Notes on Compound Word Hyphenation in TEX

Petr Sojka
Faculty of Informatics
Masaryk University Brno
Burešova 20, 602 00 Brno
Czech Republic
Email: sojka@muni.cz

Abstract

The problems of automatic compound word and discretionary hyphenation in TEX are discussed. At present, such hyphenation points have to be marked manually in the TEX source file. Several methods for tackling with these problems are presented. The results obtained from experiments with a German word-list are discussed.

Motivation

. . . problems [with hyphenation] have more or less disappeared, and I've learnt that this is only because, nowadays, every *hyphenation in the newspaper is* manually checked *by human proof-readers.*
(Jarnefors, 1995)

In (Sojka and Ševeček, 1994) (reprinted in these Proceedings) we presented a case study of problems related to achieving quality hyphenation in TEX — especially pattern generation for flexive languages like Czech. It was shown that most issues can be handled within the frame of good old TEX, but some of them definitely not, because TEX was primarily designed not as a universal tool for the typesetting of all kinds of publications in all languages, but as one for typesetting of The Art of Computer Programming (Knuth, 1968), which is written in American English.

In this paper we continue elaborating these issues, with the emphasis on the hyphenation problems in the presence of long compound words in Germanic (and Slavic) languages.

Problems

Compounds. The main problem with automatic hyphenation was nicely expressed on the ISO-10646 electronic discussion list by Jarnefors:

> "The leading Swedish daily newspaper Dagens Nyheter had severe problems with occasional incorrect hyphenations a couple of years ago. It (and its computerised typesetting) was for a time the object of much amusement, ridicule and irritation from its readers. These problems have more or less disappeared, and I've learnt that this is only because, nowadays, *every* hyphenation in the newspaper is manually checked by human proof-readers. Because of the higher frequency of long words in Swedish compared to e.g. English or French, around a third of all lines in a typical newspaper article (with approximately 30 characters per line) end with a hyphenated word.
>
> The hyphenation problems in Swedish have to do with the high frequency of compound words (the Swedish vocabulary can't be enumerated: new compounds are easily created by anyone) and the rule that a compound word shall always be hyphenated between the constituent word parts, to ease the flow of reading."

For instance, in German and Czech there are no hyphens in compound words, you take the first word, rarely a fill-character and the second word. In some languages, compounds are built with hyphens. With this construction, it is easy to break at the end of line and to spell-check. However, in most of the languages compound word boundaries cannot be deducted from syntax only.

Dependency of hyphenation points on semantics. In some cases, even the context of the sentence is needed in order to be able to decide on the hyphenation point. Collection of examples for several languages follows:

Czech *nar/val* 'narwhal' and *na/rval* 'gathered by tearing, plucked'; *pod/robit* 'subjugate, to bring under one's domination' and *po/drobit* 'to crumble'; *o/blít* 'to vomit up' and *ob/lít* 'to pour around'

Danish *træ/kvinden* 'the wood lady' and *træk/vinden* 'the draught'; *ku/plet* 'verse' and *kup/let* 'domed'

Dutch *kwart/slagen* 'quarter turns' and *kwarts/lagen* 'quartz layers'; *go/spel* 'the game of Go' and *gos/pel* 'certain type of music'; *rots/tempel* 'rock temple' and *rot/stempel* 'damned stamp'; *dij/kramp* 'cramp in the thighs' and *dijk/ramp* 'dike catastrophe'; *ver/ste* 'farthest' and *vers/te* 'most fresh'.

English *rec/ord* and *re/cord*

German *Staub/ecken* 'dusty eck' and *Stau/becken* 'traffic jam in the valley'; *Wach/stube* 'guard room' and *Wachs/tube* 'wax tube'

Exceptions. Some hyphenation points are forbidden because of unwanted connotations the new parts of the word may have:

Czech *kni/hovna*, *sere/náda*, *tlu/močení*, *se/kunda*

English *the/rapists*, *anal/ysis*

German *Spargel/der*, *beste/hende*, *Gehörner/ven*, *bein/halten*, *Stiefel/tern*

Discretionary hyphenation points.

1. `\discretionary{`*xx*`}{`*x*`}{`*xx*`}` (in German, *x* is a consonant `f`, `l`, `m`, `n`, `p`, `r` or `t`)

   Now there will be the situation that the first word ends with a double consonant and the second word starts with the same consonant. If the second letter of the second word is a consonant, nothing changes — *Sauerstoff* + *Flasche* composes to *Sauerstoffflasche*. If the second letter of the second word is a vowel, the three consonants will be reduced to two — *Schiff* + *Fahrt* composes to *Schiffahrt*. One can find meaning-dependent discretionaries: *Bett/tuch* 'sheet' vs. *Bet/tuch* 'prayer shawl'.

2. `\discretionary{k}{k}{ck}` (German)

   This discretionary (as most of the others) has the rationale in the fact that pronunciation of `c` depends on the following letter (as in other languages). If hyphen occurs just after the letter `c`, the reading is slowed down because the reader doesn't know how to pronounce it and the eye has a long way to the beginning of the next line.

   Even here the hyphenation can depend on the word meaning: word *Druckerzeugnis* is hyphenated *Druck/erzeugnis* in case of 'printed matter' or *Druk/kerzeugnis* when speaking about a 'certificate for a printer'.[1]

3. `\discretionary{a}{}{aa}` (Dutch)

   There is another type of discretionary in which a character is deleted in case hyphenation occurs — word *omaatje* becomes *oma/tje* when hyphenated.

4. `\discretionary{é}{}{ee}` (Dutch)

   Apart from character deletion another change may occur: *cafeetje* becomes *café-tje* when hyphenated.

5. `\discretionary{l}{l}{l·l}` (Catalan)

   In Catalan the word paral·lel is broken as paral|lel, intel·ligencia as intel|ligencia. l·l is considered as one character (trigraph). With this hyphenation it changes to another two characters.

Stability of a language. Another complication is the fact that language is not fixed, non-evolving entity, but it changes, sometimes quite rapidly. New words, especially compounds, are being adopted every day. An example of an adaptation of a language to the technology — the typewriter and telegraphy in this case — may serve different spelling allowed for umlauted characters ä, ö, ü and ß in German (ae, oe, ue, ss). Some compounds are becoming percepted as base words. Thus the idea of fixing hyphenation algorithm/patterns once and forever is not a clever one.[2] A solution may consist in relatively easy generation of algorithm or patterns from the updated dictionary or description of changes.

Solutions

Compounds. It is obvious that we need to take the burden of the manual markup of compound word borders from the writer and leave it to the machine (typesetting system). The proper solution of this problem is a language module for every language, with the ability of creating new words by composition from others. This module, based on the morphology of a language, is needed, e.g., in spellchecker for that language anyway. Most probably, such language modules will become a part of the language support of operating systems in near future. Such dynamic libraries will be shared among software applications. Building such a module, however, is not a trivial task, because only some of the compounds are meaningful words.

[1] The German speaking countries are in the process of introducing new rules for hyphenation, in which `ck` is not any more allowed to be hyphenated. With the new rules, an old way which was introduced in 1902 — e.g. hyphenation of *Zuk/ker* 'sugar' might change to *Zu/cker* in the future norm.

[2] When storing document for later retypesetting with TEX we also have to save the hyphenation patterns.

Table 1: Example of discretionary hyphenation table for German

pre break text 1	post break text 2	no break text 3	left context 4	right context 5	discretionary character 6	example 7
k	k	ck	c	k	c_1	Drucker
ek	k	äck	äc	k	c_2	Bäcker
ff	f	f	f	f	c_3	Schiffahrt
ll	l	l	l	l	c_4	Rolladen
mm	m	m	m	m	c_5	Programmeister
nn	n	n	n	n	c_6	Brennessel
pp	p	p	p	p	c_7	Stoppunkt
rr	r	r	r	r	c_8	Herraum
tt	t	t	t	t	c_9	Balettheater

Looking for a temporary TEX patch that will help the current TEX users, especially those writing in Germanic and Slavic languages, the following algorithm may be used (compare with Sojka and Ševeček, 1994):

1. For a particular language a special word-list is created, which contains all word forms, but only compound word borders are marked there.
2. Hyphenation patterns from this word-list are created by `PATGEN` (Liang and Breitenlohner, 1991).
3. A special pass in TEX's paragraph breaking algorithm (for detailed description consult Knuth and Plass, 1981; Knuth, 1986a; Knuth, 1986b) is added after the first (no hyphenation trial) pass. Words are hyphenated using the compound word patterns. Then, an extra penalty `\compoundwordhyphenpenalty` is associated with these hyphenation points.
4. If `\tolerance` hasn't been met by now, further hyphenation points are added using the 'standard' patterns. These new hyphenation points have associated `\hyphenpenalty`, allowing differentiation between the two types of hyphenation points.
5. Hyphenation points 'near' the word borders (specified by `\leftdiscretionaryhyphenmin` and `\rightdiscretionaryhyphenmin` are suppressed (removed).
6. The algorithm now continues with the 'old' second and eventually the third (`\emergencystretch`) passes.
7. `\compoundwordchar` (as e.g. in Cork-coded fonts `\char'027`) is included at compound word breakpoint in order to prevent ligatures spanning over the word borders *šéflékař* 'chief doctor' versus *šéflékař* which is wrong due to the appearance of the fl ligature).

Discretionary hyphenation points. Manual insertion of discretionary points is tedious and it is usually forgotten[3], leading to typographic errors.

One solution is as follows. For every language a table of possible discretionaries is created (for a German example see Table 1).

In the word-list, the words with these discretionaries are added with the "discretionary character" inserted between "left context" and "right context". From such extended word-list the patterns are generated.

The hyphenation algorithm of TEX (for details see Knuth, 1986a, parts 38–43, sections 813–965) has to be extended. Roughly speaking

1. As a first step, "normal" hyphenation points in the word in question are found.
2. The discretionary exception table is looked up (similar to the `\hyphenation` list of exceptions). If the word is found there, a discretionary is inserted and algorithm ends, otherwise continue to step 3.
3. The discretionary table is looked up and at the hyphenation points that match "left and right context" strings (columns 4 and 5 in Table 1), the "discretionary character" (column 6) is inserted. Such a word is hyphenated once again to check whether this discretionary really applies at this position. If so, the corresponding discretionary (columns 1–3 of Table 1) is automatically inserted.

[3] How many of you, English-speaking TEX users, remember to type `eigh\discretionary{t}{t}{t}een` instead of just `eighteen`?

4. "Normal" hyphenation points, which appear 'near' to "discretionary" hyphenation points (within the 'window' specified by the values of counters `\leftdiscretionaryhyphenmin` and `\rightdiscretionaryhyphenmin`), are removed.

This approach takes the advantage of the data structure used for storing the information about the hyphenation points. The patterns are stored using the *trie* data structure (Knuth, 1973, pp. 481–505). This data structure allows effective prefix and postfix compression. Because of that, the increase in the size of the patterns is negligible, as the patterns doublets share both prefix and postfix parts in the trie.

Also, the look up time in the trie is linear with respect to the word length of hyphenated words. The time needed for looking up in the trie for the second time is thus acceptable — it is only performed sometimes — when the context of a hyphenation point is matched in the discretionary table.

The algorithm is backward compatible in the sense that if discretionary table is not present for the current language, nothing changes with respect to the standard TEX behaviour.

Exceptions. The exceptions can be reasonably handled by the patterns. Although the generation of patterns for languages with lots of exceptions may lead to the complex patterns, it is much better to regenerate the patterns with the exceptions than maintaining huge lists of exceptions and to slow down the processing considerably.

Because regenerating of patterns is not always possible, to allow enrichment of the knowledge of discretionary hyphenation points compiled into the patterns, it is wise to introduce new `\discretionaryhyphenation` for this purpose.

Experiments

For experiments we had several databases of words available. For flexive languages (Czech, German), they were based on morphology, for English it was just a list of word forms. We did our `PATGEN` experiments with German word-list generated from the full word-list by our stratified sampling technique very similar to that we described on page 63 in (Sojka and Ševeček, 1994) for Czech. We took German because the problems there are the most serious. Simple statistics show how the languages differ:

Non-uniformity of languages. In the Table 2 on page 295 there are histograms of word lengths in our databases. Although it is clear that shorter words are more frequent then the long ones, we see that in German the average word is much longer than in English and also in Czech. It is interesting to compare the total number of words. As Czech is very flexive language, from about 170 000 word stems we got more than 3 300 000 word forms. One can compare that with the best English dictionaries and spellers, which have not more than 200 000 word forms. Flexive number (ratio of total number of word form and number of word stems) for German is about 3 (we have about 120 000 word stems), but for Czech it is almost 20.

The average word length depends on the word-list chosen, but in general our results are commensurable with the result published for Welsh (Haralambous, 1993) — 9.71 characters per word, but the words like Llanfairpwllgwyngyllgogerychwryndrobwllllantysiliogogogoch were not taken into account there.

Compounds (German). In the word-list, only the compound word borders and prefixes were marked. This lead to about 150 000 positions in our German word-list. The words without any breaks of this kind were not removed. The results of `PATGEN` runs applied to this word-list are summarised in tables 3 and 4. The efficiency achieved (about 90 % breaks covered) is pretty sufficient, as 'normal' hyphenation pass follows and the error when hyphenation point is classified as 'normal' instead of 'compound' reflects only different penalty associated with this break. At the expense of pattern size we can do even better (see Table 5).

Discretionary hyphenation points. In our German word-list we had 1626 words with the `c-k` discretionary and 42 words with the discretionary hyphenation of type *x*-*x*, where *x* is a consonant — (see Table 1, (Raichle, 1995) or (DUDEN, 1991) for a list of possible discretionaries in German).

Then we created doublets of these words with these discretionaries by inserting the discretionary character (column 6) at the hyphenation position and added them to our word-list. Then we applied `PATGEN` at this new word-list. The results can be compared in tables 6 and 7. The difference in pattern size is small as expected — the size of pattern file increased by less than 0.4 kB, which makes a difference in the trie structure of about 100 bytes only.

Conclusions

We are claiming that the integration of language modules with built-in knowledge about a particular language is a must in today's top-rated systems for publishing. We have suggested extensions of hyphenation algorithms of TEX that may help

with hyphenation especially in Germanic languages with high frequency of compound words and discretionary hyphenation. Suggested extensions are possible with limited changes to TEX — The Program (Knuth, 1986a). Their implementation in any conservative successor to TEX will be rather straightforward and when the community is agreed on their usefulness they will be implemented as an independent change file. We remain undecided on the extended syntax and primitives our approach needs.

Acknowledgement

The presentation of this work has been made possible due to the support of Czech Grant Agency (grant Nr. 201/93/1269). The author would like to thank Pavel Ševeček (LOGOS, Inc.) for providing language word-lists to make the experiments and for valuable discussions on these topics. I also thank everyone who helped to improve the wording of this paper.

References

DUDEN. *Duden Band 1 — Rechtschreibung der deutschen Sprache*. Dudenverlag, 20., neu bearbeitete und erweiterte Auflage edition, 1991.

Haralambous, Yannis. "Using `PATGEN` to Create Welsh Patterns". Submitted to *TUGboat*, 1993.

Jarnefors, O. "ISO-10646 email discussion list". 1995.

Knuth, D. E. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, Reading, MA, USA, 1973.

Knuth, D. E. *The Art of Computer Programming*. Four volumes. Addison-Wesley, 1968. Seven volumes planned.

Knuth, Donald E. *The TEXbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986b.

Knuth, Donald E. *TEX: The Program*, volume B of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986a.

Knuth, Donald E. and M. F. Plass. "Breaking Paragraphs into Lines". *Software—Practice and Experience* **11**(11), 1119–1184, 1981.

Liang, Frank and P. Breitenlohner. "PATtern GENeration program for the TEX82 hyphenator". Electronic documentation of `PATGEN` program version 2.0 from UNIXTEX distribution at `ftp.cs.umb.edu`, 1991.

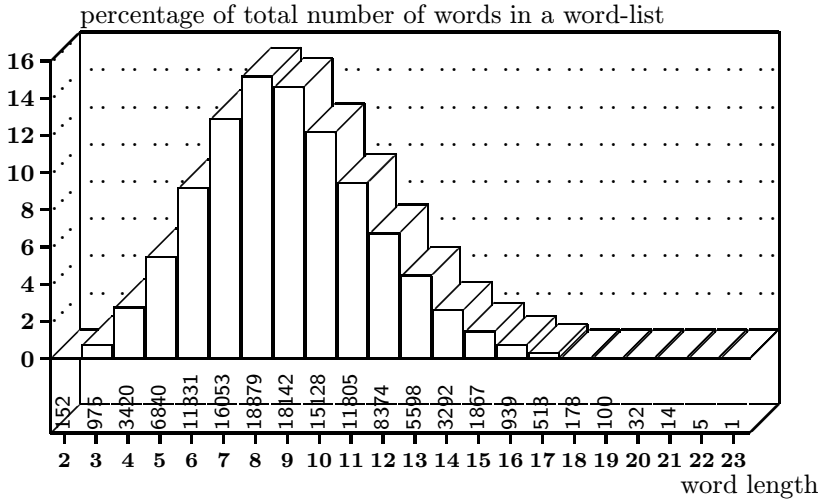
Raichle, B. "'Kurzbeschreibung – `german.sty` (Version 2.5)". 1995. Available from CTAN.

Rynning, Jan Michael. "Swedish Hyphenation for TEX". Received in electronic form from author via email `jmr@nada.kth.se`, 1991.

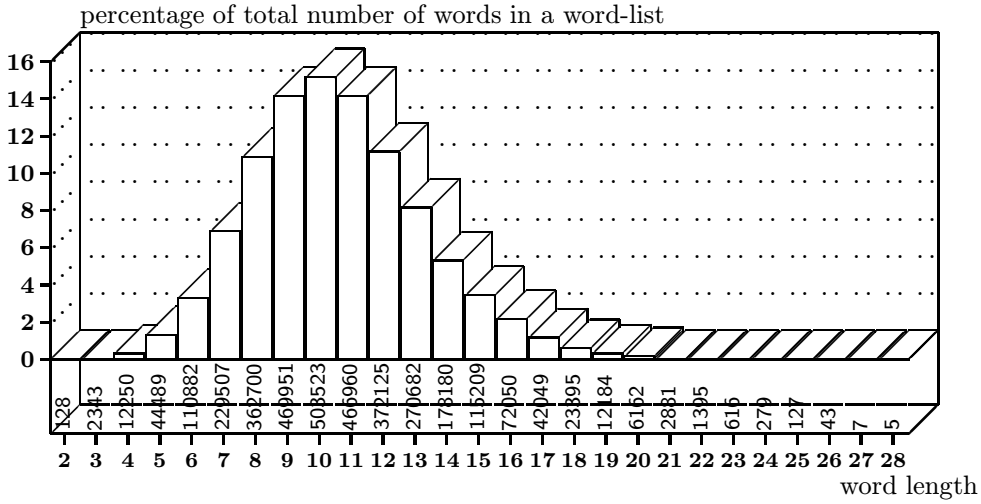
Sojka, Petr and P. Ševeček. "Hyphenation in TEX — Quo Vadis?". In *Proceedings of the 9th European TEX Conference, Gdańsk, 1994*, edited by W. Bzyl and T. Przechlewski, pages 59–68. 1994.

Table 2: Available word-lists' statistics

US English word-list (123 664 words), average word length 8.93 characters



Czech word-list (3 300 122 words), average word length 10.55 characters



German word-list (368 152 words), average word length 13.24 characters

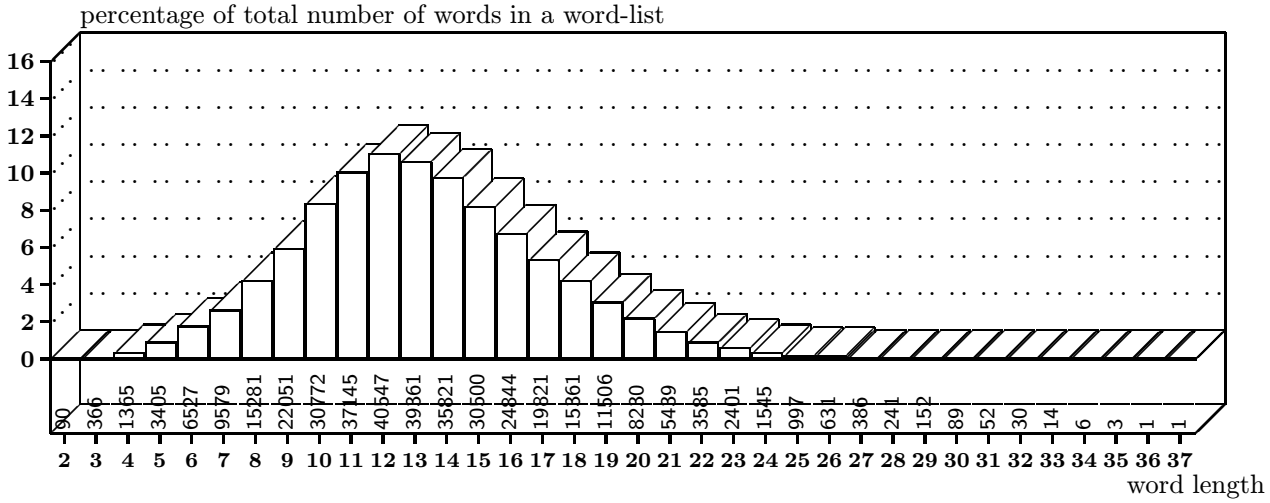


Table 3: German compound word hyphenation with pattern size optimized strategy

level	length	param	% correct	% wrong	# patterns	statistics
1	1–3	1 2 20	62.41	13.38	+ 472	good=134279
2	2–4	2 1 8	52.89	2.53	+ 712	bad=676
3	3–5	1 4 7	87.11	4.05	+2951	missed=22636
4	4–6	3 2 1	85.57	0.43	+1506	patterns size=33.6 kB

Table 4: German compound word hyphenation with different (% of correct optimised) strategy

level	length	param	% correct	% wrong	# patterns	statistics
1	1–3	1 2 20	62.41	13.38	+ 472	good=143478
2	2–4	2 1 8	52.89	2.53	+ 712	bad=698
3	3–5	1 4 3	93.06	4.23	+6612	missed=13437
4	4–6	3 2 1	91.44	0.44	+1586	patterns size=56.5 kB

Table 5: German compound word hyphenation covering even more break points

level	length	param	% correct	% wrong	# patterns	statistics
1	1–3	1 3 1	60.43	9.87	+4819	good=149502
2	1–4	1 3 2	60.24	4.21	+1714	bad=888
3	3–6	1 2 1	98.76	10.82	+1939	missed=7413
4	3–7	1 1 1	95.28	0.57	+ 353	patterns size=70.2 kB

Table 6: Standard German hyphenation patterns generation (slightly improved (size) Liang's parameters)

level	length	param	% correct	% wrong	# patterns	statistics
1	1–3	1 2 20	94.25	23.72	+ 449	good=485590
2	2–4	2 1 8	82.66	0.56	+1183	bad=48
3	3–5	1 4 7	98.59	1.08	+1737	missed=8047
4	4–6	3 2 1	98.37	0.01	+1333	patterns size=25.2 kB

Table 7: German hyphenation patterns generation with word-list with discretionary points added (the same parameters as above)

level	length	param	% correct	% wrong	# patterns	statistics
1	1–3	1 2 20	93.90	23.40	+ 456	good=492366
2	2–4	2 1 8	82.48	0.55	+1182	bad=60
3	3–5	1 4 7	98.60	1.13	+1760	missed=8155
4	4–6	3 2 1	98.37	0.01	+1388	patterns size=25.6 kB