CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

The CBS Text-to-Speech Workbench

Juel Henrichsen, Peter

Document Version
Final published version

Publication date:
2009

License
CC BY-NC-ND

Citation for published version (APA):
Juel Henrichsen, P. (2009). *The CBS Text-to-Speech Workbench*. Institut for Internationale Sprogstudier og Vidensteknologi, CBS. Working Paper / Internationale Sprogstudier og Vidensteknologi No. 2009-1

Link to publication in CBS Research Portal

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact us (research.lib@cbs.dk) providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 12. Nov. 2026

The CBS Text-to-Speech Workbench

Peter Juel Henrichsen
Copenhagen Business School
pjh.isv@cbs.dk

Abstract

This working paper presents the CBS text-to-speech tool colloquially known as the **TtT** (Tekst-til-Tale). The tool is intended for training of university-level students, especially linguists training for a degree in speech technology, and visiting foreign students wanting to improve their spoken Danish. The TtT is operated through a simple www-based user-interface. Using the TtT requires basic skills in formal grammar-writing, but no knowledge on other aspects of artificial voice development such as phonetic-acoustic quantification, prosodic modelling, and signal generation. The paper includes a user manual.

1. Introduction

The TtT workbench (Text-til-Tale, *text-to-speech*) is a simple-to-operate web-based tool for creating Danish synthetic voices. The TtT is aimed mainly at three kinds of university-level students, the computational linguist training for a degree in speech technology, the visiting foreign student learning Danish, and the phonetician. The tool is available for non-commercial teaching and research purposes.

The tool is www-based. Apart from the obvious practical advantages, this has allowed us to obtain permission to use proprietary speech technological software thanks to the low risk of accidental exposure.

User 1: The speech technology student

University level courses in speech technology often face a pedagogical problem. Even for advanced linguistics students who have already passed courses on phonetics, formal grammar, and natural language processing, reaching the level of *professional working knowledge* within a single semester can be extremely demanding. Speech technological development – be it ASR (automatic speech recognition) or TTS (text-to-speech synthesis) – requires expertise in an unusually broad range of disciplines from psycho-acoustics and advanced statistics to formal grammar and lexical phonology. Good learning results depend heavily on a carefully balanced syllabus backed up by designated tools. A number of such tools have been developed at CBS based on seven years of teaching experience. The TtT is the prime example. By hiding away most of the technical complexity of a full-blown TTS system behind a simple user-interface, the tool allows the student to concentrate on linguistically relevant aspects of synthetic voice development. The student can thus design rewriting grammars for those aspects of

morphology, compositional phonetics, and prosody that are relevant in a production system for (rule-based) artificial speech.

User 2: The foreign student learning Danish

Danish spoken language is notoriously difficult to learn. Not only is the relationship between the spelling and the sound of words often opaque, the vast vowel inventory (greatest among all European tongues), the subtle rules of stød and stress application, compound formation, and the ubiquitous vocoid cluster reductions ("det er en" → [de:n], "der er jo" → [dAo]) all conspire to frustrate the L2 learner. The many foreign CBS students who are required to learn Danish within one or a few semesters, thus face a problem.

Der til medh: så wärdas de icke heller att talla som annat folck, uthan tryckia ordhen fram lika som the willia hosta, och synas endeles medh flitt forwendhe ordhen i strupan, for æn de komma framn	*Dansk er et nordisk sprog med plattyske udtalevaner*	*Danish is not a language, it is a throat disease*
Hemming Gadh, 1510	Jørgen Rischel	anonymous

Two specific aspects of Danish pronunciation are perceived by L2 learners as particularly hard to master, the stress assignment, and the stød[1].

Standard Danish	**Typical L2 speaker errors**	
hold k²æft	h²old k²æft	(*shut up*)
skal vi drikke k²affe	sk²al vi dr²ikke k²affe	(*should we drink coffee*)
han vil ²ikke gå hj²em	han v²il ikke g²å hj²em	(*he won't go home*)
h²un'den i h²undehu'set	h²unden i h²undh²uset	(*the dog in the dog house*)

² is **tryk** (main stress)
' is **stød** (a quick glottal contraction)

Such errors are actually easy to pinpoint and eliminate. The distribution of stress, stød, vowel shortening and related phonetic features are largely regular and can hence be expressed in ordinary grammar rules. Such rules are quite easy to state and test in TtT. Many of the regularities of Danish prosody can be described with reference to grammar rules that the advanced L2 learner has internalized already. This fact is, however, not always exploited by language teachers. Academic-level courses are typically based on traditional grammar books, which in general do not consider prosodic fine structure to belong to the realm of "real grammar". In so far as pronunciation is studied at all beyond the word level, this is often done unsystematically on a trial-and-error basis with the highly idealized phonetic annotations of a standard dictionary as the only technical reference. It is then left to the learner's musical ear to discover the regularities of the grammatically governed stress reductions, stød eliminations, vowel shortenings, etc.

[1]The stød has been presented in the literature as a quick glottal stop, but is actually better described as an instance of "creaky voice". Like the tonemes of Swedish and Norwegian (of which the stød is actually a reminiscent), it has the status of a meaning separating feature in the structuralist sense. "Bønder" (peasants) and "bønner" (beans) are thus distinguished phonetically by the stød only.

User 3: The phonetics student

The phonetics student is actually preoccupied with the very same phenomena as the L2 learner, however at another level of abstraction. Whereas the latter needs training in the practical application of the Danish rules of pronunciation, the former must learn how to *design* the very same rules. The phonetics student thus begins his TtT experience with an empty rule directory to be gradually filled up over the semester whereas the L2 learner meets TtT in a fully instantiated version ready for use.

2. Presenting the TtT

At its core, the TtT Workbench has a Danish synthetic voice[2] (the voice known as Carsten, owned by Mikro Værkstedet A/S). A web-interface provides the user of the workbench with restricted access to the speech synthesizer allowing him to define the lexical descriptions and grammar rules governing the artificial voice while hiding away all further technical details. Thus, the user has full control over the linguistic parameters, but at the same time he has no influence on the technical aspects of the processing – such as the parsing strategy, the calibration of the prosodic model, the calculation of the frequency envelope of the utterance, the timing of the phone sequence, or the encoding of the sound signal. The user may, thus, concentrate on writing lexicon entries and grammar rules and forget about everything else.

Figure 1 shows the logical parts of the speech engine. To the user of the TtT Workbench, only the two uncoloured databases at the bottom are transparent while all the other modules function as a black box.

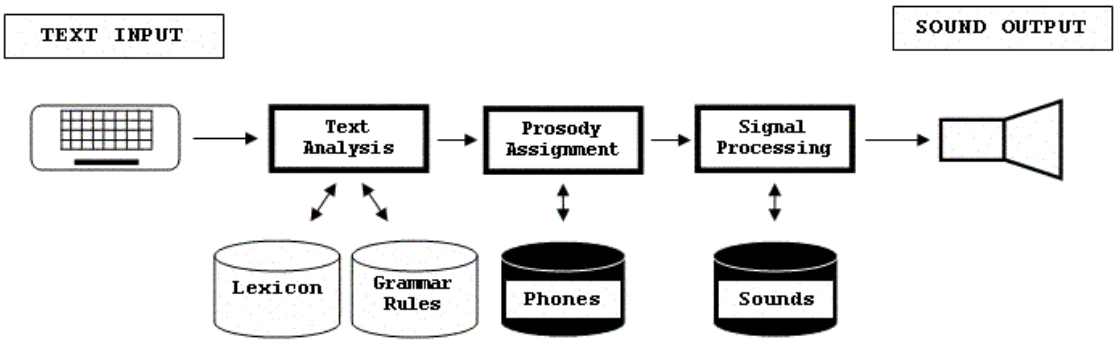

Figure 1: TtT Workbench – logical components

2.1 The Web-Interface

The web-based interface allows the user to control the speech engine using a standard browser (e.g. Mozilla or Explorer). The user must insert and upload three grammatical resources.

[2]The TtT synthetic voice was developed by the project group Dansk Syntetisk Tale, a consortium of two Danish universitites (Copenhagen and Aalborg) and a major Danish tele-communication company (TeleDanmark A/S), partly sponsored by the Danish Ministry of Research.

1. a number of **lexicon files** specifying for each entry its
    - orthographic form
    - phonetic form[3]
    - structural features (if any)
1. a file with morphological rules specifying how each input word segments into parts appearing in the lexicon files
2. a file with syntactic rule specifying how the input words relate to each other

While we expand on the roles of the various grammatical resources in the following sections, this section presents the TtT Workbench as seen from the web-user's point of view.

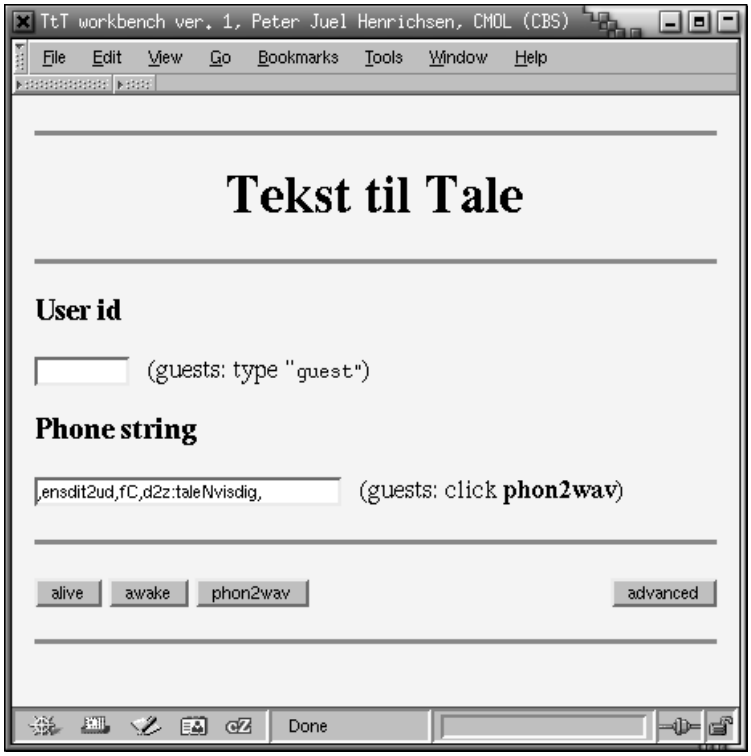

Figure 2: TtT Workbench (basic version) – browser image

Fig 2. shows the basic version. The buttons **alive** and **awake** are for testing the connection between client and server. This version allows the user to enter ready-made strings of phones such as:

User input: `,d2an!sg,aR,s2an0li,2eg0,l2Ed,`

("dansk er sandelig ikke let", *Danish is certainly not easy*)

Words may be comma separated. The commas have no influence on the acoustic rendering, but they tend to make phonetic strings more readable.

[3] The TtT Workbench phonetic inventory is based on the Danish SAMPA (**`www.phon.ucl.ac.uk/home/sampa`**), a many-to-one mapping of the IPA (International Phonetic Alphabet) on the Danish sound inventory. Since certain SAMPA symbols are inconvenient for use with regular expressions (as in the TtT server scripts) and for transfer over the Internet (used by the TtT web-interface), we use an alphanumeric SAMPA mapping. The complete TtT phone inventory can be consulted in section 7.

`2`	is "tryk" (main stress)
`!`	is "stød" (a quick glottal contraction)
`:`	is "vokalforlængelse" (vowel prolongation)
`e`	is the full vowel in e.g. "v<u>i</u>l" *(will/shall/want-to)*
`E`	is the full vowel in e.g. "vel" *(probably)*
`z`	is the full vowel in e.g. "m<u>a</u>le" *(paint)*
`0`	is schwa as in e.g. *"mal<u>e</u>" (paint)*

TtT phonetics does not include secondary stress
See the full TtT phone table in sect. 7

Table 1. TtT phone table (summary)

By pressing the button **phon2wav,** the client transmits his input to the server, which in turn returns the sound file (in **`.wav`** format) produced from the phone string. Most browsers will then allow the user to just click on the link on the answer page in order to listen to the sound file.

For pedagogical reasons, the TtT server application adopts a rather conservative style of feed-back, rejecting (with a comment) any irregular phone string. Examples of phone strings rejected:

- Strings beginning with a semivowel (e.g. `R`, `J`, or `w`) in conflict with the Danish phonotax.
- Strings with zero instances of symbol `2` (main stress); any utterance must contain a stressed syllable in order to be pronounceable.
- Strings with illegal stød. Only two stød loci are permitted, viz. immediately after a long vowel (as in "ben", [`b2e:!n`]), and immediately after a short vowel + voiced consonant (as in "bind" [`b2en!`]).

It is recommended to have students work with version 1, until they master the TtT phone inventory, before introducing them to version 2.

Figure 3: TtT Workbench (advanced version) – browser image

Each button in version 1 also appears in version 2 (advanced), accompanied by a few more: **upload** (for uploading of resource files), **dir** (for access to uploaded files), and **ort2phon** for generating phone strings from orthographic input. When the student has completed his language model by uploading valid lexicon files together with a complete and consistent grammar, the **ort2phon** button becomes active. Pushing the button will now convert an orthographic input string to a phonetic output string which in turn may be converted into a sound file using the **phon2wav** button as already explained.

4. Pronunciation patterns expressed in TtT rules

In the following sections, we present a number of grammatical case studies suggesting how problem areas of Danish pronunciation could be charted using the TtT Workbench. The cases are organized so as to progressively introduce the basic features of the TtT Workbench (Henrichsen 2001b, 2007).

4.1 The Case of Given Names and Double Names

Even if double names are perhaps not as frequent in Denmark as in Sweden or Holland, given names like Anne-Sofie and Carl-Erik are far from unusual. It is a general rule of Danish pronunciation that such double names should be rendered with one stress only, always located in the second of the connectednames. Also, all instances of stød in the first name are to be eliminated. Such rules are easily reformulated in TtT terms. Consider first a TtT lexicon file of personal names. The file id is "name".

LEXFILE: name

Niels	n2els
Henning	h2EneN
Carl	k2A:!l

Each lexical entry consists of an orthographic form and its associated phonetic rendering[4]. The morphological rules (in a separate file) make reference to the lexicon file, or files, in case there is more than one:

WORD → name

WORD → -?name dash name

The first rule permits words consisting of one personal name only (stress retained), while the second rule analyzes double names separated by dash.

The operators "-" (minus) and "?" (question mark) in the second WORD rule marks the first name component for stress reduction and stød elimination, respectively. The terminal symbol "dash" (in the 2nd WORD rule) refers to a new lexicon file with just a single entry:

LEXFILE: dash

- #

in which the symbol # means: no phonetic output.

Given an input like "Niels-Henning", this TtT grammar, thus, produces the phone string [nelsh2EneN]. "Carl-Henning" and "Niels-Carl" are rendered as [kA:lh2EneN] and [nelsk2A:!l] respectively. Notice that the stød is eliminated in the former case only which is correct.

4.2 The Case of Niels-Henning Ørsted Pedersen

You can tell from the pronunciation of a Danish name whether the speaker believes the hearer to be acquainted with the person referred to. Compare these two renderings of the name "Niels-Henning Ørsted Pedersen" (cf. phone table in sect. 7):

A. [nelsh2EneN,2xRsdED,p2e:DCs0n]
B. [nelshEneN,xRsdED,p2e:DCs0n]

Mode A is the one NHØP would probably use in a situation where he was submitting his name to, say, a social security agent. NHØP might even insert a small pause at the points of the commas to make the segmentation of his name easier to grasp. The what-is-your-name dialogue is clearly a low-redundancy situation where each name

[4] Any lexicalized phonetic string must contain at least one stress (symbol 2), since any Danish word pronounced in isolation has at least one stressed syllable. This is true even of words which are always subject to stress reduction when appearing in a phrase, such as adverbials "jo" (similar to *y'know*) and "sgu" (a mild swearword), or the infinitival particle "at" (*to*). "Jo" is, thus, lexicalized as [j2o], though it is always pronounced [jo].

component has to be transmitted without error; otherwise the whole name will fail to refer. Each word, thus, has retained its main stress (symbol 2). "Niels-Henning" is considered as a single word, since pronouncing "Niels" and "Henning" as two individual words, [n2els,h2EneN], would fail to signal the dash.

In contrast, mode B is the one used when talking to someone who has at least heard of NHØP before. In this case, the task is to simply establish a reference to an already known individual. The three words now convey a single piece of information, and this fact is prosodically encoded by the suppression of all lexical stresses except the final one.

The contrast between pronunciation modes A and B may be succinctly expressed in TtT terms. Mode A simply amounts to:

(**mode A**)

FULLNAME → $WORD_{[name]}$
FULLNAME → $WORD_{[name]}$ FULLNAME

In TtT grammars, the symbol WORD functions as the link between the syntax rules (where it plays the role of a terminal symbol) and the morphological rules (where it constitutes the top-symbol).

In the rule set above (mode A), the first FULLNAME rule permits input consisting of a single name component only (e.g. "Carl") which will then be rendered exactly as specified in the lexical phone string including the stød as well as the main stress. The mode B set differs from A at one point only, viz. the instruction "-?" to suppress the lexical stress and any stød from all non-final name components.

(**mode B**)

FULLNAME → $WORD_{[name]}$

FULLNAME → -?$WORD_{[name]}$ FULLNAME

Choosing between mode A and mode B on-the-fly may require a good deal of conversational practise to master, but the student who knows the formal rules of pronunciation, at least does not stumble around in the dark.

4.3 The Case of Compound Nouns

Consider a set of monosyllabic Danish common nouns:

{ hest,	hund,	svin,	mus,	hus,	skov,	mark,	skib }
horse,	*dog,*	*pig,*	*mouse,*	*house,*	*woods,*	*field,*	*ship*

They all share the same basic phonetic structure: a pre-vocalic consonant group followed by a full vowel and a post-vocalic consonant group. In English, most nouns of this structure form compounds by simple concatenation with little or no phonetic alteration of the components. In Danish, in contrast, compound noun formation has quite complex effects on the phonetic rendering.

In the Danish compound noun, as a general rule, all main stresses are suppressed except for the first. In some cases, this is all that happens:

mark [m2A:g] + mus [m2u:!s] = markmus [m2A:gmu:!s]

However, usually further prosodic transformations are in play. The stød of the first noun component may be eliminated, as in

skov [sg2Xw!] + svin [sv2i:!n] = skovsvin [sg2Xwsvi:!n]

(X is the full vowel in "kork").

In addition, a long vowel may be shortened:

hus [h2u:!s] + mus [m2u:!s] = husmus [h2usmu:!s]

To this comes the **fuge** (*glue particle*), a phonetic element often required to 'glue together' the components in compound nouns (and other categories as well). In modern Danish, only two kinds of fuge remain productive, the schwa-fuge and the s-fuge.

hund [h2un!] + hus [h2u:!s] = hund***e***hus [h2un0hu:!s]
skib [sg2i:!b] + hund [h2un!] = skib***s***hund [sg2ibshun!]

It is not possible, in general, to determine the fuge selection, stød elimination, and other phonetic transformations of a noun from its lexical phone string alone. In Danish, compounding is a complex interaction of lexical, morphological and prosodic rules. Advanced Danish dictionaries usually do contain some information on compounding, but it is almost impossible, even for the advanced student, to verify his pronunciation of arbitrary compound nouns on the basis of lexical look-ups alone.

Consider a toy example, a TtT grammar for compound noun formation. As seen, lexicon entries now include information on fuge selection.

LEXFILE: noun

hest	h2Esd	fuge = schwa
hund	h2un!	fuge = schwa
svin	sv2i:!n	fuge = schwa
mus	m2u:!s	fuge = schwa
hus	h2u:!s	fuge = nill
skov	sg2Xw!	fuge = nill
mark	m2A:g	fuge = nill
skib	sg2i:!b	fuge = s

LEXFILE: particle

e	0	fuge = schwa
s	s	fuge = s
#	#	fuge = nill

(# is the empty string)

Consider a first approximation of the morphological rules of stød elimination and stress reduction in compound nouns.

```
WORD      →      noun

WORD      →      ?noun  particle  –WORD
      1's fuge  is equal to  2's fuge
```

In this formulation, compound nouns retain a single main stress only, viz. that of the *initial* component. All other stresses are suppressed by the "-" operator in the second WORD rule (compare with the rules of double-names in 3.1 above, retaining the stress of the *final* component only). All stød is eliminated from non-final noun components (by the ? operator).

The condition "1's fuge is equal to 2's fuge" on the second WORD rule controls the fuge selection by feature unification. In feature conditions, symbol "1" refers to the first grammatical symbol after the arrow (in casu "?noun"), while "2" refers to the 2nd symbol ("particle"), and so on. "0" refers to the left hand symbol.[5]

The present description gets the stød and stress patterns right, even in multi-compounds with more than two component nouns:

```
skovhest      = [sg2Xw!] + [h2Esd]       = [sg2XwhEsd]
hesteskov     = [h2Esd] + [sg2Xw!]       = [h2Esd0sgXw!]

svinehundehus = [sv2i:!n] + [h2un!] + [h2u:!s]    = [sv2i:n0hun0hu:!s]
hussvinehund  = [h2u:!s] + [sv2i:!n] + [h2un!]    = [h2ussvi:n0hun!]
hundehussvin  = [h2un!] + [h2u:!s] + [sv2i:!n]    = [h2un0hussvi:!n]
```

Of course, this does not conclude the description of the prosody of Danish compounds. Many phenomena are still unexplained, e.g. why "mus" and "hus" have their vowels shortened when compounding, while "svin" and "mark" do not (compare "musesvin" [m2us0svi:!n] and "svinemus" [sv2i:n0mu:!s]). Neither have we touched on the fact that some nouns selecting an s-fuge retain their stød in compounds, contrary to the predictions of the toy grammar above, such as "stat" in "statsskov" [sd2z:!dssgXw!] and "avl" in "avlshest" [2Aw!lshEsd] (resp. *state forest* and *breeding horse*). Such problems can in fact be tackled in the TtT formalism; however, this is not the place to elaborate.

4.4 The Case of the Unit Stress

In the Danish main clause, certain types of verbal complements trigger a stress reduction on the main verb. This phenomenon is known as **enhedstryk** (*unit stress*). From a semantic point of view, its effect is akin to that of the B rule for personal names (cf. 3.2). Intuitively speaking, applying a unit stress creates a tighter, more unit-like semantic constituent to be analyzed *en bloc* rather than *en detail*. Often VPs with unit stress may be substituted by a single intransitive verb, in contrast to VPs retaining the stress on the main verb. The verb phrase "at købe varer" (*to buy goods*) exhibits the enhedstryk, and it can often be replaced by "at handle" (*to shop*); in contrast, "at købe en vare" (*to buy an item*) or "at købe varerne" (*to buy the goods*) have no enhedstryk and no similar one verb equivalent.

---

[5] The numbering convention in feature conditions is reminiscent of the formalism defined by Peter Molbæk Hansen for project Danish Synthetic Speech. The present TtT Workbench is, however, a complete rewrite and does not contain any code from the DSS project. The groundwork underlying DSS: see References, titles marked with *

To most non-native speakers of Danish, the enhedstryk is one of the prosodic puzzles that remain unsolved. However, the most important rules governing the unit stress are not difficult to express – or internalize.

Compare the pronunciation of the transitive verb "køber" (*buys*) in the following sentences (only the verb is marked for stress and stød):

Niels k²ø'ber et hus	*Niels buys a house*
Niels k²ø'ber et nyt hus	*Niels buys a new house*
Niels k²ø'ber huset	*Niels buys the-house*
Niels k²ø'ber husene	*Niels buys the-houses*
Niels køber hus	*Niels buys (a) house*
Niels køber nyt hus	*Niels buys (a) new house*
Niels køber huse	*Niels buys houses*

² is main stress
' is stød

Two conditions on the grammatical object must be fulfilled in order for the object to trigger a stress reduction (enhedstryk) on the main verb:

- indefinite form
- no overt determiner

Such enhedstryk triggering objects are called *bare*, while non-triggering objects are called *full*. Hence, enhedstryk is described in two rules:

VP → –?verb $NP_{[bare]}$
VP → verb $NP_{[full]}$

As seen, the object NPs must be marked for object type: [bare] or [full]. The marking is done in the NP rules:

$NP_{[bare]}$ → $NOUN_UNIT_{[indef]}$

$NP_{[FULL]}$ → $NOUN_{[def]}$

$NP_{[FULL]}$ → –determiner NOUN_UNIT
1's definiteness *is equal to* 2's definiteness

$NOUN_UNIT_{[indef]}$ → $NOUN_{[indef]}$

$NOUN_UNIT_{[indef]}$ → adjective+ $noun_{[indef]}$
0's definiteness *is equal to* 1's definiteness

In this section we present the TtT rules in a simplified form, see section 6 for details. Notice the condition on the second NOUN_UNIT rule, percolating the definiteness value from the adjective(s) to the NOUN_UNIT via the feature condition ("0's definiteness is equal to 1's definiteness") while requiring the noun to be in indefinite form. The grammar, thus, permits the (full) objects "det nye hus" (*the* new_{def} *house*) and

"et nyt hus" (*a new*$_{\text{indef}}$ *house*) while excluding "det nye huset" (well-formed in Swedish and Norwegian, but not in Danish)[6].

We leave it as an exercise for the reader to compile the various rules appearing in this paper into a coherent TtT grammar, upload it, and listen to well-shaped or bizarre sentences like

"Henning køber hus"

"Niels-Henning Ørsted Pedersen køber nyt hundehus"

"Carl Carl Carl køber den fine nye markmuseskibshushest"

5. Concluding Remarks

The TtT was created mainly as a tool for advanced students of speech technology and phonetics, and as such it has served well for almost eight years now. A follow-up to this working paper is in preparation reporting on our use of this (and other) tools in the class room. As experience has shown, the TtT level of grammatical abstraction makes it suitable for students who have already passed basic courses on phonetics and formal grammar, and who have some practical experience in rule writing in the Chomskian tradition.

Concerning TtT as a tool for L2 learning, we have less experience, but high hopes. We have touched on a number of rules of Danish pronunciation – rules that may be specified either in fully formal terms as in the toy grammars in section 3, or in semi-formal paraphrase as exemplified in Figure 4 below.

1. Pronunciation of non-compound names is as specified in the lexicon.
2. In compound names *N1-N2* (double names), the lexical stress is always eliminated from *N1*.
3. In compound names *N1-N2* (double names), any stød is eliminated from *N1*.
4. Pronunciation of full personal names *N1 N2 .. Nm* has two modes, call them A and B.
    - In mode A (name unknown to the hearer) all names are pronounced as specified by their morphology (cf. 1-3 above).
    - In mode B (name known to the hearer) all lexical stresses and stød are eliminated, except in the final name component which is pronounced as specified by its morphology.

Figure 4. Rules of Danish prosody.

[6] The rules governing the unit stress are actually more complex. For instance, the verb phrase "køber mange huse" (*buys many houses*) does exhibit the unit stress, while "køber nogle huse" (*buys some houses*) does not; this difference is not predicted by our example grammar, both VPs obviously containing overt determiners.

Memorizing and utilizing prosodic rules at this level of abstraction is extremely demanding on the language student (indeed this pedagogical practice has long been deprecated). On the other hand, abandoning rules altogether and relying solely on the student's intuition is not ideal either, denying him the insight that prosody *is,* to a large extent, rule governed.

An attractive compromise could be to allow the student to play around with the formal principles of pronunciation while listening to the prosodic effects of his grammatical choices. Working with formal grammar rules develops a high level of grammatical consciousness. As a supplement to traditional conversational training, CALL tools based on speech technology may, therefore, be of assistance to the advanced student.

Even if quite a few ambitious learners have trained with TtT over the years, the current version can hardly be marketed as an off-the-shelf pedagogical tool for L2 teaching. To serve that purpose, the user-interface would have to be developed graphically and accompanied by structured exercises, low-tech manuals, and even didactic principles. It is our hope that some reader of this paper would take up that challenge.

6. User's manual

6.1 Input

The TtT Workbench (tekst-til-tale, *text-to-speech*) is available in two versions, 'basic' and 'advanced'. The latter comprises all the features of the former (plus several more) while the opposite is not true. Therefore we only present the advanced version here.

There are two ways of uploading information to the advanced version of the TtT Workbench (hereafter TtT for short), an input line (IL) and an input area[7] (IA). The user can insert one line of text into the IL, and any number of lines into the IA. While there are no formal restrictions to the amount of input, TtT is not a convenient tool for building large-scale applications. IL data of up to 100 bytes and IA up to a few thousand are thus reasonable practical limits.

6.2 The buttons and their functions

Most TtT functions return a query status in the form of a string of numerals. When the string consists of 0s only, this is a sign that the query has been (i) evaluated and found to be well-formed, and (ii) processed without technical problems. Any non-0 digits refer to error types, to be reported if the technical staff at CBS are contacted.

alive

Pushing this button should get you the reply "I'm alive" as a signal that the TtT web-server is in service and your own web-connection is effective. Please operate this button before reporting any error to the TtT administrators. This function does not require a password.

awake

This button is for validating your TtT id and TtT password. The reply should be "I'm ready" together with an acknowledgement of your system id; otherwise your id is not accepted (or your TtT account has been blocked). Other possible replies include:

I'm asleep *(the TtT application is currently off-line)*
I'm lost *(you didn't provide a user id or a password)*
I'm confused *(your password is formally ill-formed)*
I'm suspicious *(your password is formally well-formed, but incorrect)*
I'm stoned *(the cpu is currently over-exposed or facing other problems)*
I'm flabbergasted *(the query is nonsensical, did you fiddle with the URL?)*

phon2wav

Given a well-formed phonetic string in IL, this function will return a sound file (in .wav-format) generated as specified in your input. The phonetic string must be

[7] In HTML-terms, the input line and input area correspond to form-elements <input type=text...> and <textarea...>, respectively.

preceded and followed by the separator symbol `,` (comma). Optionally, `,` may be inserted at any point in the phonetic string, e.g. between words, for improved legibility. This has no effect on the produced sound file. Examples of valid input:

`,en,t2iq:C,aR,m2e:C,En,en,f2Emq:C,`	(en tiøre er mere end en femøre)
`,2eN0n,k2EnC,n2ad0n,`	(ingen kender natten)
`,nels,hEneN,QRsdED,p2e:!dCs0n,`	(Niels-Henning Ørsted Pedersen)

dir

REQUIRES A VALID PASSWORD.

This function returns a list over all files currently held in your TtT domain. The filenames are clickable. On click, the contents of a file are returned into the IA for you to modify and re-submit (cf. Upload below).

ort2phon

REQUIRES A VALID PASSWORD.

The ort2phon (orthographics-to-phonetics) function calls the TtT interpreter, translating your IL input to a phonetic string as specified in your lexicon and grammar rules. This function is only available if all the following source files are found in your domain: a syntax file, a morphological file, and one or more lexicon files. Syntax rules, morphology rules, and lexicons entries must co-refer in a carefully controlled fashion as specified below.

upload

REQUIRES A VALID PASSWORD.

This function transfers the input in IA to your TtT domain. There are three valid kinds of input,

A set of morphological rules
A set of syntax rules
A set of lexical entries

The first line in IA must contain the filename. Valid filenames are

`MORPHOLOGY`
`SYNTAX`
`????``.lex` (*`????`* are four lower-case letters in a-z, i.e. not æøå)

Examples of valid filenames (also known as TtT identifiers).

```
MORPHOLOGY
SYNTAX
leds.lex
abcd.lex
voka.lex
```

Examples of invalid filenames frequently encountered: `MORFOLOGI`, `MORFOLOGY`, `morphology`, `MORPHOLOGY (ver. 2-2-2009)`, `SYNTAKS`, `navne.lex`, `Verb.lex`, `cons`, `voka.LEX`, `leds,lex`, `åben.lex`

6.3 Lexicon files

The first line of a lexicon file is occupied by the lexicon identifier (four letters plus ".lex". Each of the following lines defines a lexical entry and must conform to this line format:

```
Ostring  Pstring  Featurestructure
```

`Ostring` is a string of graphemes (usually, but not necessarily, alphabetical letters and numerals) defining an orthographic form (usually, but not necessarily, a morpheme or a full word). Space characters are not allowed in *`Ostring`* while underscores are. *`Pstring`* is a well-formed phonetic form, consisting exclusively of the phones defined in the TtT phone table (sect. 7) in a formally valid sequence. Notice that some of the symbols required in input to Ort2Phon are not used in lexicon entries. *`Featurestructure`* is a sequence of grammatical feature-value pairs (optional). Each pair has the form

```
feature . value
```

Both symbols are written with lower-case letters in a-z. Examples:

```
gnd.neutrum  num.plur  definiteness.indef
```

This particular feature-value set could come from a lexical entry like "børn" (children) coded for neuter gender and plural+indefinite form. The complete lexical entry would then be:

```
børn  b2xR!n  gnd.neutrum  num.plur  defin.ubest
```

Any number of space characters may be inserted between the lexical segments (`ostring`, `pstring`, and feature-value pairs), but nowhere else. Blank lines are permitted everywhere except as the first line in the file which must carry the filename. Example of a full lexicon file (with the TtT identifier `abcd`).

```
abcd.lex

s          s
y          2y
12         tCl!
sk         sg
du         d2u

dø         d2q:!       form.inf diatese.active
Mortensen  m2O:d0ns0n  cat.pn case.neu
A38        2y:!mC      gnd.ut num.sing taste.sour
```

6.4 The nill symbol

The hash character (`#`) serves as a general nill symbol. It may replace an *`ostring`* (to the effect that no orthographic input is read) or a *`pstring`* (no phonetic symbols produced), or even both (representing the *jump arc* known from finite state grammar theory). Warning: Use the nill symbol very cautiously, especially when engaged as *`ostring`* . Careless use may lead to endless loops.

6.5 The MORPHOLOGY file

The file named MORPHOLOGY contains the morphological rules, i.e. those describing the relations between the parts of a token (usually a word). The first line of the file must be the identifier "MORPHOLOGY", followed by lines with rules in one of these formats:

```
NONTERM  --->  TERM
   Feature-test

NONTERM  --->  TERM  NONTERM
   Feature-test
```

where

- *`NONTERM`* is a non-terminal symbol
- *`TERM`* is a terminal symbol
- *`Feature-test`* is used for placing restrictions on the applicability of the rule. Feature-tests are optional.

The rewriting arrow `--->` is written with three dashes and a greater-than sign; it must be preceded and followed by one or more space characters. Readers trained in formal grammar will recognize that MORPHOLOGY rules are finite-state, as the rewriting arrow must be followed by a terminal symbol (as opposed to the context-free rules of the SYNTAX file).

Further specifications:

- Non-terminal symbols are written with upper-case letters in A- Z (i.e. not using ÆØÅ). Examples: `POSTVOK`, `FULDVOKAL`, `KLUSIL`, `X` .
- Terminal symbols are written with four lower-case letters in a-z (not æøå). Each terminal symbol refers to the lexicon file with that identifier.
- The top symbol of the `MORPHOLOGY` grammar is `WORD` by definition. This symbol must be re-written; in other words, it must occur as the left-hand symbol of at least one rule.

6.6 Jump

The nill symbol (#) may replace a terminal symbol in any rule, resulting in a so-called "jump rule" (corresponding to a jump arc in a finite state network). Jumps do not involve lexical lookup; they can sometimes be used for smoother grammar design. Here is an example of a grammatically well-formed `MORPHOLOGY` file:

```
MORPHOLOGY

WORD      ---> prae MORE
WORD      ---> prae STILLMORE
MORE      ---> stop
STILLMORE ---> prae STILLMORE
STILLMORE ---> stop
STILLMORE ---> # MORE
```

Notice that jumps must be used cautiously. Of course, recursive rules including # will cause endless looping, but you should also be aware of mutual recursiveness among two or more rules, as in:

```
A ---> # B
B ---> # A
```

The use of feature structures and feature tests will be introduced later.

6.7 The SYNTAX file

The `SYNTAX` rule format is similar to the `MORPHOLOGY` format, but with a few important differences.

- The rewriting arrow is `-->` (dash dash greater-than)

- The permitted rule inventory is larger:

  NONTERM `--->` *TERM*
  NONTERM `--->` *TERM NONTERM*
  NONTERM `--->` *NONTERM TERM*
  NONTERM `--->` *NONTERM NONTERM*

  (again, each rule may be followed by a feature-test)

- The top symbol is `LINE`

- The only permitted terminal symbol is `WORD`

The enriched rule inventory allows for context-free SYNTAX rules as in this classical example:

```
SYNTAX

LINE --> NP VP
NP   --> N
VP   --> V NP
N    --> WORD
V    --> WORD
```

In this grammar, feature-tests would be needed to impose categorical restrictions on the N and V rewritings.

6.8 Stress reduction

Any symbol (terminal or not) can be marked for stress reduction with the prefix "-" (minus sign). Marking a terminal symbol means that the lexical stresses are removed from the phonetic form for that particular word. If a non-terminal symbol is marked, all stresses will be removed from all words dominated by that symbol in the parse tree.

```
NP --> -NAVNE NAVN
NAVNE --> NAVN
NAVNE --> NAVN NAVNE
```

These SYNTAX rules removes all stresses from compound proper names except the final one (cf. 4.2). Stress reduction is also useful in MORPHOLOGY rules:

```
KOMPOSITUM ---> subs -FLERE
FLERE ---> subs FLERE
FLERE ---> subs
```

These morphological rules (notice the three-dash arrows) rewrite a compound noun as a string of stems, all but the initial one being marked for stress reduction. With a suitable lexicon file subs, a word like "plastiktandbørstehår" (*plastic-tooth-brush-hair*) will be pronounced with the correct stress pattern.

6.9 Stød elimination

Elimination of stød is similar to stress reduction, using ? instead of −. Any combination of ? and − is allowed. Examples of MORPHOLOGY and SYNTAX rules:

```
COMPOUND ---> ?noun -noun       (as in "husmus")

PP       -->  -?P NP            (as in "0på den ene side")
VGROUP   -->  -?VAUX PTC        (as in "0[skulle have været] spist")
```

6.10 Feature tests

Any rewriting rule in SYNTAX and MORPHOLOGY may be followed by an optional feature test.

```
LHS0 arrow RHS1
   Feature-test

LHS0 arrow RHS1 RHS2
   Feature-test
```

where *LHS0* is the left-hand side symbol to be rewritten, *RHSn* the right-hand symbols, and *arrow* either --> or ---> . Notice that the left-hand side symbol is indexed with 0, and the right-hand symbols with 1 or 2. These indices are used for reference in the feature-test. The feature-test consists of any number of conditions written one at a line. This rewriting rule thus has 4 conditions.

```
VP --> VAUX VINF
  1.type < aux
  1.form < finit
  2.valens < intrans
  2.form < infinit
```

The infix operator < means "has the value", and *ix.fea* refers to feature *fea* of symbol *ix*. Only lower-case letters in a-z are allowed in features and values. The first condition 1.type < aux thus requires that the feature type of the right-hand symbol VAUX has the value aux. In case type is uninstantiated (has no value set) by the time the rule is called, type will be set to aux as a side-effect of the evaluation (this value cannot be changed again). If type was already instantiated, the condition is fulfilled only if its value is equal to aux.

The other infix operator used in feature tests is <> (less-than greater-than). It means "has the same value as". Its use is best illustrated by an example.

```
NP --> DET CN
  1.gender <> 2.gender
  0.gender <> 1.gender
  1.num    <> 2.num
  0.num    <> 1.num
```

This NP rule is conditioned on the congruence of the determiner and the noun wrt. gender and number. "Et hus" ($a_{NEU+SNG}$ $house_{NEU+SNG}$) og "en cykel" (a_{UT+SNG} $bike_{UT+SNG}$) are wellformed noun phrases, hence accepted by the grammar, while "et cykel" ($a_{\mathbf{NEU}+SNG}$ $bike_{\mathbf{UT}+SNG}$) and "en cykler" ($a_{UT+\mathbf{SNG}}$ $bikes_{UT+\mathbf{PLUR}}$) are rejected. If the CN is underspecified for gender, as is the case for certain Danish nouns like "fond", (*foundation*), the gender feature of the NP will be properly instantiated as a side-effect of the <> application in the condition 0.gender <> 1.gender.

A feature test is fulfilled if and only if each of its conditions is met. If the test fails, the rewriting rule does not apply.

```
VP --> -V OBJECT
  2.unitstress < yes
VP --> V OBJECT
  2.unitstress < no

OBJECT --> DET N
  0.unitstress < no
OBJECT --> N
  1.defness < yes
  0.unitstress < no
OBJECT --> N
  1.defness < indef
  0.unitstress < yes
```

This example shows how the 0-index can be used for coding a phrase according to a syntactic property. As discussed in sect. 4.4, the application of the Danish unit stress in verb phrases is (partly) a function of the topology of the grammatical object rather than properties of the participating lexemes. The decision concerning stress reduction in the VP rule thus depends on information which none of the lexemes can provide. The suggested solution uses a feature `unitstress` defined exclusively in the `SYNTAX` file to mediate between the `VP` and `OBJECT` constituents.

The underscore symbol is used for underspecified values in the lexicon. This can often be useful for lexemes with underspecified feature values.

```
trist   tr2isd  gnd.ut   num.sng   def.no        (adj. sad)
trist   tr2isd  gnd.neu  num.sng   def.no

blåt    bl2Cd   gnd.neu  num.sng   def.no        (adj. blue)
blå     bl2c:!  gnd.ut   num.sng   def.no
blå     bl2c:!  gnd.neu  num.plur  def.no
blå     bl2c:!  gnd.ut   num.plur  def.no
blå     bl2c:!  gnd.neu  num.sng   def.yes
blå     bl2c:!  gnd.ut   num.sng   def.yes
blå     bl2c:!  gnd.neu  num.plur  def.yes
blå     bl2c:!  gnd.ut   num.plur  def.yes
```

Using the underscore, these rather redundant entries can be replaced by:

```
trist  tr2isd   gnd._    num.sng   def.no

blåt   bl2Cd    gnd.neu  num.sng   def.no
blå    bl2c:!   gnd.neu  num.plur  def.no
blå    bl2c:!   gnd.ut   num._     def.no
blå    bl2c:!   gnd._    num._     def.yes
```

6.11 File deletion

Use the IL for deleting unwanted files from your TtT directory. If you wish to abolish your `SYNTAX` file, simply type "SYNTAX" (followed by nothing) and push **Upload**. Similarly for `MORPHOLOGY` and lexicon files.

6.12 Output

Output from TtT is presented in the form of an html-page containing information on the query processing (normal/abnormal), comments and/or error messages, and most importantly, the outcome of the grammatical analysis including a variety of morphological and syntactic details. Many parts of the parsing information are clickable, such as the individual morphological analyses. On click, these results will be transferred to the TtT main page and inserted in the IL for easy testing with the **ort2phon** and **phon2wav** buttons.

In phonetic strings, areas which have been marked for stress reduction as an effect of the parsing are marked with red color while areas with stød elimination are shown in *italic font*.

7. TtT phone definitions

Phonetic symbols permitted in TtT input and lexical entries. The TtT phone inventory is inspired by the Danish SAMPA alphabet, with small modifications, cf. http://www.phon.ucl.ac.uk/home/sampa/. TtT phone symbols are all alphabetical, while diacritics (stød, vowel prolongation, etc) are not. Digits are reserved for current and future degrees of stress, e.g. [2] for lexical main stress.

The TtT vowel inventory includes only one symbol for the three vowels in e.g. "Sommerslot" [s2CmCslCd]. The phonological distinction between the full vowel in "nok" [n2Cg] and the vowel found in morphological endings (sometimes called schwa-2) as in "baner" [b2z:nC], "længere" [l2ENCC] is thus collapsed in TtT phonetics. In our experience, students and experts alike find it very hard to distinguish consistently between the two, e.g. in words of foreign origin: "meter", "motor", "cancer", "censor", "Alexander", "Hektor".

[z] is almost always prolonged, notable exceptions being words using the letter as a morpheme: "A4-ark", "ABC-bog", "A38" . [a] and [C] are rarely prolonged, except in interjections like "ja", "så". [0] (*schwa*) is never prolonged or stressed, neither does it licence stød.

Table 2. Vowel symbols in TtT with nearest IPA equivalents (PRAAT encoding)

Vowel		**Phonetic variants**				**As in**
TtT	IPA	Short	Long	Stød	Stress	
i	i	+	+	+	+	vi, hvile, hvil
e	e	+	+	+	+	det, dele, del
E	\ef	+	+	+	+	gæs, næse, hæs (adj.)
z	\ae	–	+	+	+	vane, las
a	a	+	–	+	+	vande, vand
A	\as	+	+	+	+	bak, bark, barm
y	y	+	+	+	+	tys, nyse, lys
q	\yc	+	+	+	+	øl, øse, pøl
Q	\o/	+	+	+	+	søn, høne, løn
x	\oe	+	+	+	+	grøn-(kål), trøje, grøn
u	u	+	+	+	+	guld, gule, fuld
o	o	+	+	+	+	jo (adv.), tone, rod
c	\ct	+	+	+	+	kul, kåbe, kål
C	\vt	+	–	+	+	som, tom
X	\ab	+	+	+	+	vor, båre, vår
0	\ic	+	–	–	–	e.g. infinitival -e ("schwa")

Table 3. Consonant symbols in TtT with nearest IPA equivalents (PRAAT encoding)

Consonant		**Syllabic positions and stød-licencing**				**As in**
TtT	IPA	Inital	Other prevoc.	Post-vocalic	Stød base	
b	b	+	+	+	–	by, op
d	d	+	+	+	–	dø, at
g	g	+	+	+	–	gå, tak
v	v	+	+	+	+	ve, ulv
f	f	+	–	+	–	fe, hof
s	s	+	–	+	–	sø, es
S	\sh	+	–	+	–	sjat, tusch
h	h	+	–	–	(–)	hø
m	m	+	+	+	+	må, om
n	n	+	+	+	+	nå, an
l	l	+	+	+	+	lå, øl
j	j	+	+	–	(–)	ja
r	\rc	+	+	–	(–)	rå
p	p	+	–	–	(–)	på
t	t	+	–	–	(–)	tå
k	k	+	–	–	(–)	ko
J	j	–	+	+	+	hej, haj
R	\ri	–	+	+	+	Per, ser
D	\dh	–	+	+	+	hid, tid
N	\ng	–	+	+	+	gong, gang
w	w	–	+	+	+	hav, av!

7.1 Further conventions

A "full vowel" is any vowel with the exception of schwa (symbol [0]). As mentioned, the TtT conventions thus does not support the "schwa-2" analysis of the vowel [C] in words like "venner" [v2EnC], "længere" [l2ENCC] proposed by some phonologists.

TtT uses the colon [:] for vowel prolongation ("penne" [p2En0] vs. "pæne" [p2E:n0]). Any full vowel may be prolonged (see table 2).

Main stress is marked with [2], prefixed to a full vowel (e.g.[2i]). Any full vowel can carry a main stress.

Stød is annotated with the exclamation mark [!]. It may occur in one of two different positions, (i) just after a prolonged vowel and (ii) just after one of these consonants [mnlJRwDN] if preceded by a short, full vowel (see table 3). Examples:

"vej"	[v2AJ!]	but not	[v2A!J]
"hegn"	[h2AJ!n]	but not	[h2A!Jn], [h2AJn!]
"pæn"	[p2E:!n]	but not	[p2E:n!]
"pen"	[p2En!]	but not	[p2E!n]

References

* Henrichsen, P. J. 2001a. Transformation-based learning of Danish stress assignment. Proceedings of Eurospeech-01.

* Henrichsen, P. J. 2001b. Kompendium i Taleteknologi – med særlig vægt på syntese af dansk talesprog. Copenhagen Business School Press

* Henrichsen, P. J. 2007. Tekster til Taleteknologi – med særlig vægt på syntese af dansk talesprog. Copenhagen Business School Press

* Holtse, P. et al. 1991. IAAS/TFL Speech Synthesis Project. report 3: Cph. Work. Pap. in Ling. 1. 146-168. (reprinted in Henrichsen 2001b, 2007).

* Holtse, P. et al. 1993. Syntetisk tale – en introduktion; Teleteknik 93/1 (reprinted in Henrichsen 2001b, 2007).

* Petersen, N. R. & Hansen, P. M. 1994. Fundamental frequency resettings, pauses, and syntactic boundaries in read-aloud Danish prose. Acta Linguistica Hafniensia. 383-400.

* Rischel, J. 1996. Frembringelse af syntetisk dansk tale: Problemområder. Univ. of Copenhagen (reprinted in Henrichsen 2001b, 2007).

* Thorsen, N. (alias Grønnum, N.) 1978. An acoustical investigation of Danish intonation. J. of Phon. 6. 151-175.

Grønnum, N. (alias Thorsen, N.) 2005. Fonetik og Fonologi, almen og dansk; Copenhagen: Akademisk Forlag

*Papers marked with * concern the groundwork of the Danish synthetic voice, developed in the project Dansk Syntetisk Tale.*