Transformation-Based Learning of Danish Stress Assignment

Peter Juel Henrichsen

Department of Computational Linguistics
Copenhagen Business School, Denmark
pjuel@id.cbs.dk

Abstract

In Danish, as in other languages, *prosody assignment* is fairly well described as a function of lexical and syntactic structure. So in principle, prosodic clue assignment should be open to machine learning techniques. This paper presents an experiment using transformation-based ML for unsupervised learning of Danish main stress assignment. The trained stress assigner is compared to the leading Danish text-to-speech system. In conclusion, ML for prosody assignment is advocated as an attractive alternative to naive word mapping as well as to labour-intensive grammar writing.

1. Introduction

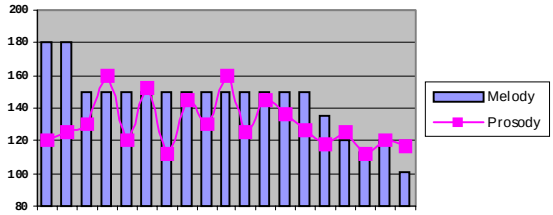
Prosody is often described as the *melody* of language. This is a somewhat misleading description. In a songline the relation between syllable and fundamental frequency is arbitrary,[1] whereas in speech it is largely predictable.

Der er noget galt i Danmark,
Dybbøl Mølle maler helt ad helved' til

This is the opening line[2] of a Danish political song from 1974. Its *melodic* shape constitutes a message in itself: sol-sol-mi-mi-mi-mi-mi-mi-mi-mi-mi-mi-mi-re-do-ti-do-la, with the insisting, even angry repetition of mi.

A spoken reproduction of John Mogensen's line must convey the anger by other means, since the *prosodic* shape has to remain within limits dictated by the distribution of main stresses (MStr) over syllables.

Figure 1: "Der er noget galt i Danmark ..."



As argued by Nina Grønnum ([1], [2]), the F0-contour of a Danish sentence is largely derivable from its MStr-distribution. As shown in schematic form in fig. 1, the fundamental frequencies (F0) of the MStr'ed syllables tend to gather on a straight, falling F0-line, with a resetting at each sentence end: cf. syllables 3, 5, 7, *RESET*, 9, 11, 15, 17 (where the remaining syllables go depends on the Danish dialect in question).

As Grønnum's model has shown descriptively satisfactory as well as computationally applicable, it has become the de facto standard prosodic model of the Danish (short-to-medium-length) sentence. As a consequence, to the text-to-speech system designer *main stress pattern* is the sole most important linguistic key to the F0-contour.

Table 1: Main stress pattern

	MStr	LC	SC	Rank	POS
der	–	3	1	#12	PRO
er	–	2	1	#4	AUX
noget	+ –	5	2	#58	PRO
galt	+	4	1	#955	ADJ
i	–	1	1	#2	PREP
Danmark	+ –	7	2	#120	PN

. **LC** = letter count; **SC** = syllable count; **POS** = part-of-speech

2. Naive Stress Assignment

As suggested by table 1, simple statistical methods can be devised for predicting the MStr pattern of a Danish sentence:

[1] Certain composers have developed melodic styles relating explicitly to the idiom of speech (e.g. Leos Janacek). Such relations are however ones of inspiration, not mimicry: tonal melodies still remain within the discrete F0s of the chromatic scale, while prosody has no such preferred absolute pitches.

[2] Litterally 'there is something wrong i Denmark; Dybbøl Mølle mills quite of/to hell to', *Something is wrong in Denmark, D.M.* [historical mill symbolizing the political situation in the country] *mills bloody poorly*.

Naive stress assignment algorithms (1)

Apply MStr to all words, except

01. short words
02. monosyllabic words
03. highly frequent words
04. function words
05. selected words {"*i*", "*er*", "*der*"...}

Algorithm 03 needs access to a word frequency list derived from a text corpus; algorithm 04 requires a corpus annotated with grammatical tags, and 05 demands a corpus marked up for MStr. All these requirement are met with the Danish PAROLE-corpus[1] ([3]), which was selected for reference corpus in the reported experiment. (All corpus figures and learning results appearing in this paper are derived from PAROLE or subparts thereof, unless otherwise stated)

Even if none of the five algorithms is perfect, some of them are remarkably efficient, if properly adjusted. With 01, 03, and 04 above, maximum precision is reached when defining 'short words' as strings of ≤3 letters, 'highly frequent words' as words of ≤#50, and 'function words' as those tagged[2] as PRO, PREP, CONJ, or AUX.

The performance of the five naive algorithms are shown in table 2 below: 16 to 17 words in 20 are tagged correctly by each algorithm (the measured precision of 03 and 05 were cross-checked on other large text corpora and found to be persistent).

Table 2: Naive tagging algorithms

Algorithm	Tagging precision
01 ('short')	82.5%
02 ('1-syllabic')	79.6%
03 ('frequent')	85.5%
04 ('function wds')	84.6%
05 ('named wds')	87.0%

[1] Prosodic tagging was done by Nina Grønnum and her students. The M-tagged part comprises about 50% of PAROLE (150.000 tokens).

[2] These tags belong to the *reduced* PAROLE set (≈40 tags), obtained from the full set (≈150 tags) by a many-to-one mapping; cf. [4].

The early Danish text-to-speech system InfoVox used an algorithm not much more sophisticated than these[3]. Performance was not quite up to the mark, but was hard to improve upon due to the lack of syntactic analysis.

i. der er noget/M galt/M (2)
ii. der er noget/U salt/M
iii. hun står/U på/U ski/M
iv. hun står/M på/U en/U ski/M

In ii. – as opposed to i. – 'noget' functions as a determiner (*there is some salt*) and appears unstressed.[4]

Examples iii. and iv. illustrate the socalled *unit stress*. In Danish, the transitive verb loses its MStr when combined with a nominal object without a determiner – compare iii. and iv. (literally: 'she stands on ski' vs. 'she stands on a ski', *she is ski-ing* vs. *she is standing on a ski*). Many other sorts of semantic units or phrases show like behaviour (viz. losing all MStrs except for one):

Lars/M	***vs.***	Lars/U von/U Trier/M	(3)
en flaske/M	***vs.***	en flaske/U mælk/M	
S/M	***vs.***	S/U A/U S/M	

Algorithms based on simple word-mapping are blind to such regularities. For this reason and for others, InfoVox was finally abandoned around 1990 and replaced by a far more ambitious text-to-speech project building on traditional PSG-analysis ([5]). This application is still being developed and is considered as the leading Danish text-to-speech application.

However, even if the PSG-based application does outperform InfoVox, it still faces the problems inherent in traditional PSG-systems. Grammar writing is an extremely laborious task. To this comes that PSG-analysis is computationally expensive and is very vulnerable to ungrammatical input (which any realistic text-to-speech system has to face). Also one may suspect that a fully-fledged *sentence* grammar is an overkill, since the actual objective is *word* tagging and *word* disambiguation only.

3. Transformation based learning

The dilemma could be escaped by turning to methods of *limited* syntactic look-out. One such method is the transformation based learning algorithm (TBL) developed by Eric Brill ([6], [7]). There are many other (mainly statistically based) algorithms available, but TBL has the advantage of being open to linguistic analysis. During a TBL learning session, a number of *rules* are produced which can be read and even revised by a linguist. Upon end of session,

[3] Personal communication with Peter Molbæk Hansen who developed the text analysis module of the Danish InfoVox over the eighties.

[4] '/M' is used for words carrying at least one MStr, (abstracting away from its actual locus). '/U' is for all other words and '/X' for unpronounced tokens, e.g. interpunction. The tag set {'M', 'U', 'X'} is referred to as MUX-tags. Only tags relevant to the discussion are shown.

the rule pile can be used for tagging unknown text (note that the rules are ordered and must be applied in a given sequence).

A TBL session requires a *master corpus* in which each token is annotated with a tag from a user-defined tagset. The applied tagging must be in near-perfect agreement with the tag definitions, which usually means that the master corpus must be marked up manually. From the master corpus, a *dummy version* is derived: same tokens, but now tagged according to a very simple initial rule (e.g. *all words are nouns*). The dummy corpus provides the point of departure while the master corpus defines the desired end state. *Learning* consists in reducing the distance between dummy and master while remebering how it was done.

A learning session has two stages: First the *lexical rule learner* is invoked. This learner produces a pile of rules for guessing the tag of an unknown token (in our case, the lexical rule learner is actually replaceable by any one of the algorithms 01-05 above).

Second, the *contextual rule learner* is invoked. As opposed to the former, this learner studies the input corpora through a *window* of seven adjacent token positions, [–3][–2][–1][0][1][2][3], quantifying over all possible instantiations of the following rule templates:

Contextual rule templates (4)

Change tag[0] from *X* to *Y* if

tag[*n*] is *Z*	(n=–1 ∨ n=1)
tag[*n*] or tag[2*n*] is *Z*	(n=–1 ∨ n=1)
tag[*n*] or tag[2*n*] or tag[3*n*] is *Z*	(n=–1 ∨ n=1)
token[*0*] is *A*	
token[*n*] is *A*	(n=–1 ∨ n=1)
token[*0*] is *A* and token[*n*] is *B*	(n=±1 ∨ n=±2)
token[*n*] is *A* and token[2*n*] is *B*	(n=–1 ∨ n=1)
token[*0*] is *A* and tag[*n*] is *Z*	(n=±1 ∨ n=±2)

where token[*P*] and tag[*P*] refers to the token in position *P* and its tag, respectively. In each iteration, a preferred rule is selected and applied in the dummy corpus, always minimising distance between dummy and master.

The session terminates when no more rules can be found (or when a preset limit is reached).

3.1. MUX tagging

TBL was developed specifically for tagging text with *grammatical* tags, but as the rule templates are quite general nothing prevents using the algorithm for other purposes[1] – such as MUX-tagging.

In the first training experiment, PAROLE in the MUX-tagged version was installed as the master corpus:

... (5)
Jeg/U har/U decideret/M ulyst/M ./X
Ka'/U du/U forstå/M det/M ?/X
Det/M tror/M jeg/U .../X
...

In this experiment, the contextual learning session spanned 181 iterations and was completed in less than one hour. Shown below are the first 10 contextual rules found.

1) U M NEXTWD af (6)
2) U M WDPREVTAG U så
3) U M WDNEXTTAG er X
4) U M NEXTWD ham
5) U M WDNEXTTAG Så U
6) U M WDAND2AFT den ,
7) U M WDPREVTAG UNS over
8) M U CURWD sætter
9) U M NEXTWD hende
10) U M WDAND2AFT med at

In paraphrase, rule 1 says "Change a U-tag to an M-tag in case the following token is 'af' ", or in paraphrase: Stress-mark any token followed by 'af' ". Rule 2 says "Stress-mark any occurrence of 'så' following an unstressed token". Out of the initial 24 rules, seven M-marks a token preceding a personal pronoun: 'ham', 'hende', 'vi', 'dem', 'jeg', 'hun', and 'du', respectively. There is also a high percentage of rules stress-marking auxiliary verbs in sentence final position, 'er', 'vil', 'være', 'var', etc. (cf. rule 3).

Admittedly, these rules hardly qualify as true linguistic generalisations (we return to this point shortly). Yet the performance of the trained tagger is surprisingly good. The lack of linguistic elegance is more than compensated by the large number of specific observations. As it seems, *quantity* does it.

3.2. Test results

Table 3: TBL- and PSG-based tagging algorithms

Sentence lengths	Correct (ML)		Correct (PSG)	
	%Wds	%Ss	%Wds	%Ss
1-5 wds	96.1	90.3	82.0	59.8
6-9 wds	94.6	66.3	77.4	19.6
10-14 wds	95.5	57.0	75.7	8.7
15-20 wds	95.9	50.0	70.1	1.8
21-29 wds	96.0	38.9	66.6	0.8
30+ wds	96.2	25.8	62.8	0.0
all	95.8	56.7	70.3	16.9

%Wds (**%Ss**) = words (sentences) MUX-tagged without errors

[1] Lanch Ramshaw & Mitch Marcus ([8]) have used the Brill-tagger for NP-recognition; Dan Hardt ([9]) describes a technique for using the tagger to learn to identify grammar errors (incorrect commas, incorrect article-noun agreement).

The deteriorating %Wds figures in the PSG-column is the fingerprint of an algorithm based on complete grammatical analysis: In case no analysis is found, the result is an arbitrary M distribution (in the tested application, *all* words end up M'ed in that situation).

3.3. Hybrid tags

In order to enhance the production of linguistically meaningful rules, two more experiments were carried out. By combining the MUX-tagged and the POS-tagged versions of the PAROLE-corpus, two new master corpora were compiled.

In the first of them, each token *T* was replaced by $T\#POS_T$, POS_T being *T*'s original PAROLE-tag.

```
...                                                    (7)
Det#PRO/M tror#VERB/M jeg#PRO/U ...#XP/X
...
```

As the contextual rule learner does not analyse tokens, this mapping only served the purpose of lexical disambiguation: Token 'Det#PRO' is definitely a *pronoun*, as opposed to 'Det' which may be a determiner (cf. (2) above, examples i. and ii.). The effect of the disambiguation was a rise in performance from 56.7% to 58.8% correctness at the sentence level (from 95.8% to 96.1% at the word level). The overall impression was that the new rules were slightly more focused – but the difference was small.

In the final experiment, the POS tags were glued to the MUX tags instead:

```
...                                                    (8)
Det/PRO#M tror/VERB#M jeg/PRO#U .../XP#X
...
```

This learning session took about eight hours and produced 417 rules, many of which clearly recognisable as rules of Danish grammar. A few examples will have to suffice:

(rule 5) U-marked determiners are changed to M-marked when followed by an M-marked noun (pronouns in this position are necessarily *demonstrative*, hence stressed).

(rule 6) U-marked prepositions are changed to M-marked when followed by U-marked pronouns (PPs are *units* (cf. discussion of (2) iii. and iv. above), hence carry a single MStr only – transferred to the preposition in case the head is a socalled *light* pronoun).

(rule 9) U-marked verbs become M-marked when followed by a full stop (verbs cannot be part of a unit clause in this position, hence retain their MStr, cf. (2)).

(rule 13) M-marked proper names become U-marked when followed by a proper name (a name in this position is likely to be part of a PN-unit, hence loses its MStr, cf. (3)).

Performance in the final experiment topped with 62.2% correctly tagged sentences, 96.4% correctly tagged words.

4. Next steps

Of course, main stress assignment is not all there is to text-to-prosody mapping. Other problems that have shown hard to solve with traditional PSG-based methods include: disambiguation of homographs, loci of pauses, and loci of F0 resetting. All of these areas could be considered for TBL treatment. Consider the case of F0 resetting for an example.

In the text-to-speech system, long sentences (15+ words) are particularly hard to segment into prosodic units. Even if loci of F0 resettings are known to be often found near major syntactic boundaries, such boundaries cannot be reliably identified as PSG-analyses of long sentences are very often defective (as discussed above). In such cases, resetting loci have to be applied quasi-randomly, often leading to incomprehensible output.

According to [10], in practice resettings may also occur in syntactically unmotivated positions. There are, however, certain positions where resettings *never* occur. So it may be that the problem of resetting is better attacked by *blocking* certain positions. As the forbidden positions are typically situated low in the syntax tree and hence does not require a full overview over the sentence, perhaps they could be nosed out by a TBL application.

In conclusion, there are reasons to believe that certain of the problems that have haunted the Danish speech synthesis group for years can soon find their solution.

5. References

[1] Thorsen, N. (alias N. Grønnum) "An Acoustical Investigation of Danish Intonation", *J. of Phonetics 6*, 1977.

[2] Grønnum, N. "Fonetik og Fonologi - Almen og Dansk", *Akademisk Forlag, 1998* [in Danish].

[3] Keson, B.-K. *Vejledning til det Danske Morfosyntaktisk Taggede PAROLE-korpus*, Det Danske Sprog- og Litteraturselskab, 1999 [in Danish].

[4] Haltrup Hansen, D. "Evaluering af NP-genkendere", *M.S. Thesis, Univ. of Copenhagen (unpubl.)*, 2000 [in Danish].

[5] Holtse, P., et al "IAAS/TFL Speech Synthesis Project, Report 3", *Copenhagen Work. Pap. in Ling. vol. 1*, 1991.

[6] Brill, E., *A Corpus-Based Approach to Language Learning*, Ph.D. thesis, Department of Computer and Information Science, Univ. of Pennsylvania, 1993. [get a free TBL distribution at **http://www.cs.jhu.edu/~brill**]

[7] Brill, E., "Some Advances in Transformation -Based Part of Speech Tagging", *Proceedings of the 12th National Conference on Artificial Intelligence AAAI-94*, 1994.

[8] Ramshaw, L., Marcus, M., "Text Chunking Using Transformation-Based Learning", *Proceedings of the 3rd ACL Workshop on Very Large Corpora*, 1995

[9] Hardt, D. "Transformation-Based Learning of Danish Grammar Correction", 2001 *(to appear)*

[10] Reinholt Petersen, N., Molbæk Hansen, P., "Fundamental Frequency Resettings, Pauses, and Syntactic Boundaries in Read-aloud Danish Prose", *Acta Linguistica Hafniensia Vol. 27*, 1994.