und nicht für die neue?

2. Gibt es Erkenntnisse zur Qualität der Trennmuster für die deutsche Rechtschreibung, sowohl die traditionelle als auch die neue? Wie gut oder schlecht sind die existierenden Trennmuster?
3. Gibt es einen Bedarf für neuberechnete Trennmuster?
4. Ist die Arbeit an Ausnahmelisten zur Zeit sinnvoll?
5. Welche Berührungspunkte gibt es mit anderen freien Programmen zur Textbearbeitung?
6. Wie kann eine Infrastruktur aussehen, die verteiltes Arbeiten an zu schaffenden Listen korrekter Worttrennungen für die traditionelle und neue Rechtschreibung ermöglicht?
7. Was sind gute Wortquellen?
8. Wie funktioniert die Worttrennung in TEX?
9. Wie werden Trennmuster erstellt?
10. Welchen Einfluss haben ungewöhnliche Wörter auf die Trennmuster?
11. In welcher Form kann und möchte DANTE e.V. unterstützend bei der Pflege deutscher Trennmuster und Ausnahmelisten wirken?
12. Ist es zweckmäßig auch Trennmuster und Ausnahmelisten nur mit Haupttrennstellen zu erstellen?

Anmerkungen zu den Fragen

Es folgen Anmerkungen (keine Antworten) zu den aufgeworfenen Fragen:

Zu 1) Aus welchem Grund wurde eine Ausnahmeliste für die traditionelle Rechtschreibung vorgestellt und nicht für die neue?

Zur vorgestellten Ausnahmeliste für die traditionelle Rechtschreibung bemerkt Werner Lemberg in [5]:

> Entfernt man alle Zeilen, die [. . .] »-st« enthalten, kann man die folgenden Korrekturen auch mit den neuen deutschen Trennmustern benützen . . .

Dieses Vorgehen birgt Probleme. Zum Beispiel wird das Wort »Abendstern« mit `dehypht.tex`[1] korrekt »Abend-stern« getrennt. Mit `dehyphn.tex` wird es dagegen falsch »Abends-tern« getrennt. Zum einen wäre der Eintrag »Abendstern« in einer gemeinsamen Ausnahmeliste redundant, falls die traditionelle Rechtschreibung verwendet wird. Zum anderen würde er gestrichen, sobald

[1] Die Datei `dehypht.tex` enthält die Trennmuster für die *t*raditionelle, `dehyphn.tex` diejenigen für die *n*eue deutsche Rechtschreibung. Die Trennmuster werden beim Laden der Pakete `german`, `ngerman` oder `babel` mit den entsprechenden Optionen aktiviert.

nach der oben angegebenen Regel die Ausnahmeliste für die neue Rechtschreibung abgeleitet wird. Diese fehlerhafte Trennung ist so nicht korrigierbar. Aus technischer Sicht wäre es daher vorzuziehen, zwei getrennte Ausnahmelisten zu führen. Frage 1 könnte daher auch anders formuliert werden: Welche Ausnahmelisten (und Trennmuster) sollten gepflegt werden? Sprechen nichttechnische Gründe dafür, nur die einen oder die anderen zu pflegen?

Zu 2) Gibt es Erkenntnisse zur Qualität der Trennmuster für die deutsche Rechtschreibung, sowohl die traditionelle als auch die neue? Wie gut oder schlecht sind die existierenden Trennmuster?

Werner Lemberg hat in [5] einige Beobachtungen zu den Trennmustern für die traditionelle Rechtschreibung zusammengestellt. Meine Meinung zu den existierenden Trennmustern ist: Sie sind akzeptabel. Das heißt, LaTeX trennt nicht auffallend besser oder schlechter als andere Anwendungen. Eine manuelle Kontrolle der Trennungen ist in jedem Fall notwendig, einige Korrekturen sind es meistens auch. Systematische Betrachtungen zu den Trennmustern sind mir allerdings nicht bekannt. Existieren solche?

Zu 3) Gibt es einen Bedarf für neuberechnete Trennmuster?

Bei der Beantwortung dieser Frage spielen meiner Meinung nach mehrere Punkte eine Rolle:

1. Ausnahmelisten sind unbestritten sinnvoll, da in der deutschen Sprache beliebige Wortzusammensetzungen und somit Buchstabenkombinationen gebildet werden können. Dadurch verlieren die Buchstabenmuster, an denen sich der Trennalgorithmus orientiert, an Signifikanz und fehlerhafte Worttrennungen sind unvermeidlich. Die Trennalgorithmen sollten im Idealfall jedoch auf Komposita beschränkt bleiben. Werner Lembergs Ausnahmeliste `dehyphtex.tex` (Stand: 28. 1. 2006) enthält 2090 fehlerhafte und fehlende Trennungen *einfacher* Wörter. Ab welcher Zahl besteht ein Bedarf für neue Trennmuster?

2. In [5] berichtet Werner Lemberg, dass die originale Wortliste »verschollen« ist, aus der die Trennmuster für die traditionelle Rechtschreibung erzeugt wurden. Die Trennmuster können daher nicht jederzeit wieder neu erstellt werden. Für eine freie Software wie TeX ist dies kein wünschenswerter Zustand.

3. Die Trennmuster für die neue Rechtschreibung `dehyphn.tex` wurden (aus diesem Grund) nicht aus einer Wortliste berechnet. Walter Schmidt hat in Handarbeit die Datei `dehyphn.tex` an die modifizierte s-t-Trennregel,

> vermehrte Doppel-s-Schreibung und weitere Änderungen angepasst. So ist auch die unterschiedliche Trennung von »Abendstern« zu erklären. Allerdings wird auch das einfache Wort »Fassade« mit `dehypht.tex` korrekt getrennt, mit `dehyphn.tex` (Stand: 7. 5. 2001, Revisionlevel 31) jedoch die erste Trennstelle nicht erkannt. Können solche Fehler durch neuberechnete Trennmuster für die neue Rechtschreibung vermieden oder verringert werden?

Zu 4) Ist die Arbeit an Ausnahmelisten zur Zeit sinnvoll?

Falls Frage 3 positiv beantwortet werden kann, stellt sich die Frage, welchen Nutzen Ausnahmelisten zum jetzigen Zeitpunkt bringen. Durch mühsame Arbeit können damit zwar einige fehlerhafte Trennungen korrigiert werden, systematische Fehler in den Trennmustern jedoch nicht. Dies gilt umso mehr, da die originale Wortliste nicht mehr zur Verfügung steht und deren Qualität daher schlecht eingeschätzt werden kann.

Wäre es möglicherweise sinnvoller, sich zunächst auf das Erstellen neuer Trennmuster zu konzentrieren? Dabei sollten die bekannten Trennfehler einfacher Wörter selbstverständlich in die Erstellung einbezogen werden. Allerdings wird für das Erstellen neuer Trennmuster weit mehr benötigt als eine Liste einiger derzeit falsch getrennter Wörter. Nämlich eine qualitativ möglichst hochwertige Liste getrennter deutscher Wörter. Zwei Qualitätsmerkmale der Liste sollten nach Werner Lemberg der Listenumfang an einfachen und zusammengesetzten Wörtern und die Fehlerfreiheit sein.[5]

> Im Gegensatz zu früher ist es wohl nicht so dringend notwendig, die Trennmuster klein zu halten [. . .] sollte eher Augenmerk auf wirklich fehlerfreie Trennung einer möglichst großen Anzahl von einfachen und zusammengesetzten Wörtern gelegt werden.

Zu den Fehlern in der vorliegenden Wortliste urteilt Werner Lemberg in [5]:

> Mit großer Wahrscheinlichkeit habe ich etliche Einträge übersehen oder falsch behandelt.

Sofern in die *Ausnahmeliste* nur fehlerfreie Einträge aufgenommen werden, sind unentdeckte Trennfehler unproblematisch, da sie keinen zusätzlichen Schaden anrichten. Sollen jedoch *Trennmuster* erzeugt werden, sind Fehler in der zugrundeliegenden Wortliste unbedingt zu vermeiden. Der nächste Schritt wäre meiner Meinung nach, auf Werner Lembergs Arbeit aufzubauen und eine zweite, dritte, vierte und x-te Kontrolle der existierenden Liste(n) durch

»andere Augen« durchzuführen. Daran können und sollten sich möglichst viele beteiligen. Die dazu erforderliche Infrastruktur wird in Frage 6 thematisiert.

Zu 5) Welche Berührungspunkte gibt es mit anderen freien Programmen zur Textbearbeitung?

Neben TEX gibt es eine ganze Reihe von freien Programmen zum Verfassen und Bearbeiten von Texten. Wie werden die Probleme der Rechtschreibkontrolle und Worttrennung dort gelöst? Können die existierenden Wortlisten für die Nutzung mit TEX aufbereitet werden? Sind dort Trennungen enthalten? Gibt es lizenzrechtliche Probleme? Beispielhaft seien Programme wie `OpenOffice`, `Aspell` oder `SiSiSi` erwähnt.

Zu 8) Wie funktioniert die Worttrennung in TEX?

Die folgenden beiden Fragen betreffen die grundlegende Arbeitsweise des Programms TEX: Absatzumbruch und Worttrennungen sind in TEX eng miteinander verwoben. Der Schwerpunkt dieser Frage soll darauf liegen, wie aus den Trennmustern mögliche Trennstellen ermittelt werden.

Zu 9) Wie werden Trennmuster erstellt?

Hier geht es um das umgekehrte Problem, wie werden aus Listen mit Worttrennungen Trennmuster abgeleitet?

Die letzten beiden Fragen finde ich auch abseits der hier behandelten Problematik interessant. Ich wundere mich, wie wenige Beiträge in »Die TEXnische Komödie« die Grundlagen von TEX und LATEX beleuchten. Gehört »The TEXbook« [3] zum Allgemeinwissen eines jeden LATEX-Anwenders? Ich kann von mir leider nicht behaupten, dieses Buch vollständig gelesen und verstanden zu haben.

Zu 10) Welchen Einfluss haben ungewöhnliche Wörter auf die Trennmuster?

Gibt die Antwort auf Frage 9 Hinweise darauf, ob es Wörter gibt, die *nicht* in Trennmuster aufgenommen werden sollten, da sie deren Qualität (Signifikanz und Trennschärfe bestimmter Buchstabenmuster) verringern oder die Trennmuster deutlich vergrößern würden? Welche Wörter sind dies? Komposita, wie zum Beispiel »Stabs-chef«, mit einer Trennung innerhalb der sonst untrennbaren Buchstabenkombination »sch« wurden bereits erwähnt. Gibt es Wörter, die grundsätzlich besser durch Ausnahmelisten abgedeckt werden (Fremdwörter, Eigennamen, geografische Bezeichnungen)?

Zu 11) In welcher Form kann und möchte DANTE e.V. unterstützend bei der Pflege deutscher Trennmuster und Ausnahmelisten wirken?

Ansprechpartner zu den Trennmusterdateien sind zur Zeit Walter Schmidt für `dehyphn.tex` und Bernd Raichle bzw. »DANTE e.V., Koordinator german.sty« für `dehypht.tex`. Für die Ausnahmeliste `dehyphtex.tex` ist es Werner Lemberg. Ist DANTE e.V. in die Pflege institutionell eingebunden? Kann DANTE e.V. die Arbeit an Wortlisten unterstützen, zum Beispiel durch Bereitstellung von technischer Infrastruktur oder Projektmitteln? Da ich mir wenige Aufgaben vorstellen kann, die so unmittelbar im Fokus von DANTE e.V. stehen sollten wie die langfristige Pflege von deutschen Wortlisten, Trennmustern und Ausnahmelisten, erscheint mir eine Unterstützung der derzeit damit betrauten Personen durch den Verein sinnvoll (sofern dies nicht bereits der Fall ist).

Zu 12) Ist es zweckmäßig auch Trennmuster und Ausnahmelisten nur mit Haupttrennstellen zu erstellen?

Werner Lemberg hat in [4] verschiedene Ansätze erwähnt, die sich dem Problem der Unterscheidung zwischen Haupt- und Nebentrennstellen widmen. Die Ansätze von Clasen und Sojka benötigen dafür Trennmuster, die gewichtete oder nur Haupttrennstellen enthalten.[8, 1]

Auch ohne die beschriebenen Erweiterungen des Trennalgorithmus können Trennmuster sinnvoll sein, die nur Haupttrennstellen enthalten. Verwendet man in einem Dokument zum Beispiel eine ausreichende Zeilenlänge, so kann durch Trennungen ausschließlich an Haupttrennstellen die Lesbarkeit des Textes erhöht werden. Würden diese zusätzlichen Trennmuster als weitere Sprachen in Babel integriert, könnte man als Hauptsprache für ein Dokument zum Beispiel `ngerman-ht` wählen und nur für kritische Textstellen, für welche aus irgendeinem Grund mit diesen Trennmustern kein befriedigender Absatzumbruch gefunden werden kann, mit einem Sprachwechsel vorübergehend die herkömmlichen, umfangreicheren Trennmuster `ngerman` aktivieren.

Die Pflege dieser Trennmuster würde wohl weitgehend unabhängig von der der herkömmlichen Trennmuster erfolgen müssen. Zum Beispiel würde die Ausnahmeliste wahrscheinlich erheblich mehr Komposita enthalten müssen und diese mit anderen Trennstellen. Damit würde sich die Zahl der zu verwaltenden Dateien verdoppeln. Besteht ein Bedarf an Trennmustern und Ausnahmelisten nur mit Haupttrennstellen?

Anmerkungen zu Werner Lembergs Skripten

Abschließend möchte ich Änderungen an zwei der von Werner Lemberg vorgestellten Skripten vorschlagen: `strippunct.sed` und `prepare-wordlist.sh`. Außerdem wird eine neue Version des Skripts `log2words.sed` vorgestellt, welche auch von Werner Lemberg stammt.

Kurz zur Funktionsweise des bisherigen Skripts `strippunct.sed` (vgl. Listing 5 in [5]): In Zeile 1 wird eine Auswahl an Sonderzeichen entfernt, jedoch bleiben einige nichtalphabetische Zeichen erhalten. In den Zeilen 2 und 3 werden Wörter entfernt, die einen Bindestrich oder ein (falsches) Apostroph im Wortinnern, am Wortanfang oder -ende enthalten. Zeile 4 ersetzt schließlich Leerzeichen durch Zeilentrennzeichen, damit alle verbleibenden Wörter in eine neue Zeile geschrieben werden.

Eine Schwäche des bisherigen Skripts ist, dass nicht alle Sonderzeichen entfernt werden. Beispielsweise verbleiben mit Werner Lembergs Skript in der aus dem Freedict-Wörterbuch (siehe letzter Abschnitt) extrahierten Wortliste viele geklammerte Wörter. Mir fällt jedoch kein Grund ein, aus dem in der aufbereiteten Wortliste irgendwelche nichtalphabetischen Zeichen auftreten sollten. Das vorgeschlagene Skript erfüllt die folgenden Anforderungen:

1. Es werden alle Wörter entfernt, die im Wortinnern mindestens ein *beliebiges* nichtalphabetisches Zeichen enthalten.
2. Es werden wie bisher Wörter entfernt, die mit mindestens einem Bindestrich, (falschen) Apostroph und/oder einer Ziffern beginnen oder enden.
3. Es werden schließlich sämtliche verbliebenen nichtalphabetischen Zeichen entfernt, insbesondere auch Klammern und Ziffern.
4. Neben Leerzeichen gelten auch Tabulatoren als Worttrenner. Aufeinanderfolgende Worttrenner werden durch ein Zeilentrennzeichen ersetzt.

Anforderung 1 erfasst wie bisher den »Wochenend-Einkaufszettel«, zusätzlich aber auch Abkürzungen, Formeln oder Gebilde wie »a. a. O.«, »H_2O« oder »Z******t«. Durch die Anforderung 2 werden Wortrümpfe wie »Bildungs-« oder »3fach« entfernt. Die Zahl der durch Anforderungen 1 und 2 entfernten *brauchbaren* Wörter sollte gering sein, etwa durch Schrägstrich getrennte Wörter, wie »schwarz/weiß«. Hauptsächlich dürfte es sich um Komposita handeln.

Ausdruck	Beschreibung
`[:alpha:]`	Posix-Zeichenklasse: alle alphabetischen Zeichen der Landessprache einschließlich akzentuierte Buchstaben
`[:space:]`	Posix-Zeichenklasse: Leerzeichen, Tabulator, Zeilen- und Seitentrenner
`[:digit:]`	Posix-Zeichenklasse: Ziffern
`[:punct:]`	Posix-Zeichenklasse: Interpunktionszeichen
`[[:alpha:]]`	passt auf ein beliebiges alphabetisches Zeichen
`[^[:alpha:]]`	passt auf ein beliebiges nichtalphabetisches Zeichen
`*`	Quantor: »nicht oder beliebig oft«
`\{1,\}`	Quantor: »mindestens einmal«

Tabelle 1: Einige reguläre Ausdrücke.

Im Vergleich zum bisherigen Skript `strippunct.sed` wurde auch die Reihenfolge der Aktionen geändert. Da Anforderung 3 auch die in Anforderungen 1 und 2 verwendeten Interpunktionszeichen einschließt, wurde diese Aktion nach hinten verschoben. Die bereits erwähnten Beispiele würden sonst zu früh zu »WochenendEinkaufszettel«, »aaO« usw. schrumpfen und könnten nicht mehr entfernt werden.

Außerdem werden im vorgeschlagenen Skript statt expliziter Zeichenmengen weitgehend Posix-Zeichenklassen verwendet (vgl. Tabelle 1). Grundsätzlich würde der Suchausdruck `[[:punct:][:digit:]]` Anforderung 3 genügen. Ich habe jedoch den theoretisch allgemeineren Ausdruck `[^[:alpha:][:space:]]` verwendet, der auf alles außer alphabetische Zeichen und *Whitespace* passt. Der Quantor `\{1,\}` entspricht übrigens dem `+` in erweiterten regulären Ausdrücken. Dieser Ausdruck wurde hier jedoch umschrieben, da `+` in Posix-konformen Programmen mitunter nicht zur Verfügung steht.

Das überarbeitete Skript `strippunct.sed` ist in Listing 1 zu sehen. Der Suchausdruck in Zeile 1 könnte zwar auch etwas kürzer formuliert werden, ich habe allerdings diesen Suchausdruck gewählt, da er symmetrisch ist und in dieser Form leichter verständlich sein sollte. Die Zeilen 2 bis 4 sind stark an das ursprüngliche Skript angelehnt.[2]

[2] Zeile 2 erledigt mehr als Anforderung 2. Es werden sämtliche Bindestriche, Apostrophe und Ziffern entfernt. Wichtig ist jedoch, dass nicht nur diese Zeichen entfernt werden, sondern gegebenenfalls auch unmittelbar führende und folgende Wortteile.

Listing 1: `strippunct.sed`, Version 2.0.

```
1 s/[[:alpha:]][^[:space:]]*[^[:alpha:][:space:]][^[:space:]]*[[:alpha:]]//g
2 s/[[:alpha:]]*[[:digit:]'´-]\{1,\}[[:alpha:]]*//g
3 s/[^[:alpha:][:space:]]//g
4 s/[[:space:]]\{1,\}/\n/g
```

Im Skript `prepare-wordlist.sh` wird in Zeile 6 das Programm `comm` mit einer nicht überall verfügbaren Option `-i` verwendet, um zwei Dateien ohne Beachtung von Groß- und Kleinschreibung auf nicht übereinstimmende Zeilen zu prüfen. Dieselbe Funktionalität kann auch mit `grep` erreicht werden. Dazu ist Zeile 6 des Skripts mit der aus Listing 2 zu ersetzen.

Listing 2: Ein Ersatz für `comm -i`.

```
6 | grep -Fixvf words.txt -
```

Die von `grep` zu suchenden Muster werden aus der angegebenen Datei `words.txt` gelesen, der Liste bereits geprüfter Wörter (letzte Option `f` mit folgendem Dateinamen). Es findet ein einfacher Stringvergleich statt (Option `F`), Groß- und Kleinschreibung werden ignoriert (Option `i`), die Muster müssen zeilenweise passen (Option `x`) und es werden *die* Zeilen der zu durchsuchenden Datei ausgegeben, für die *keine* übereinstimmenden Muster in `words.txt` gefunden wurden (Option `v`).

Einige zusätzliche Hinweise: a) Die Filterung mit `grep` läuft langsamer ab als mit `comm` (bei großen Dateien erheblich), da nicht ausgenutzt werden kann, dass die zu vergleichenden Dateien sortiert sind. Dafür sollte dieser Aufruf überall funktionieren. b) Mir ist aufgefallen, dass der Aufruf aus Listing 2 unter Umständen nicht korrekt arbeitet, wenn die Datei `words.txt` leer ist.[3] Daher sollte zunächst zur Datei `words.txt` ein beliebiger Eintrag hinzugefügt werden, zum Beispiel das Wort »die«. c) Außerdem ist in einer gemischten Umgebung von Unixshell und Windowskommandozeile auf die richtige (gleiche) Behandlung von Zeilenenden zu achten. Der letzte Hinweis betrifft nicht nur `grep`.

In Absprache mit Werner Lemberg stelle ich hier auch eine neue Version des Skripts `log2words.sed` vor, welche er mir schickte, während dieser Beitrag entstand. Das in Listing 3 zu sehende Skript, kommt besser mit überlangen Wörtern zurecht.

Listing 3: `log2words.sed`, Version 2.0.

[3] Zum Beispiel mit `GNU grep` 2.4.2 unter MinGW.

```
1,/Underfull/ d
/^Underfull/ d
/^\\hbox/ d
/^$/ d
\|\[\] \\T1/cmr/m/n/10 | {
  s|\[\] \\T1/cmr/m/n/10 ||
  : loop
  N
  s|\n\(..*\)$|\1|
  t loop
  s|\n||
  s|\[\]||
  s/ÿ/ß/g
}
/(\./,$ d
```

Ausblick

Obwohl Werner Lemberg auf seine Beiträge kein allzu großes Echo bekam, hoffe und vermute ich, dass die hier gestellten Fragen, die auch grundlegende Themen betreffen, für einen größeren Kreis von Interesse sind. Außerdem denke ich, dass die Pflege von Wortlisten als Grundlage von Trennmustern und Ausnahmelisten ein Langzeitprojekt sein sollte und der Mitarbeit vieler interessierter Helfer und Kontrolleure bedarf – zumindest bis die Wortlisten einen gewissen Umfang erreicht haben. Ich halte »Die TEXnische Komödie« aus diesen Gründen für ein geeignetes Diskussionsforum rund um dieses Thema und freue mich, wenn an dieser Stelle die eine oder andere der gestellten Fragen näher beleuchtet wird oder weitere Fragen aufgeworfen werden.

Zur Zeit findet die Diskussion des Projekts »Freie Wortlisten und Trennmuster für die deutsche Sprache« in der neu eingerichteten Google-Gruppe `trennmuster-opensource` statt.[2] Wer sich an der Diskussion beteiligen möchte, benötigt ein Zugangskonto bei Google, die Diskussion kann online aber auch ohne Konto verfolgt werden.[4] Der E-Mailverkehr vor der Gruppeneinrichtung ist im Dateibereich der Gruppe zu finden (Datei `mbox-vorgeschichte.zip`), außerdem auch eine kurze Projektbeschreibung, die hier fehlenden Abschnitte

[4] In naher Zukunft ist geplant auf einen anderen Softwareprojekthost umzuziehen.

meines Entwurfs, sowie neue Trennmuster, die aus Werners aktueller, erheblich erweiterter Wortliste abgeleitet wurden.

Das in Frage 7 thematisierte Problem der Wortlistenbeschaffung kann dank Werner Lemberg und Georg Verweyen als (fast) gelöst betrachtet werden. Zur Zeit verfügt das Projekt über drei Listen mit Umfängen zwischen vierhunderttausend und vier Millionen Wörtern. Eine weitere Liste mit einem repräsentativen Wortschatz aus dem Kernkorpus des Projekts »Digitales Wörterbuch der Deutschen Sprache« steht in Aussicht. Werner Lembergs Liste steht in einem öffentlich zugänglichen `Git`-Repositorium und kann mittels

```
git clone git://repo.or.cz/wortliste.git
```

bezogen werden. Genauere Angaben zu den Listen können der Projektbeschreibung entnommen werden. Die Hauptaufgabe besteht nun »lediglich« darin, diese Listen oder Teile davon auf Rechtschreibung und korrekte Trennung zu kontrollieren, damit sie die Grundlage für neue Trennmuster bilden können. Mathias Nater hat bereits einen Entwurf für eine Eingabemaske und die Datenbankanbindung fertiggestellt.[6, 7] Er ist für jede Hilfe bei der Arbeit dankbar. Interessierte sollten sich an die Gruppe wenden.[2]

Literatur

[1] Matthias Clasen: *Proposals for extensions to TEX*; `CTAN:systems/tex-extensions/clasen`.

[2] Google-Gruppe: *Trennmuster-Opensource*.

[3] Donald E. Knuth: *The TEXbook*; Bd. A von *Computers and Typesetting*; Addison-Wesley; Reading, MA, USA; 1986.

[4] Werner Lemberg: *Hyphenation Exception Log für deutsche Trennmuster*; *Die TEXnische Komödie*; 15(2), S. 28–31; 2003.

[5] Werner Lemberg: *Hyphenation Exception Log für deutsche Trennmuster*; *Die TEXnische Komödie*; 17(2), S. 24–51; 2005.

[6] Mathias Nater: *`http://www.mnn.ch/hyph/silbentrennung1.html`*.

[7] Mathias Nater: *`http://www.mnn.ch/opendehyph/index.php`*.

[8] Petr Sojka: *Notes on Compound Word Hyphenation in TEX*; *TUGboat*; 16(3), S. 290–296; 1995.