Automatic non-standard hyphenation in OpenOffice.org

LÁSZLÓ NÉMETH
`nemeth (at) openoffice dot org`

Abstract

The hyphenation algorithm of OpenOffice.org 2.0.2 is a generalization of TeX's hyphenation algorithm that allows automatic non-standard hyphenation by competing standard and non-standard hyphenation patterns. With the suggested integration of linguistic tools for compound decomposition and word sense disambiguation, this algorithm would be able to do also more precise non-standard and standard hyphenation for several languages.

Introduction

Standard hyphenation consists of splitting a word and including a hyphen at the end of the first part of the split word (unless the word already contained a hyphen or n-dash at the break). While standard hyphenation is widely applicable, several languages also use non-standard hyphenation.

Table 1 shows examples for non-standard hyphenation: character deletions and other changes at hyphenation break points in European writing systems. Some non-standard hyphenation can be handled easily by computer, like the mandatory middle dot deletion from Catalan digraph *l·l*. But complex analysis is necessary for languages, like German,[1] Hungarian, Swedish and Norwegian to recognize hyphenation points. For instance, the Swedish word form *glassko* has three different meanings, and can be hyphenated as *glas- sko* (glass shoe), *glass- ko* (ice cream cow) and in the non-standard way, *glass- sko* (ice cream shoe).

Such non-standard hyphenation plays an important role in good typesetting. Commercial DTP programs, even word processors, support automatic non-standard hyphenation, often by licensing third party libraries. The most important free alternatives, such as Apache FOP, GNU Troff, KDE KOffice, OpenOffice.org, Scribus, and TeX and its variants, do not support automatic non-standard hyphenation. TeX has a hyphenation primitive, the `\discretionary` command. There are TeX macros in the Babel package for non-standard hyphenation, for instance, `\lgem` for Catalan *l·l*, `\ck` or `"ck` for German, `~ssz` for Hungarian, `\=` for Polish, but there is no real automatic non-standard hyphenation in TeX. Omega 2 has promising developments towards implementing sophisticated automatic non-standard hyphenation for German and other languages [4, 5].

The aim of the present project was to implement language-independent automatic non-standard hyphenation in OpenOffice.org. In this article we present our results, introduce old and new hyphenation algorithms, extension of the Hungarian hyphenation patterns and finally show the possibility of integrating compound word decomposition and word sense disambiguation to our algorithm.

Results

TeX's hyphenation is the de facto standard in the free software world, because the hyphenators of the free programs mentioned in the previous section are all based on Liang's hyphenation algorithm from TeX82 [9], and use the TeX hyphenation patterns. Thus, we have developed an extension for OpenOffice.org's ALT Linux LibHnj hyphenator to do automatic non-standard hyphenation. The result is based on a generalization of Liang's original algorithm which also allows easy integration of special linguistic tools to handle compound word decomposition and word sense disambiguation in automatic hyphenation. The Hungarian hyphenation patterns were extended with non-standard hyphenation patterns.

The improved hyphenation library (without integrated linguistic tools) is part of the OpenOffice.org 2.0.2 with the extended Hungarian hyphenation patterns. Developers can download a standalone version

[1] German orthography before the spelling reform of 1996.

Language	Example	Hyphenation	Description
Catalan	*paral·lel*	*paral- lel*	digraph *l·l* represents long (geminated) *l*
Dutch	*reëel*	*re- eel*	diaeresis and hyphenation sign syllable breaks
	omaatje	*oma- tje*	vowel lengthening with diminutive *-tje*
English	*eighteen*	*eight- teen*	suggested pretty hyphenation by D. E. Knuth [6]
German	*Zucker*	*Zuk- ker*	digraphs *ck* and *kk* represent long *k*
	Schiffahrt	*Schiff- fahrt*	triple consonants at compound word boundary
Greek	Μαΐου	Μα- ίου	diaeresis and hyphenation sign syllable breaks
Hungarian	*asszonnyal*	*asz- szony- nyal*	simplified double digraphs (long *sz* and *ny* phonemes)
Norwegian	*bussjåfør*	*buss- sjåfør*	triple consonants at compound word boundary
Polish	*kong-fu*	*kong- -fu*	repeated hyphen at line begin
Swedish	*tillåta*	*till- låta*	triple consonants at compound word boundary

Table 1: Non-standard hyphenation in European languages

```
. a l g o r i t h m .
   4l1g4
     l g o3
       1g o
            2i t h
                4h1m
  ----------------
   4 1 4 3 2 0 4 1
  a l-g o-r i t h-m
```

Figure 1: TEX hyphenation of 'algorithm'

of this library with an example executable from the Lingucomponent project home [14].

Liang's hyphenation algorithm

Franklin M. Liang's hyphenation algorithm is based on competing hyphenation patterns. The patterns can give excellent compression for a hyphenation dictionary, and using these patterns the fast hyphenator algorithm can also correctly hyphenate unknown (non-dictionary) words most of the time. Liang's work covers also the machine learning of the hyphenation patterns and exceptions by PatGen pattern generator.

The hyphenation patterns can allow and prohibit hyphenation breaks on multiple levels. Figure 1 shows the pattern matching on the word 'algorithm'. The TEX English hyphenation patterns `4l1g4`, `lgo3`, `1go`, `2ith` and `4h1m` match this word and determine its hyphenation. Only odd numbers mean hyphenation breaks. If two (or more) patterns have numbers in the same place, the highest number wins. The *al- go- rith- m* hyphenation is bad, but the last one-letter hyphenation is suppressed by TEX, so we end up with the correct *al- go- rithm*.

One of the most notable features of this pattern-based hyphenation is the human-readable format of the knowledge database, in contrast to an equivalent finite state machine or a similarly good artificial neural network. This format is good for manual checking and corrections.

Missing features

In TEX's automatic hyphenation the most wanted features are non-standard hyphenation, compound word analysis, word sense disambiguation and taboo word filtering [12, 13].[2]

Sojka's non-standard hyphenation extension

In [12] Petr Sojka suggests a non-standard hyphenation extension for Liang's algorithm. His algorithm first searches all hyphenation points of a word using Liang's algorithm, and then matches patterns from a non-standard hyphenation table at valid hyphenation points, replaces the matching pattern with a special character, and rechecks the hyphenation of the new word at this special character with Liang's algorithm. The non-standard hyphenation point will be chosen if the second hyphenation is successful. Using a *ck*→*% (k- k)* pattern data from a non-standard hyphenation table, the German word *Zucker* will be *Zu%er* after

[2] Liang's hyphenation algorithm and its compact implementation using packed trie data structure was perfect twenty-five years ago for English and for computers with less than a few MB RAM. Nowadays internationalization (handling multiple languages) is a standard in software industry and free software development. Modern personal computers have much more memory and speed to enable using additional special linguistic tools in hyphenation.

Pattern	Example	Hyphenation
`l·1l/l=l`	*paral·lel*	*paral- lel*
`e1ë/e=e`	*reëel*	*re- eel*
`a1atje./a=t,1,3`	*omaatje*	*oma- tje*
`eigh1tee/t=t,5,1`	*eighteen*	*eight- teen*
`c1k/k=k`	*Zucker*	*Zuk- ker*
`schif1f/ff=f,5,2`	*Schiffahrt*	*Schiff- fahrt*
1ΐ/=ί	Μαΐου	Μα- ίου
`s1sz/sz=sz`	*asszonnyal*	*asz- szony-*
`n1ny/ny=ny`		*nyal*
`bus1s/ss=s,3,2`	*bussjåfør*	*buss- sjåfør*
`7-/=-`	*kong-fu*	*kong- -fu*
`.til1lå/ll=l,3,2`	*tillåta*	*till- låta*

Table 2: Extended hyphenation patterns for Table 1

the replacement, and the pattern `zu%1er` permits non-standard hyphenation with *k- k* (*Zuk- ker*).

Problems

It's possible to use the pattern generator on a prepared input dictionary for Sojka's algorithm, but then we lose the human-readable format of hyphenation patterns. The biggest problem is to use competing patterns on multiple levels. That is why instead of using difficult redundant patterns with special hyphenation characters, Sojka suggests global parameters (left and right non-standard hyphenation penalties) to forbid standard hyphenations near the non-standard hyphenation points. But German, Hungarian, Norwegian and Swedish non-standard hyphenation need true competing patterns.

OpenOffice.org's extension

To keep the flexibility of Liang's algorithm, OpenOffice.org augments the original hyphenation patterns with extended patterns defining non-standard hyphenation points as subregions and replacements of the subregions. To keep the clear syntax, a non-standard hyphenation pattern is denoted as a plain hyphenation pattern and a record separated by a slash.

For example, the pattern `zuc1ker/k=k,3,2` represents the hyphenation of *Zucker*. This means the non-standard hyphenation subregion will be replaced by `k=k`, where the `=` indicates the break point with a hyphen. The subregion begins at the 3rd character, and contains 2 characters (*ck*).

```
. a s s z o n n y a l .
    s1s z/sz=sz,1,3
            n1n y/ny=ny,1,3
  -------------------
   0 1 0 0 0 1 0 0 0/sz=sz,2,3,ny=ny,6,3
  a s-s z o n-n y a l/sz=sz,2,3,ny=ny,6,3
```

Figure 2: Hyphenation of *asszonnyal*

Table 2 shows possible hyphenation patterns for Table 1. The dots in the patterns match the word boundaries. The first dot doesn't affect the character positions in the non-standard hyphenation subranges: `.zuc1ker/k=k,3,2`. Figure 2 shows the result of applying multiple non-standard pattern matching.

Rules

A single subregion must contain exactly one hyphenation point (indicated by an odd number in Liang's syntax). There may also be explicit non-breakable points (indicated by even numbers) in the subregion, and any breakable or non-breakable points out of the subregion.

A standard and a non-standard hyphenation pattern matching the same hyphenation point must not be on the same hyphenation level. For instance, `c1` and `zuc1ker/k=k,3,2` are invalid, while `c1` and `zuc3ker/k=k,3,2` are valid extended hyphenation patterns.

Unicode character encoding

Unicode is the basis for internationalization.[3] Thanks to the unambiguous start positions of the multibyte-characters, Liang's algorithm works perfectly with the UTF-8 Unicode encoding. Subregion parameters of non-standard hyphenation patterns use Unicode character (not byte) positions and lengths.

Changing hyphen

Missing or alternative hyphenation marks are handled by using underline characters instead of equal signs in our non-standard hyphenation patterns, where underline character indicates only the break point, without an implied hyphen. For example, using the underline with an explicit hyphen, `k-_k` and `k=k` are equivalent

[3] Not only for exotic writing systems. Affix-rich languages can combine different 8-bit character codes in one word. For example, *Nexøről* (*about Nexø* in Hungarian) contains special characters from Latin-1 and Latin-2 character tables.

Example	Hyphenation	Description
meggy	*meggy*	noun with long phoneme *gy* (*sour cherry*)
meggyez	*megy- gyez*	derived verb (*to do something with sour cherry*)
meggyíz	*meggy- íz*	compound (*sour cherry jam*)
meggyőz	*meg- győz*	verb prefix *meg-* + verb *gyõz* (*persuade*)
esszé	*esz- szé*	long phoneme *sz* (*essay*)
Jamesszé	*James- szé*	noun *James* + suffix *-szé* (*[to become] James*)
samesszé	*samesz- szé*	noun *samesz* + suffix *-szé* (*[to become] verger*)
esszék	*esz- szék*	noun *esszé* + plural *k* (*essays*)
vizesszék	*vizes- szék*	compound (special *chair* (*szék*) in Hungarian folklore)
rekesszék	*rekesz- szék*	verb *rekeszt* + suffix *-jék* (third-person plural *obstruct!*)
berekesszék	*berekesz- szék*	prefix *be* + verb *rekeszt* + suffix *-jék* (third-person plural *adjourn!*)
kirekesszék	*kirekesz- szék*	prefix *ki* + verb *rekeszt* + suffix *-jék* (third-person plural *exclude!*)
kerekesszék-	*kerekes- szék-*	compound (*wheel chair*)

Table 3: Hungarian hyphenation examples with ambiguous *ggy* and *ssz* patterns

patterns.[4] This notation is functionally equivalent to TeX's \discretionary command.

Extending Hungarian hyphenation patterns

The Hungarian language uses simplified forms to represent its double digraph and trigraph consonants (*sz+sz→ssz*, *dzs+dzs→ddzs*, etc.), but hyphenation undoes the simplification (*sz- sz*, *dzs- dzs*). Some ambiguity results from this non-standard hyphenation in Hungarian, caused by rich compounding and affixation, see Table 3.

Manual extension of the Huhyphn Hungarian hyphenation patterns based on Hungarian vocabularies and morphology has been accomplished, and the result contains over two thousand non-standard hyphenation patterns. For example, Figure 3 shows the competing patterns matching the word *esszé* (*essay*).

The Huhyphn distribution consists over 63 thousand hyphenation patterns generated from a 2.5 million word hyphenation dictionary by PatGen [10]. Our experience shows that with the manual extension of this database, the results are as good as the Hungarian commercial hyphenator MorphoLogic Helyesel[5]. What's more, extended Huhyphn works well on unknown words, resulting in significantly better automatic typesetting.[6]

```
. e s s z é .
     1s z é
       1z é
. e2
    s2s z
      s2z
         2é .
    s3s z é .
. e s5s z é/sz=sz,2,3
  ---------
   2 5 2 2/sz=sz,2,3
  e s-s z é/sz=sz,2,3
```

Figure 3: Non-standard hyphenation of *esszé*

Linguistic tools for better hyphenation

Pattern-based hyphenation doesn't work well on languages with an unlimited number of compound words [7]. Compound word decomposition by patterns results in an enormous number of hyphenation patterns in the Huhyphn distribution. However, within a few minutes, an expert could be able to find a dozen badly hyphenated compound words in Magyar webkorpusz, a Hungarian gigaword corpus with 21 million word forms. We need more sophisticated compound word decomposition methods, like SiSiSi [1, 7, 8]. OpenOffice.org's Hunspell spell checker also has morphological analyzer capability to decompose compound words. We suggest a simple method and formalism to integrate these tools with the pattern-based hyphen-

[4] It doesn't work in OpenOffice.org, yet!

[5] Hyphenator of Hungarian MS Office, Adobe InDesign, Adobe PageMaker and QuarkXPress.

[6] Helyesel hyphenates only known words, and it cannot handle many proper compounds, because its compound decomposition algorithm cannot decompose compounds from three or more dictionary words.

```
  g l a s s k o        g l a s s k o
. g l a s . s k o .  . g l a s s . k o .
          .7                     .7
  ---------------      ---------------
   0 0 0 0 7 0 0        0 0 0 0 0 7 0
   0 0 0 7 0 0          0 0 0 0 7 0
  g l a s-s k o        g l a s s-k o

  g l a s s k o
. g l a s . . s k o .
          .7
            .7
        s .8.9s/ss=s,1,4
  -----------------
   0 0 0 0 8 9 0 0/ss=s,4,4
   0 0 8 9 0 0/ss=s,4,2
  g l a s-s k o/ss=s,4,2
```

Figure 4: Hyphenation by decomposition

ation. Another advantage of the integration is that the external linguistic tools could also provide word sense disambiguation (for example, using part-of-speech taggers) to hyphenate the ambiguous words in hyphenation dictionaries.

Dots within patterns

Dots denote word boundaries in Liang's algorithm. Extending this formalism, let us also allow dots to denote the word boundaries within compounds. The compound word decomposition makes only a boundary annotation with dots, and we can hyphenate the decomposed word by dotted hyphenation patterns.

For instance, the Swedish word *glassko* would be *glas.sko* or *glass.ko* after compound word decomposition, and can be hyphenated with the pattern `.7` as in Figure 4.

Double dots

We denote non-standard compounding by double dots, as in *glas..sko*. This annotated word can then be hyphenated with a non-standard hyphenation pattern, such as `s.8.9s/ss=s,1,4` in our example.

The annotation is removed from the output of the hyphenation algorithm, as in the three possible annotated and hyphenated forms of *glassko* in Figure 4. With a suitable word sense disambiguation, the pattern based hyphenator is given exactly one of them. (Without word sense disambiguation, *glassko* is not annotated and hyphenated).

Conclusion

The new version of OpenOffice.org contains state-of-the-art Hungarian hyphenation, solving the problem of automatic non-standard hyphenation in a generalized way. The extended version of Liang's hyphenation algorithm is suitable for other languages. With the suggested formalism and minimal extension, the algorithm can also be integrated with sophisticated linguistic tools to handle compound word decomposition and word sense disambiguation in automatic hyphenation.

Acknowledgments

OpenOffice.org's improved hyphenation was demonstrated at the 3rd Conference on Hungarian Computational Linguistics in 2005, Szeged, Hungary, thanks to Media Education and Research Center (MOKK), University of Technology and Economics, Budapest. I would like to thank Stefan Baltzer and other developers at Sun Microsystems for testing and quick integration of the extended ALT Linux LibHnj library into OpenOffice.org, Péter Szabó for his helpful comments and Yannis Haralambous for his information about Dutch and Greek non-standard hyphenation. Also thanks to Marcin Miłkowski and the authors of various Internet sources [2, 3, 11] for the Polish and other examples. Special thanks for the editorial work of *TUGboat*.

References

[1] W. Barth and H. Nirschl. Sichere sinnentsprechende Silbentrennung fur die deutsche Sprache. In *Angewandte Informatik*, volume 4, pages 152–159, 1985.

[2] Linda Andersson et al. *Performance of Two Statistical Indexing Methods, with and without Compound-word Analysis*. `http://www.nada.kth.se/kurser/kth/2D1418/uppsatser03/LindaAndersson_compound.pdf`.

[3] Dave Fawthrop. *Hyphenation by algorithm of English/American and other languages*. `http://www.hyphenologist.co.uk/`, 2000.

[4] Yannis Haralambous. New hyphenation strategies in Omega v2. In this volume, pp. 98–103.

[5] Yannis Haralambous and Gábor Bella. Omega becomes a texteme processor. In *Actes d'EuroTEX*, pages 99–110, 2005.

[6] Donald E. Knuth. *The TEXbook*, page 314. Addison-Wesley, 1984.

[7] Gabriele Kodydek. A word analysis system for German hyphenation, full text search, and spell checking, with regard to the latest reform of German orthography. In *Text, Speech and Dialogue: Third International Workshop (TSD 2000)*, pages 39–44, 2000.

[8] Gabriele Kodydek and Martin Schönhacker. Si3trenn and Si3Silb: Using the SiSiSi word analysis system for pre-hyphenation and syllable counting in German documents.

[9] Franklin M. Liang. *Word Hy-phen-a-tion by Com-put-er*. Stanford University, 1983. `http://www.tug.org/docs/liang`.

[10] Bence Nagy. *Huhyphn — Magyar elválasztás TEX-hez, Scribushoz és OpenOffice.org-hoz*. `http://www.tipogral.hu`, 2003.

[11] Ole Michael Selberg. *Nohyphbx.tex introduction*. `http://home.c2i.net/omselberg/pub/nohyphbx_intro.htm`, 2005.

[12] Petr Sojka. Notes on compound word hyphenation in TEX. *TUGboat*, 16(3):290–296, September 1995.

[13] Petr Sojka and Pavel Ševeček. Hyphenation in TEX — Quo Vadis? *TUGboat*, 16(3):280–289, September 1995.

[14] Standalone version of ALT Linux LibHnj hyphenation library. *OpenOffice.org Lingucomponent project*. `http://lingucomponent.openoffice.org/`.