HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Deliverable D2.2

Description of the final version of the new front-end

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 287678.

SEVENTH FRAMEWORK PROGRAMME

Participant no.	Participant organisation name	Part. short name	Country
1 (Coordinator)	University of Edinburgh	UEDIN	UK
2	Aalto University	AALTO	Finland
3	University of Helsinki	UH	Finland
4	Universidad Politécnica de Madrid	UPM	Spain
5	Technical University of Cluj-Napoca	UTCN	Romania

Project reference number	FP7-287678
Proposal acronym	SIMPLE4ALL
Status and Version	Complete, proofread, ready for delivery: version 3
Deliverable title	Description of the final version of the new front-end
Nature of the Deliverable	Report (R)
Dissemination Level	Public (PU)
This document is available from	`http://simple4all.org/publications`
WP contributing to the deliverable	WP2
WP / Task responsible	WP2 / Task T2.2
Editor	Martti Vainio (UH)
Editor address	martti.vainio@helsinki.fi
Author(s), in alphabetical order	Stig-Arne Grönroos, Peter Smit, Antti Suni, Martti Vainio, Oliver Watts
EC Project Officer	Pierre Paul Sondag

Abstract

On of the main goals of the SIMPLE4ALL is to replace the traditional approach to text-to-speech front-end text processing with fully data-driven approaches based on machine learning and to develop unsupervised language-independent methods for linguistic representation estimation. This report describes the final version of the linguistic front-end of the SIMPLE4ALL system. The system for handling non-standard words, such as abbreviation, numbers and acronyms, the system for building linguistic representations in a unsupervised fashion, and an automatic prosody modelling system based on word prominences are described in Deliverable 2.1. This deliverable describes the additional work done towards finalising the linguistic front-end.

Contents

1 Introduction

Building a statistical text-to-speech synthesiser relies on large amounts of textual data and pre-recorded speech signals. Moreover, the speech signals have to be labeled according to their written form. This is usually very time consuming, and relies on manual effort from experts; it is, therefore, expensive and does not scale well to building systems for large numbers of languages. However, the hypothesis that SIMPLE4ALL is testing is that all of the methods for preparing data for TTS voice building can be automated; modern machine learning techniques that are fully data-driven can replace the expensive human labor in the process.

Replacing the traditional linguistic front-end of TTS with a fully data-driven approach based on machine learning is one of the main goals of SIMPLE4ALL . In general, this calls for a set of language-independent methods for linguistic representation estimation from data, which has itself possibly been acquired in a semi-automatic fashion from non-standard sources and/or provided by non-expert users.

The project aims to demonstrate the construction of complete speech synthesis systems starting only from speech and text, employing our novel methods for the front end in conjunction with a conventional state-clustered context-dependent HMM waveform generation module.

This report describes the final version of the linguistic front-end of the SIMPLE4ALL system. The system for handling non-standard words, such as abbreviations, numbers and acronyms, the system for building linguistic representations in a unsupervised fashion, and an automatic prosody modelling system based on word prominences are described in Deliverable 2.1. This deliverable describes the additional work done towards finalising the linguistic front-end. The new features implemented to the linguistic front-end include improved methods for both lexical decomposition into morph-like units and improved language modelling, as well as prosodic labeling and automatic syllabification. The overall frame-work and its evaluation is described in Section 1, followed by descriptions on automatic syllabification, morphological decomposition, and automatic prosodic tagging using wavelet decomposition.

The current version of the system documentation is also attached as an Appendix to this Deliverable.

2 Description of the Python framework

2.1 System description

Whilst the software implementation has been cleaned up and rationalised considerably, the overall framework remains very similar to that already presented in Section 3 of D2.1, and that description will not be repeated in full here. To briefly recap, however, a front-end built using the Python framework consists of a sequence of utterance processors, each of which accepts and enriches an XML representation of an utterance. The framework is designed to be flexible so that the exact number of processors and their roles can be reconfigured by the developer/user. As an example, a voice built using a typical naive configuration (such as the voices whose evaluation is discussed in Section 2.2) will use the following types of processors:

Tokenisation Processors which: tokenise text based on regular expressions querying Unicode character classes, classify tokens as word/space/punctuation etc., supply ASCII-safe representations of input text

Alignment Processors which, during training, extract acoustic features and create a time-alignment of text units with those features

Vector Space Modelling Processors which in training construct vector space models from text data, and, at voice training and synthesis time, tag textual units with VSM features

Pause prediction and phrasing Processors which in training find pauses detected during alignment and create a predictor of those pauses, and which add phrase structure to utterance based on detected or predicted pauses

Rich contexts and label generation Processors which extract rich context label and question files suitable for use in acoustic model training or at synthesis time

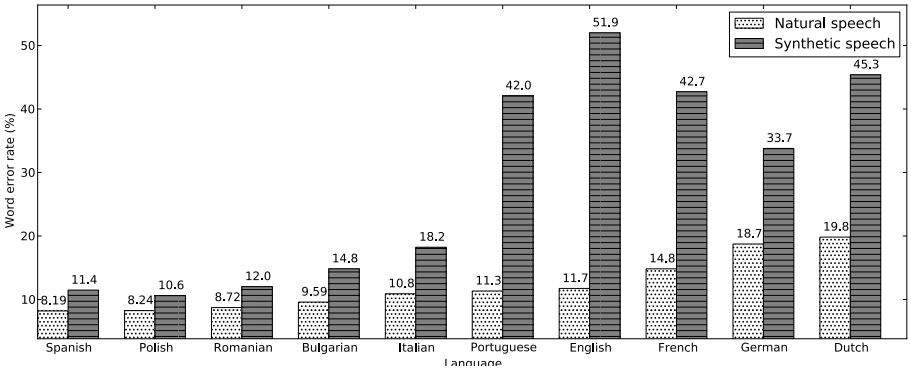

Figure 2.2a: *Absolute WERs for the full Tundra evaluation.*

The most recent documentation for end users which is distributed with the released tools, probably constitutes the most useful sort of system description, and interested readers are referred to the downloadable code (see D6.5) for details.

2.2 Evaluation

22 synthetic voices (in 21 different languages) have now been built using the methods described above, and synthetic speech from 16 of them has subjectively evaluated. The building of 14 of these has been published in [1], which is appended in full to the current document. Briefly, that paper describes systems for Bulgarian, Dutch, English, French, German, Italian, Polish, Portuguese, Romanian, Spanish, Russian, Hungarian, Danish and Finnish. Each of these systems was built from around 1 hour of 'found' data, gathered with minimal supervision, and using the naive system configuration. Systems in the first 10 languages listed above were subjectively evaluated for intelligibility using a crowdsourcing service; systems in the final four languages listed were not evaluated because it was not possible to recruit native listeners using the chosen service. As we do not have access to conventional systems in most of these languages, and as we do not have Semantically Unpredictable Sentences (SUS) for them, we set aside some chapters of the found material for use as a test set, and compared the systems' synthetic speech with natural speech. Full results for all languages were not available for publication by time of submission for [1], and so an expanded graphical presentation of those results is presented here: Figure 2.2a shows the absolute word error rates of listeners' transcriptions of synthetic and natural speech for all 10 languages where systems were evaluated.

Another way to visualise these results is as the ratio of the natural to the synthetic WER: these ratios are shown in Figure 2.2b. These figures answer the question: *How many times less intelligible than natural speech is synthetic speech?* For comparison, the corresponding ratio for the 2011 Blizzard Festival unit selection benchmark system (albeit on SUS) was 1.47. It therefore appears that there is a group of languages where performance is reasonable (Polish, Romanian, Bulgarian, Italian, and perhaps German), and a second group where performance is markedly worse (Dutch, French, Portuguese and English).

However, the results of this evaluation must be treated with caution, because as well as language-specific differences, we have obviously not been able to control for other differences between the data in the different languages, such as speaker characteristics, recording conditions, and the inherent difficulty of transcribing material from the various sources used. It is our working hypothesis that most of the differences in intelligibility (in terms of ratio WER) between the different languages could be accounted for by language- and script-specific factors, such as grapheme-to-phoneme complexity. On-going work – which may partly be carried out as part of a postgraduate Masters dissertation project at UEDIN – will focus on further experiments in order to isolate the language-specific factors from other factors such as speaker and recording quality (by, for example, building voices on multiple datasets and speakers per language).

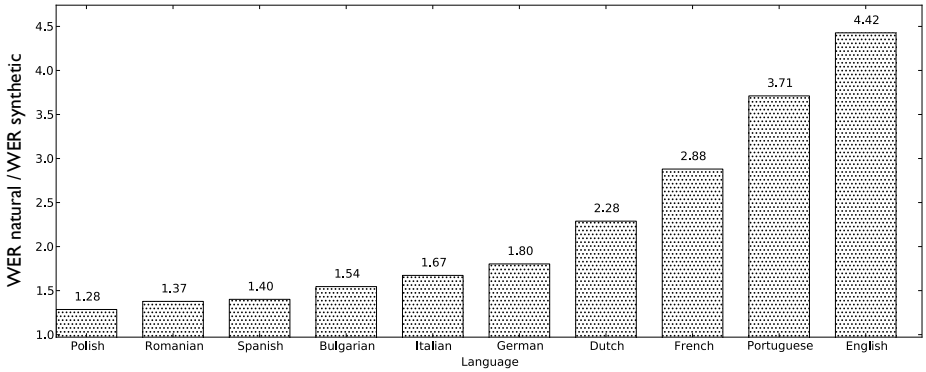

Figure 2.2b: *Ratio WERs for the full Tundra evaluation.*

As mentioned in D2.1 (pp. 31–2), voices have been built previously using our tools for the 7 languages of the IIIT-H databases: Bengali, Hindi, Kannada, Tamil, Malayalam, Marathi and Telugu. The 2013 Blizzard Challenge presented an opportunity to rigorously evaluate systems in the first four of these languages: full details of the Simple4All entry are given in [2] which is appended to the current document for convenience, and we just briefly summarise it here. The initial STRAIGHT-based acoustic models that we built on this relatively noisy data produced apparently poor quality synthetic speech. Therefore for the Challenge, new acoustic models were built for the four relevant languages, using the denoising techniques we first used successfully on the Tundra data, some manual selection to remove the most reverberant section of the Bengali data, and GlottHMM-derived acoustic features.

Performance of our techniques relative to the other entries (which generally use traditional supervised and resource-intensive approaches) for the Indian language tasks is most encouraging. For the speaker similarity and naturalness sections of the evaluation for all 4 languages, our system tends to score somewhere in the middle of all TTS systems. The intelligibility results published for Hindi and Kannada follow a similar pattern. In the Hindi test, 4 TTS systems achieved lower (i.e., better) WERs than ours, 1 was worse, and 1 scored within 1% WER of our system; in the paid listener subset, our system achieves precisely the middle rank in the Hindi intelligibility results. In both listener group sections of the Kannada intelligibility test, our system also achieves precisely the middle rank. No intelligibility results were available for Tamil or Bengali.

We regard the middling performance of our system in this evaluation of Indian language systems as a success, given that our system makes no use of expert script knowledge, while other systems made use of at least the phonetic annotation distributed for the challenge. This is the first formal evaluation of our letter-based front-end as applied to a non-alphabetic script: we regard its reasonable performance on these four alphasyllabic scripts as a validation of the unsupervised approach applied in a key target domain: under-resourced languages.

Besides the Indian language tasks, the Challenge included two English language tasks, for one of which we prepared an entry. The poor performance of our system on this task was no surprise in light of results such as those shown in Figure 2.2a, and of the high level of expertise that has been accumulated in English TTS where there is no self-imposed limit on the amount of target-language expertise that can be used in a system. We chose, however, to submit an English system built on exactly the same principles as for other languages we have tackled. It is possible that the individual listener responses obtained for English may still be useful as a form of user feedback; for example, they could be useful for developing lightly supervised and unsupervised lexicon induction techniques by guiding us towards problematic words. Because the Blizzard stimuli will be released after the challenge, it is possible to evaluate and later improved system by re-running the evaluation locally on a smaller scale, using a subset of selected benchmark systems from the challenge which allow new results for improved systems to be placed among existing Blizzard results. Having our own baseline among the original results is useful for sanity-checking when projecting results for new systems into the space of existing results.

Finally, and separately from the Tundra and Blizzard voice building exercises, our techniques were used to build

a voice from web-scraped Malay data, by a visitor at UEDIN , Lau Chee Yong (Universiti Teknologi Malaysia). These results have not yet been published, but some initial findings will be briefly summarised here. Four voices were evaluated to explore 2 factors in voice-building: the use of automatically-harvested data (see Section 6.1 of D1.6) versus purpose-recorded TTS data, and the use of wholly naive letter-based synthesis versus the partial grapheme-phoneme rules obtained with active learning techniques described in D4.1). A listening test showed that using active learning to manually disambiguate pronunciations of a single letter can greatly improve intelligibility. Word error rates of 28 listeners' transcriptions of Semantically Unpredictable Sentences were reduced from 41.67% to 15.69% in the case of voices built on studio-recorded data, and from 54.65% to 40.42% for voices built on found data. These initial results suggest that the active learning interface already developed for this experiment was useful, and so should be integrated more closely with the front-end code.

3 Syllabification

Syllable is an important level in phonological hierarchy, carrying information on lexical stress, accent and rhythmic properties of speech. In supervised TTS systems, syllable is always included in the modelling and it is certainly desirable to attempt including it in the SIMPLE4ALL framework, too. Universally, syllable is composed of onset, nucleus and coda, where nucleus consists of one or more vowels, and onset and coda of zero or more consonants. Thus, the problem of unsupervised text-based syllabification of languages with alphabetic script can roughly be split to four separate tasks;

1. Identifying which letters correspond to vowels and consonants.
2. Identifying syllable boundaries within consonant clusters
3. Identifying vowel sequences that form diphthongs
4. Identifying compound word boundaries

In addition, languages with weak letter to sound relationship would require identifying digraphs and trigraphs in various contexts, such as 'th' -> /dh/ or 'ou' -> /u/ in english or 'sch' -> /S/ in German, but this problem is not currently dealt with. The initial language independent syllabification method described below is based on simple, known universal tendencies in world's languages. Future improvements will include augmenting the method with acoustic evidence and user feedback.

3.1 Vowel – Consonant identification

Tho vowel-consonant classification is performed with iterative Sukhotin's algorithm [3], where the only assumptions made of the language are that vowels and consonants tend to alternate, and that the most frequent letter of the language is a vowel. Observing the performance of the default algorithm, slight modifications were made, by including information on word boundaries and treating letter clusters of high mutual information as single letters. The performance of the modified algorithm on selected languages can be found below. The remaining problems relate to ambiguity of vowel status of some letters, like 'y' in English ('You' -> /ju:/ 'reallY' -> /ri@li/), which would require context dependent processing.

Figure 3.1a shows results of the syllabification algorithm in terms of grouping vowels and consonants for eight European languages.

3.2 Identifying syllable boundaries within consonant clusters

Typically, three principles are mentioned in literature on the placement of syllable boundaries:

```
Finnish
vowels: a e ä i o u ö y
consonants: g c b d f h k j m l n p s r t v

Hungarian
vowels: a á e i í ű o ő é ó u ö y ú ü
consonants: l t c b d g f h k j m n p s r w v z

Russian
vowels: у е ы э ь я ё а е и о
consonants: б г в д з ж й л к н м п с р т х ф ч ц щ ш ю

Slovak
vowels: a á e i í o é ó u ô y ú ý
consonants: č n ľ ň b d š c ť g f h k j m l p s r t w v z ž

German
vowels: a e ä i o u ö ü
consonants: m j l n s t g b ß c d f h k p r w v ?y x z

English
vowels: a e i o u
consonants: m s t v c g x d b f k j l n p r w h ?y z

Spanish
vowels: a á e i í o é ó u ú
consonants: j l ñ ?y v c g b d f h k m n p s r t w x z

Romanian
vowels :a ă â e i o î u
consonants: ţ c ş d b g f h k j m l n p s r t w v y x z
```

Figure 3.1a: Letters grouped to vowels and consonants using the Sukothin algorithm.

1. *Legality principle* (LP), states that consonant cluster can act as onset or coda only if the said cluster can begin or end words in the language.

2. *Maximal onset principle* (MOP), states that in ambiguous cases, the onset of the syllable should be extended in expense of the coda of the previous syllable.

3. *Sonority sequencing principle* (SSP), states that the sonority (loudness and voicing) of the sounds should increase from onset to nucleus and then decrease from nucleus to coda.

Of these, SSP can not be implemented without additional acoustic analysis on the sonority of the letter. LP and MOP have been implemented in modified form; Of LP, only onsets are considered, with additional frequency constraint. As real texts often contain foreign names and loanwords with atypical orthography, only word-initial clusters above certain percentage of all word-initial sequences are considered legal. On the whole, the exact placement of boundaries might not be important for TTS purposes, as long as nucleii are correctly placed.

3.3 Identifying diphthongs

We have applied a diphtong guessing method that is, based on the assumption that vowels that are adjacent more often than separated by one letter tend to be diphthongs [4]. Unfortunately, this method is rather weak and the probleme, therefore, lends itself naturally to user input and crowd-sourcing.

3.4 Identifying compound word boundaries

Syllable boundaries should always coincide with compound word boundaries, which are to be acquired by Morfessor catmap analysis, when integrated to the system.

3.5 Initial observations of the method

The described method has been implemented and preliminary evaluation has been performed against supervised methods on Romanian and Finnish. The results differ considerably; for Finnish, over 90% of token types were correctly syllabified, whereas for Romanian there is only 50% agreement on word level. For Finnish, the remaining problems concern diphtong identification, violation of the maximum onset principle and identification of compound word boundaries. In Romanian, more context-dependency in syllabification would be required; vowel sequences form diphthongs depending on position in word as well as morphology, and some letters may represent either vowels or consonants depending on the position in the word. Figure 3.5a shows randomly chosen examples of syllabification for Finnish, Russian, Romanian, German, and Spanish trained and tested on open subtitles word lists (`http://invokeit.wordpress.com/frequency-word-lists/`).

Based on these observations, further development will concern acoustic evidence and user feedback on diphtong identification, integrating Morfessor with syllabification for morphological context dependency and compound words, adding context dependency to vowel identification.

To assess the amount of resources needed, evaluation will be perfomed on system level, comparing voices built with unsupervised syllabification to both baseline method and systems built with gold standard syllabification.

Finnish: muut-tu-vat hyö-kä-tä sil-mäl-lä teh-kää saa-toin poi-ka
mie-luum-min os-ta-maan kau-pun-ki-a ker-to-kaa to-del-li-nen o-len
päi-väl-tä kai-kes-ta dan-ny ku-kas tar-koi-tat-ko kuo-len ta-val-
laan sait-te a-ja huo-leh-tii päi-vä-nä hen-gi-tä o-pet-ta-ja vaih-
toeh-to-a kol-men kään-tyy pää-see kuo-le-man tu-lem-me ker-ran lä-
hes-tyy ra-hat aa-vis-tus-ta-kaan ky-sy-myk-sen i-kui-ses-ti riit-ti
vuo-ro-si kut-sut-tu mui-ta-kin häi-vy ku-lut-tu-a yh-tään pel-kään-
pä saak-ka ta-voin ar-voi-nen bob-by var-mis-ta mul-la voim-me-ko
Russian: -би-вают и-ди-те ти-на мо-ло-же же-нат мес-тах смот-ре-ли тро-га-ть
ма-ме за-ме-ти-ть не-ль-зя вой-ну бо-ль-шом на-ча-ли си-ла ми-ром по-жи-вае-
шь ос-та-но-ви-те-сь та-ко-ва ся-дь не-на-ви-жу зво-нит пла-не-та предс-тав-ляю
пой-му по-ве-де-ни-е уз-нае-те хре-на ру-ки но-чью хо-ро-шая на-вер-но пост-
ро-и-ть по-лу-чил сле-ды про-дол-жи-ть любо-пыт-но сог-ла-сен все-му на-хо-
дят-ся до-ка-за-те-льст-во де-рев-не пос-ко-ль-ку та-ки-е взгля-ни-те кар-
Romanian: dist-rac-ţie tră-dat vân-tul ha-i-de bu-ni-ca scă-ri chi-na
ca-do-u-ri fu-rios con-ti-nu-at me-di-ca-le bal-ta ga-ta pis-to-lul
a-duc re-pe-ta fa-mi-li-i-le a-lea ur-mă-ri ve-ti ba-tem pa-sa su-năm
con-cur-sul fe-te-le ba-ră do-va-dă fe-ri-ci-te ca-zi tâ-năr şo-fe-
rul lu-mi-na prin-ci-pa-lă sol-da-ţi ju-rul pur-tat a-le-gi prie-te-
nul pi-cioa-re surp-rins nu-mes-te na-sul ha-i gân-de-şti ne-ce-sar
din-ţi a-ştept cla-sa u-ma-ne a-ra-ti ver-de sa-la a-mân-do-i tre-zi
German: wo-chen bi-bel sol-cher mit-te ve-rär-gert an-statt mo-ral
ar-me wei-te-res hun-dert bil-lig ein-stel-lung star-ken freun-din
ter-ro-ri-sten gü-te ge-sich-ter städ-te kin-der ü-berp-rü-fen lei-
hen per-son kno-chen zu-ruck pa-ter e-arl jah-ren fürch-ten sag-te
pa-tien-ten way-ne ge-kämpft fo-tos kal-te mur-taugh ju-les un-ten a-
li-ce of-fi-cer in-te-res-siert lie-ben um-so auf-neh-men ge-lan-gen
mor-gens jo-nes rus-si-sche dau-er-te mi-nu-te verb-re-cher loc-ker
dien-sten lö-schen rus-sell ca-sey zeu-gen größ-te te-leg-ramm den-
ken geb-racht blau-en ka-ne mil-lion raus-kom-men dop-pelt ak-ten
Spanish: pre-sen-te par-ti-ci-par po-li-cías bien-ve-ni-do val-le pa-
ta vi-a-jar o-cur-rir ár-bol su-ce-da con-ta-ré gor-do nomb-re vi-vía
sa-lón a-cei-te hi-cie-ra rep-re-sen-ta mue-re di-je-ra cu-ar-tel
men-ti-ras bo-ni-ta al-bert cor-rec-ta co-mió tra-ba-ja-do-res nue-
vos be-bé ra-di-o at-rac-ti-va sob-re-vi-vir mué-ve-te di-as ru-mo-
res ce-nar gra-ve u-sa-do pue-sta i-a bus-can-do dá-me-lo que-da-do
ri-dí-cu-lo per-fec-to ge-ren-te to-tal as-pec-to se-rás a-hi en-can-
to se-guir sos-pe-cho-so ar-chi-vo lla-man-do viu-da ent-re-vi-sta
ha-ber-me mi-nu-to a-gen-tes ver-los que-das pu-tas ga-ra-je co-mien-
do en-se-ñó tri-bu-nal a-ma-da es-co-ger de-bi-ste pi-dió ron-da

Figure 3.5a: Random examples of syllabification for Finnish, Russian, Romanian, German, and Spanish based on the methods described in the text.

4 Pseudo-morphological analysis (Morfessor)

A small proportion of the text in this section is similar to the part of D4.1 concerning Morfessor – this is intentional and is designed to improve the readability of both deliverables in isolation without requiring excessive cross-referencing.

4.1 Morfessor Baseline

Morfessor Baseline is a method for unsupervised segmentation of words into morphs using a probabilistic model. A morph is theoretically the smallest part of language carrying meaning. Although similar to syllables in appearance (a segmentation of a word into one or more substrings), morphs actually segment a word into information units instead of phonological ones. Originally developed at AALTO [5, 6], improvements made within SIMPLE4ALL [7] have now been integrated into the new front-end.

In training, a model is defined with a probabilistic model based on the Minimum Description Length principle [8]. The MAP estimate of this model is optimized using an iterative procedure that tries to (recursively) split all morphs present in the model to see if there is any increase in the sum of the lexicon cost and the likelihood of the data. The final parameters of the model correspond to the segmentation of the training data. The method is in its basic form completely unsupervised. No segmentation examples have to be provided to train the model.

After training of the model single utterances can be processed by the Morfessor module. The words are segmented using Viterbi-segmentation to find the most likely split. The resulting morphs are stored as nodes below the word level. The segmentation of morphs can be used in other tasks like the prediction of prominence tags, but already information about the number of morphs is useful in synthesis.

As the model is completely probabilistic and does not contain any language specific component it is useful for all languages that have words that can be split into morphs. The best results are obtained for agglutinative languages like Finnish and Hungarian. Also other letter based scripts can be split with a varying degree of success. On languages with a logosyllabic script (e.g. Chinese), this method will not result in a useful segmentation.

4.2 FlatCat extension

A new method in the Morfessor family called Morfessor FlatCat has been developed in this project.

FlatCat combines the morphotactic constraints from the Morfessor Cat-ML [9] and Cat-MAP [6] models, with the corpus likelihood weighting and semi-supervised learning introduced in Morfessor Baseline. As opposed to Cat-ML, FlatCat uses maximum a posteriori estimation, removing the need for heuristic controls for the model complexity. The hierarchic lexicon of Cat-MAP, in which morphs can consist of two submorphs, has been replaced with a flat representation where each morph must be spelled out. This has been done to facilitate the corpus likelihood weighting.

FlatCat differs from Morfessor Baseline through the use of morph categories. FlatCat assigns to each morph one of four categories: prefix, stem, suffix, or non-morpheme. Category membership probabilities use as features the length of the morph and the predictability of the context in which it occurs. The use of categories allows word internal relations between morphs to be modeled. An example of the benefit is refraining from incorrectly using a valid suffix as a prefix, such as segmenting the word *swing* as *s + wing*.

The morph categories assigned by FlatCat can be used as features in prominence prediction and other tasks. Tasks that benefit from stemming can emulate it by using only the morphs categorized as stems.

FlatCat introduces three new aspects to the Morfessor family: a novel weight learning method, a new shift operation for the search for an optimal model, and the possibility to receive user feedback through the introduction of labeled data during online training.

4.3 Evaluation

For evaluation of Morfessor Baseline on both English and Finnish the datasets of the Morpho Challenge 2010 [10] are used to test the unsupervised and semi-supervised segmentation that can be done by Morfessor. These results are also published in a more extensive format in [7].

The choice of English and Finnish as testing data make the results cover a broad set of languages. Whereas Finnish is a agglutinative, highly morphological language, English is highly irregular and does not contain many natural morphs.

In Table 4.3a the description of both datasets is shown.

Table 4.3a: The numbers of word types in the English and Finnish Morpho Challenge 2010 data sets [10].

	English	Finnish
Unannoted training set	878 036	2 928 030
Annotated training set	1 000	1 000
Test sets	10×1 000	10×1 000

Table 4.3b: Semi-supervised training. (Morpho Challenge 2010 training data, test set scores.)

Run	Epochs	Pre. (%)	Rec. (%)	F-s. (%)
English				
unsupervised	5	81.42	64.31	71.85
semi-supervised	5	81.93	76.53	79.14
Finnish				
unsupervised	5	82.33	39.18	53.09
semi-supervised	2	82.89	54.26	65.58

For evaluation of the segmentation the micro-average segmentation boundary F-score [11] is used. As shown in Table 4.3b, both in English and Finnish the semi-supervised version improves both in precision and recall, and hence also the F-score. These semi-supervised results are important as the annotations used can be given by the user as feedback, which is relevant for tasks in WP4.

5 Wavelet-based prominence tagging

The basic prosodic component in the system is based on prominence, namely word prominence, which is based on the assumption that the lexical level is the main carrier of utterance and phrase internal structure that is not directly computable from the text. That is, the utterance and phrase level prosody follows the textual structure that is mainly marked with punctuation, whereas the sub-lexical structure can be directly computed from syllable and segmental structure. The word prominence is directly related to all prosodic parameters that the system is designed to handle: more prominent syllables are typically louder – reflected in their spectral and gain structure – and longer in duration, moreover, they typically have a rising-falling type fundamental frequency contour.

We have in earlier studies used successfully four levels of prominence ranging from 0 (totally non-prominent or unaccented) to 3 (emphatic prominence, e.g., narrow prosodic focus). The four levels have been successful for both Finnish ([12]) and English ([13, 14]). The levels are also easy for non-experts to label [15].

In order to be used in a fully unsupervised system, the prominences need to be automatically tagged using acoustic data. To this end we have studied several methods that use supervision to different degrees (see Deliverable 2.1, section 5.9). Here we describe the fully automatic labeling system based on continuous wavelet transform (CWT) that has been implemented into the SIMPLE4ALL system. The labeling system, as well as a full system

that also uses CWT based synthesis for f_0 are further described in two recent publications by the group at UH [16, 15]. The proposed CWT based method simplifies signinficantly the front-end design as it does not rely on iteration in terms of synthesis training.

In [15] we studied how well the CWT decomposed f_0 features corresponded with human labeled prominences for more than 7600 separate words, which were all labeled by three separate labelers (all phonetics students). The results were encouraging with the CWT contour peak level matching temporally to the word level explaining more than 53% of the variance in human based labels (see Figure5.0b.

In addition to f_0, the final system uses a weighted prominence estimate from both f_0 and energy (the gain parameter from the GlottHMM analysis).

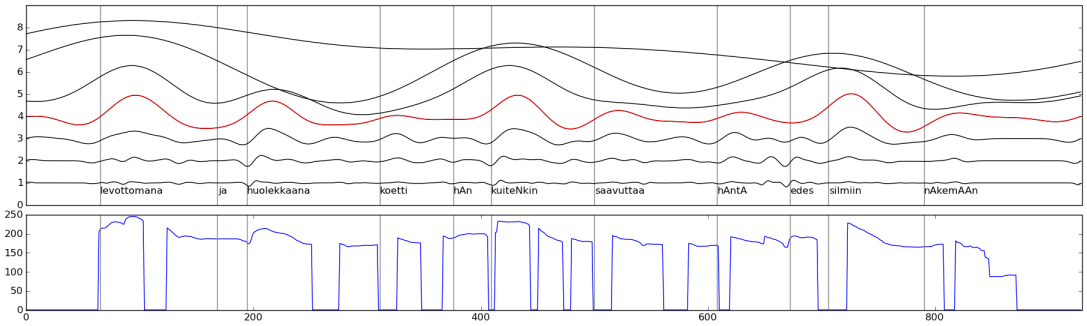

Figure 5.0a: The word prosody scale is chosen from a discrete set of scales with ratio 2 between ascending scales as the one with the number of local maxima as close to the number of words in the corpus as possible. The upper pane shows the representations of f_0 at different scales. The word level (4.2 Hz; see text) is drawn in red. The lower pane shows the f_0 curve. The abscissa shows the frame count from the beginning of the utterance (5 ms frame duration).

5.1 System description

The method described in [15] has been implemented in the voice training part of the Python framework. In addition to f_0, wavelet analysis is also performed on signal energy, and a weighted sum of the normalized word-level wavelet scale of two acoustic features are used to derieve word prominence related contextual features for HMM-training. In addition, a method to add duration features to prominence labelling using a similar wavelet approach is in preparation.

In synthesis time, the prominence labels are predicted with decision-trees, which are currently trained on positional features and morph vector space models. The evaluation of the prominence method will be conducted on Tundra corpora, as the syllabification and morfessor get integrated to the framework.

The initial version of prosody prediction was described in Deliverable 2.1, section 5.6, which centered on pause and phrase prediction.

5.2 Evaluation

The suitability of the CWT based tagging scheme was evaluated for Finnish (as described in [15]). Figure 5.0a shows how the word level changes in f_0 are more varied than e.g., the phrases, which contribute to the overall contour in a regular fashion. Similarly the highest level, which can be interpreted as a general downtrend (or declination) follows from the the whole utterance. The fact that the prominence related changes in the word level cannot be straight-forwardly calculated from the surface f_0 signal is further illustrated in Figure 5.0b how the underlying temporal level corresponding to the word contributes to the overall f_0 in a non-obvious way. In

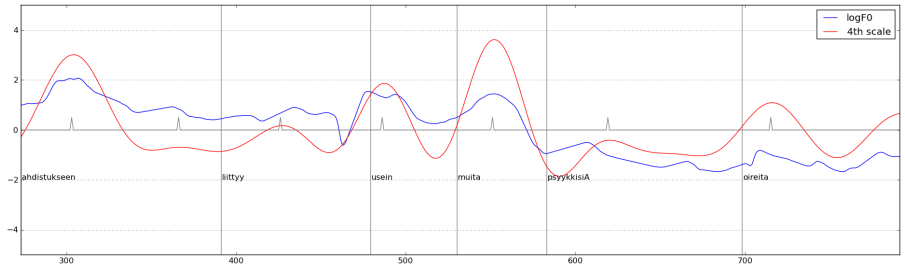

Figure 5.0b: Comparison of selected word scale and original f_0 contour with detected peaks marked with gray triangles. Observe that the wavelet contour is free of noise and declination trend.

summary, the CWT based prominence labeling corresponds very well with the human based values explaining more than 53% of the variance in manual labels. In terms of the very few (four) distinct categories used for prominence in the system, the automatic labeling based on CWT should outperform other methods as it can be extended to include both f_0 and other prominence related variables.

References

[1] Oliver Watts, Adriana Stan, Rob Clark, Yoshitaka Mamiya, Mircea Giurgiu, Junichi Yamagishi, and Simon King. Unsupervised and lightly-supervised learning for rapid construction of tts systems in multiple languages from 'found' data: evaluation and analysis. In *8th ISCA Workshop on Speech Synthesis*, pages 121–126, Barcelona, Spain, August 2013.

[2] Oliver Watts, Adriana Stan, Yoshitaka Mamiya, Antti Suni, Jos Martn Burgos, and Juan Manuel Montero. The Simple4All entry to the Blizzard Challenge 2013. In *Proc. Blizzard Challenge 2013*, August 2013.

[3] BV Sukhotin. Optimization algorithms of deciphering as the elements of a linguistic theory. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 645–648. Association for Computational Linguistics, 1988.

[4] Thomas Mayer. Toward a totally unsupervised, language-independent method for the syllabification of written texts. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 63–71. Association for Computational Linguistics, 2010.

[5] Mathias Creutz and Krista Lagus. Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Technical Report A81, Publications in Computer and Information Science, Helsinki University of Technology, 2005.

[6] Mathias Creutz and Krista Lagus. Inducing the morphological lexicon of a natural language from unannotated text. In Timo Honkela, Ville Könönen, Matti Pöllä, and Olli Simula, editors, *Proceedings of AKRR'05, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 106–113, Espoo, Finland, June 2005. Helsinki University of Technology, Laboratory of Computer and Information Science.

[7] Sami Virpioja and Peter Smit. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report, Aalto University, 2013.

[8] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[9] Mathias Creutz and Krista Lagus. Induction of a simple morphology for highly-inflecting languages. In *Proceedings of the 7th Meeting of the ACL Special Interest Group in Computational Phonology: Current Themes in Computational Phonology and Morphology*, pages 43–51, Barcelona, Spain, 2004. Association for Computational Linguistics.

[10] Mikko Kurimo, Sami Virpioja, and Ville T. Turunen. Overview and results of Morpho Challenge 2010. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 7–24, Espoo, Finland, September 2010. Aalto University School of Science and Technology, Department of Information and Computer Science. Technical Report TKK-ICS-R37.

[11] Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90, 2011.

[12] M. Vainio, A. Suni, and P. Sirjola. Accent and prominence in finnish speech synthesis. *Proceedings of the 10th International Conference on Speech and Computer (Specom 2005), University of Patras, Greece*, pages 309–312, 2005.

[13] A. Suni, T. Raitio, M. Vainio, and P. Alku. The GlottHMM entry for Blizzard Challenge 2010. *The Blizzard Challenge 2010 workshop*, 2010.

[14] A. Suni, T. Raitio, M. Vainio, and P. Alku. The GlottHMM entry for Blizzard Challenge 2012: Hybrid Approach. 2012.

[15] Martti Vainio, Antti Suni, and Daniel Aalto. Continuous wavelet transform for analysis of speech prosody. *In prof. Tools and Resources for Speech Prosody (TRASP), Aix-en-Provence*, 2013.

[16] Antti Suni, Daniel Aalto, Tuomo Raitio, Paavo Alku, and Martti Vainio. Wavelets for intonation modeling in hmm speech synthesis. *In prof. of Speech Synthesis Workshop (SSW) 8, Barcelona*, 2013.

Appendix: Published Papers

Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis

O. Watts[1], *A. Stan*[2], *R. Clark*[1], *Y. Mamiya*[1], *M. Giurgiu*[2], *J. Yamagishi*[13], *S. King*[1]

[1]The Centre for Speech Technology Research, University of Edinburgh, UK
[2]Communications Department, Technical University of Cluj-Napoca, Romania
[3]National Institute of Informatics, Japan

{adriana.stan, mircea.giurgiu}@com.utcluj.ro, Simon.King@ed.ac.uk,
{owatts, Yoshitaka.Mamiya, robert, jyamagis}@inf.ed.ac.uk

Abstract

This paper presents techniques for building text-to-speech front-ends in a way that avoids the need for language-specific expert knowledge, but instead relies on universal resources (such as the Unicode character database) and unsupervised learning from unannotated data to ease system development. The acquisition of expert language-specific knowledge and expert annotated data is a major bottleneck in the development of corpus-based TTS systems in new languages. The methods presented here side-step the need for such resources as pronunciation lexicons, phonetic feature sets, part of speech tagged data, etc. The paper explains how the techniques introduced are applied to the 14 languages of a corpus of 'found' audiobook data. Results of an evaluation of the intelligibility of the systems resulting from applying these novel techniques to this data are presented.
Index Terms: multilingual speech synthesis, unsupervised learning, vector space model, text-to-speech, audiobook data

1. Introduction

Collecting and annotating the data necessary for training a corpus-based text-to-speech (TTS) conversion system in a new language requires considerable time and expert knowledge. Conventionally, audio data for training a synthesiser *back-end* (or waveform generator) will be gathered during a specially-arranged recording session. For this, a recording script must be prepared, a suitable studio must be found, a voice talent must be recruited and speech recording must be carefully supervised. One of the primary goals of the *Simple4All*[1] project is to reduce the time and expert knowledge needed to produce new TTS systems. In [1] we presented a toolkit – developed as part of this project – for segmenting and aligning existing freely-available recordings (audiobooks), circumventing to some extent the need to engineer purpose-recorded speech corpora. The outcome of applying those tools to audiobooks in 14 languages is what we have released under the name of the *Tundra corpus*.

However, the problems associated with TTS data-collection do not stop when we have obtained transcribed speech data for training a synthesiser back-end. TTS systems also require a *front-end* (or text analysis module), which accepts input text and outputs a representation of an utterance suitable for input into the back-end. TTS systems generally represent utterances in terms of units and features based on linguistic knowledge, such as phonemes, syllables, lexical stress, phrase boundaries etc. The components of the front-end that predict these from input text are either made up of hand-written rules or statistical modules; acquiring the expert knowledge required either to manually specify those rules, or to annotate a learning sample on which to train the statistical models, represents a major obstacle to creating a TTS system for a new target language and requires highly specialised knowledge. Such non-trivial tasks include, for example, specifying a phoneme-set or part of speech (POS) tag-set for a language where one has not already been defined; annotating plain text with POS tags, as required to train a POS tagger and annotating the surface forms of words with phonemes to build a pronunciation lexicon.

The toolkit we are developing in *Simple4All* includes tools for constructing TTS front-ends which make as few implicit assumptions about the target language as possible, and which can be configured with minimal effort and expert knowledge to suit arbitrary new target languages. To this end, the modules rely on resources which are intended to be universal, such as the Unicode character database, and employ unsupervised learning so that unlabelled text resources can be exploited without the need for costly annotation. The current paper presents these tools and explains how they were applied to the data of the Tundra corpus to produce TTS systems in 14 languages. We present the results of a listening test of the intelligibility of those systems, and thus evaluate the entire pipeline implemented by our toolkit, which begins with raw found data and ends with trained TTS systems. An initial public version of tools for this whole pipeline (for segmenting and aligning found data and for producing TTS systems with minimal expert knowledge) is due to be released in November 2013.

In prior work addressing the bottleneck in TTS system construction represented by the front-end, unified systems aimed at producing complete systems have generally taken the strategy of providing infrastructure to ease the collection by non-experts of the conventional resources necessary for system construction. This infrastructure might take the form of user-friendly development environments [2], or training and on-going support [3]. Prior work has also presented unsupervised methods for building systems based on letters rather than phonemes [4, 5], induction of phone-sets [6, 7], syllable-like units [8, 9], or lexicons [10]. However, this work has not been presented as an integrated framework for producing end-to-end TTS systems. Furthermore, despite the significant work on unsupervised learning in Natural Language Processing [11, 12] and Information Retrieval [13, 14], potentially useful techniques developed in those fields have not been applied to the problem of TTS front-end induction.

[1] www.simple4all.org/

2. Database

The Tundra corpus [1] is a standardised multilingual corpus designed for text-to-speech research with imperfect or found data. It consists of 14 audiobooks in 14 different languages (Bulgarian, Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Polish, Portuguese, Romanian, Russian and Spanish) and amounts to approximately 60 hours of speech. A complete list of the audiobooks with their sources and durations can be found here `http://tundra.simple4all.org`.

The corpus provides utterance-level alignments obtained with a lightly supervised process described in [15] and [16]. The accuracy of the alignment method, as described in [16] is of 7% SER and 0.8% WER, therefore some light post-processing is required in order to eliminate some of the erroneous utterances. Initial segmentation of the audiobooks into utterance-size chunks was performed using the lightly supervised GMM-based VAD described in [17]. As most of the used audiobooks are recorded in non-specialised environments, the speech data underwent a light cleaning process: normalising the DC offset, applying a multi-band noise gate removal and an RMS-based deverberation method, as described in [1].

3. System Construction

For each of the 14 languages of the Tundra corpus, a TTS system was trained with no reliance of language-specific expertise. Although speaker and recording differences mean that meaningful comparison between languages is difficult, we wished to make the training conditions for the 14 voices as uniform as possible. Therefore, we selected a 1 hour subset of each of the languages' data on which to train voices for this evaluation: the method of data selection we used is explained in Section 3.1. Then text analysis and waveform generation components were trained on that selected data as explained in Sections 3.2 and 3.3, respectively.

3.1. Lightly-supervised data selection

Our principal current interest in audiobook data is that it presents a source of 'found' data from which TTS training databases can be harvested without the need to construct a recording script, recruit a native speaker of the target language, and supervise the recording of a script from scratch. In the present work, therefore, we ignore the other possible advantage of using audiobook data: that harnessing the variety of speaking styles present in audiobooks might enable us to produce less 'mechanical'-sounding TTS systems. Although this is a longer-term goal, we here follow an approach similar to the one presented in [18], which aims to select a neutral subset of a database containing *diverse* speech. In that paper, 9 utterance-level acoustic features are used along with several textual cues to exclude diverse speech from the training set. Thresholds over these features are set manually by the system builder to exclude non-neutral utterances.

For the current work we perform utterance selection using an active learning approach, with uncertainty sampling [19]. Rather than being required to tune thresholds manually, the system builder is presented with example utterances and asked to indicate whether or not they are spoken in a neutral style. The interface therefore insulates the user from the details of the features used, and lets the user focus on what should be key: their intuitive response to hearing speech samples. The procedure we used is as follows:

1) **Feature extraction** First, frame-level features (F_0, energy and spectral tilt – approximated by 1st mel cepstral coefficient) are obtained, from which utterance-level features are computed. The fact that no thresholds need to be manually tuned means that we can afford to use a great many more features than the 9 employed in [18]. Our feature set is based on the one described in [20]: we compute mean, standard deviation, range, slope, minimum and maximum (6-level factor) for $F0$, spectral tilt, and energy (3-level factor) in the following sub-segments of each utterance: entire utterance, 1st and 2nd halfs, all 4 quarters, first and last 100ms, first and last 200ms (11-level factor), giving a total of 198 features.

2) **Initial labelling** The user is presented with the audio of s randomly-selected *seed utterances* from the whole corpus (via a text-based user interface) and asked to label them *keep* or *discard* – utterances are labelled with the user's decision.

3) **Classifier training** A classifier is trained on the labelled examples. Our choice of classifier is a bagged ensemble of decision trees [21] because it can be trained quickly (allowing online active learning in real time), is robust against noisy features and able to accept unnormalised input variables, and mixtures of discrete and continuous input variables (allowing a great many different acoustic features to be used, and different types of features), allows the space of utterances to be partitioned recursively (enabling complex interactions between features to be detected), and provides robust estimates of class probabilities (important for step 4).

4) **Uncertainty sampling** The set of u uncertain examples (utterances about which the classifier is most uncertain – in the present case, the utterances which have closest to 0.5 *keep* probability). The utterances in this set are presented to the user for labelling.

5) Steps 3 and 4 are repeated as many times as time allows.

6) The set of utterances either labelled *keep* by the user are kept for training, as well as enough of the utterances to which the trained classifier gives the highest *keep* probability to, to make up the desired quantity of training data.

For the work presented here, s was set to 15 and u was set to 1. That is, the user was asked to provide 15 labels at the outset, and presented with a single uncertain example at each iteration. The stopping criterion we used in this work was to limit the number of iterations to 15 – in the present, utterance selection time was limited to approximately 20 minutes per language, and 15 was found to be a reasonable number of iterations in that time. Informal comparison suggested the approach outlined is beneficial for this task, but in ongoing work we are testing this rigorously and comparing uncertainty sampling with random sampling, as well as applying our active learning tool to other TTS tasks.

3.2. Front-end construction with unsupervised learning

The TTS front-end building tools used for this work are based on ideas outlined in [22] and applied to Spanish TTS in [23]. Input to the system consists of the audio of utterances selected as described in Section 3.1, together with their text transcription (aligned at the utterance level): in the present case, these are taken from the Tundra corpus, and had been obtained as summarised in Section 2. As an additional input, 5 million words of running text data were obtained from Wikipedia in the target languages for construction of the word- and letter-representations described below.

Text which is input to the system is assumed to be UTF-8 encoded: given UTF-8 text, text processing is fully automatic and makes use of a theoretically universal resource: the Uni-

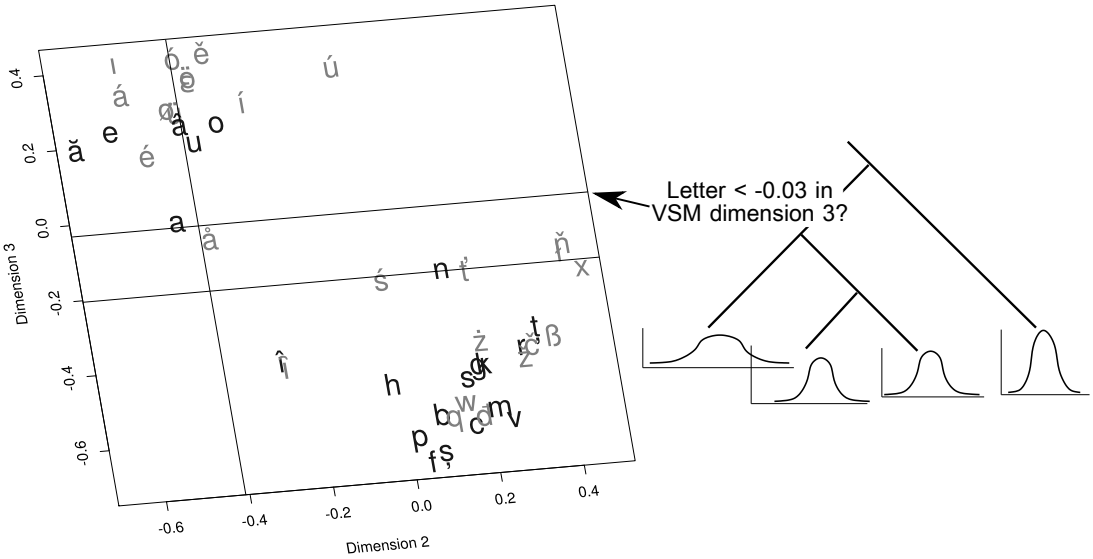

Figure 1: Use of a letter space to replace phonetic knowledge in decision-tree based state-tying. Shown here are 2 dimensions of the actual letter space induced in training the Romanian system described in the paper. The 3 lines bisecting the space represent the 3 questions actually asked in the uppermost fragment (first three 'generations') of the state-tying decision tree for the central state of the model for spectral envelope features. Letters shown in black are 'heard' by the system (i.e. are present in the transcriptions of the audio training data) but ones shown in grey are only 'seen' (i.e. appear only in textual training data) and are mainly foreign language letters.

code database. Unicode character properties are used to tokenise the text and characterise tokens as words, whitespace, punctuation etc. Our modules have so far been successfully applied to a variety of alphabetic (Latin-based, Cyrillic) and alphasyllabic (Brahmic) scripts. Our front-ends currently expect text without abbreviations, numerals, and symbols (e.g. for currency) which require expansion; however, the lightly supervised learning of modules to expand such non-standard words is an active topic of research [24], and we hope to integrate such modules into our toolkit in the near future.

A letter-based approach is used, in which the names of letters are used directly as the names of speech modelling units (in place of the phonemes of a conventional front-end). This has given good results for languages with transparent alphabetic orthographies such as Romanian, Spanish and Finnish, and can give acceptable results even for languages with less transparent orthographies, such as English [22, 4, 5, 7].

The induced front-ends make use of no expert-specified categories of letter and word, such as phonetic categories (vowel, nasal, approximant, etc.) and part of speech categories (noun, verb, adjective, etc.). Instead, features that are designed to stand in for such expert knowledge but which are derived fully automatically from the distributional analysis of unannotated text (speech transcriptions and Wikipedia text) are used. The distributional analysis is conducted via vector space models (VSMs); the VSM was originally applied to the characterisation of documents for purposes of Information Retrieval. VSMs are applied to TTS in [22], where models are built at various levels of analysis (letter, word and utterance) from large bodies of unlabelled text. To build these models, co-occurrence statistics are gathered in matrix form to produce high-dimensional representations of the distributional behaviour of e.g. word and letter types in the corpus. Lower-dimensional representations are obtained by approximately factorising the matrix of raw co-occurrence counts by the application of slim singular value decomposition. This distributional analysis places textual objects in a continuous-valued space, which is then partitioned by decision tree questions during the training of TTS system components such as acoustic models for synthesis or decision trees for pause prediction. For the present voices, a VSM of letters was constructed by producing a matrix of counts of immediate left and right co-occurrences of each letter type, and from this matrix a 5-dimensional space was produced to characterise letters. Token co-occurrence was counted with the nearest left and right neighbour tokens (excluding whitespace tokens); co-occurrence was counted with the most frequent 250 tokens in the corpus. A 10-dimensional space was produced to characterise tokens.

Two dimensions of the letter space induced in training the Romanian system are shown in Figure 1. It can be seen that in these dimensions of the space, vowel and consonant symbols are clearly separable. When a decision tree for clustering acoustic model states is built and allowed to query items' positions in these 2 dimensions, it can use all partitions of the space orthogonal to its axes. A decision tree question such as *Is the letter's value in VSM dimension 3 < -0.03?* is very nearly equivalent to a question based on linguistic knowledge such as *Is the letter a consonant?* The categories of vowel and consonant are useful for clustering acoustic models, and so decision trees actually built using this space use such partitions of the space: the 3 lines shown bisecting the space in the figure represent the 3 questions actually asked in the uppermost fragment (first three 'generations') of the state-tying decision tree for the central state of the model for spectral envelope features.

Distributional analysis places linguistic or textual units in a continuous space which is then partitioned on acoustic evidence. The space constrains the possible groupings of objects that can be considered during decision tree growing. Distributional analysis also allows splits made to generalise to items

that are 'seen' by the system in text data but not 'heard' in the audio data. This is most obviously useful where units such as words are concerned, where many items not present in the training speech corpus are likely to occur at run-time. It can, however, also be useful where letters are concerned, and some examples that illustrate our models' ability to generalise beyond what is heard can be seen in the letter space shown in Figure 1. There, letters shown in black are 'heard' by the system but ones shown in grey are only 'seen' – these are mainly due to foreign language words within Romanian Wikipedia entries. It can be seen that unheard foreign vowels such as *á* and *ö* are suitably placed near the Romanian vowels, and unheard consonants such as *ß* and *q* are placed near the consonants that are actually heard. Splits such as those shown – made only on the basis of the heard items – therefore generalise to unheard items. In the case of letters, this allows rare and foreign letters to be handled despite their absence in the transcriptions of acoustic training data. It can also allow better handling of non-standard spellings: in the case of the vowel *î* (*i* with circumflex), there is a variant (with inverted breve instead of circumflex) which is not present in any of the speech transcriptions but which is used in a few Wikipedia articles. From Figure 1 it can be seen that almost identical representations are learned for these two letters, meaning a decision tree built using those representations will be able to handle the variant form correctly at run-time, even though no instances of that variant were seen in the transcription of the speech training corpus.

The front ends make use of decision trees to predict pauses at the junctures between words. Data for training these trees are acquired automatically by force-aligning the training data with their transcriptions, and allowing the optional insertion of silence between words. The independent variables used by the trees are whether words are separated by punctuation or space, and the VSM features of the tokens preceding and following the juncture.

A rich set of contexts is created using the results of the analysis described here for each letter token in the database. Features include the identity of the letter and the identities of its neighbours (within a 5-letter window), the VSM values of each of those letters, and the distance from and until a word boundary, pause, and utterance boundary. In the current systems, word VSM features are not included directly in the letter contexts, but are used by the decision tree for predicting pauses at runtime.

3.3. Back-end construction

For training the waveform generation modules for the 14 voices, the waveforms of the training corpora were parameterised almost as described in [25]. The one difference is that instead of the committee of different pitch-trackers used in the earlier work, pitch tracks obtained from a glottal source signal estimated by glottal inverse filtering [26] were used for their greater accuracy.

For all systems, speaker-dependent acoustic models were built from this parameterised speech data and the annotation described in Section 3.2, using the speaker-dependent model-building recipe described in [27].

Static and interactive demos of the resulting voices are available at `http://tundra.simple4all.org/demo`. A screen shot of the geographically-organised demo page is shown in Figure 2.

Figure 2: Demo screenshot: this geographical interface to voices can be found at `http://tundra.simple4all.org/demo`.

4. System Evaluation

4.1. Procedure

We are primarily interested in having our systems produce *intelligible* speech; evaluation therefore focused on the intelligibility of TTS output as measured by the word and letter error rates of listeners' transcriptions of those outputs. Conventionally in TTS evaluation, listeners are asked to transcribe semantically unpredictable sentences (SUS) [28]. However, such SUS are not currently available in all the Tundra languages and it is not trivial to construct new SUS, and so we resorted to using short natural sentences from the held-out test sets of the Tundra corpus.

For all 14 Tundra languages, 40 sentences were manually segmented from the held-out chapters of the relevant audiobook. Note that these test sets are distributed with the Tundra corpus, and so the results presented below can be considered benchmarks for future work. An attempt was made to select sentences of 6–8 words in order to make the inherent difficulty of transcription as uniform as possible. However, in some languages these thresholds had to be relaxed; Table 1 gives statistics of test-sentence lengths in all languages.

Subjects for the evaluation were recruited through a web-based crowdsourcing service. The advert for the evaluation specified that native speakers of the relevant language were required; in addition, participation in each part of the evaluation was restricted to users registered in countries where the relevant language is an official or majority language. We attempted to recruit listeners to evaluate all 14 systems built. However, as the option to restrict participation to workers registered in Denmark, Finland and Hungary was not available in the service we used, listening test for only 11 of the systems were publicised. The number of responses from participants varied greatly between languages. At the time of writing, responses from a sufficient number of listeners (25+) had been collected in only 5 of the languages (Bulgarian, English, Italian, Polish and Romanian) . Results for these five languages are presented here; evaluation of the remaining voices is left for future work.

In all languages, two conditions were evaluated: the natural speech of the natural sentences from the test set, and the

Table 1: Statistics of Tundra test-sentence lengths (number of words)

Language	Mean	Standard deviation
German	6.63	0.87
Finnish	6.8	0.91
Bulgarian	6.85	0.83
English	6.88	0.94
Italian	6.9	0.87
Polish	6.95	0.88
Hungarian	7.05	0.81
Russian	7.13	1.18
Danish	7.4	1.19
Portuguese	8.08	1.47
Dutch	8.1	2.15
Romanian	8.55	1.97
French	8.58	1.96
Spanish	8.8	1.65

TTS system reading the same text. In the four languages of the Simple[4]All consortium members (including two of the languages for which results are presented here: Romanian and English), however, SUS were available, and so for those languages a third condition was evaluated: the TTS system producing SUS texts. This is designed to provide a way of broadly gauging the relative difficulty of transcribing natural and SUS sentences, although language and text differences mean it is obviously not advisable to treat extrapolation of the differences to the remaining languages with any great confidence.

The evaluation was run as a set of webpages where participants were asked – using headphones – to listen to the samples and to type in what they heard. Multiple listens were allowed as some of the the natural sentences were longer than the short SUS we would typically use. For the first two conditions, a balanced design was used so that each listener heard each utterance text only once, while each text was heard an equal number of times in both conditions over the whole evaluation. Each listener heard 20 sentences spoken in each condition. For English and Romanian where the SUS condition was also included, listeners heard a further set of 20 SUS sentences.

4.2. Results

Word error rates for the first 2 conditions are shown in Figure 3. For all languages besides English, a similar pattern can be observed: listeners' transcriptions of natural speech attain a WER of 8–12%, and in all cases the TTS system attain WERs approximately 1.5 times worse. This is consistent with the difference between WERs for natural speech and decent benchmark systems in larger scale evaluations on standard corpora. For example, natural speech and the Festival benchmark system attained WERs of 17% and 25% respectively in the 2011 Blizzard Challenge evaluation [29]. The results for English are the exception to the general pattern: the WER for synthetic speech is over 4 times worse than that of natural speech. From prior knowledge and from looking at listeners' transcriptions, it seems clear that this is due to the fact that TTS is based on letters in a language with such an opaque letter-to-sound relationship. In all languages except Polish, the difference between the first two conditions (natural speech and TTS) found to be statistically significant (with $\alpha = 0.05$) using the bootstrap procedure of [30].

As expected, WERs for the SUS sentences are much higher

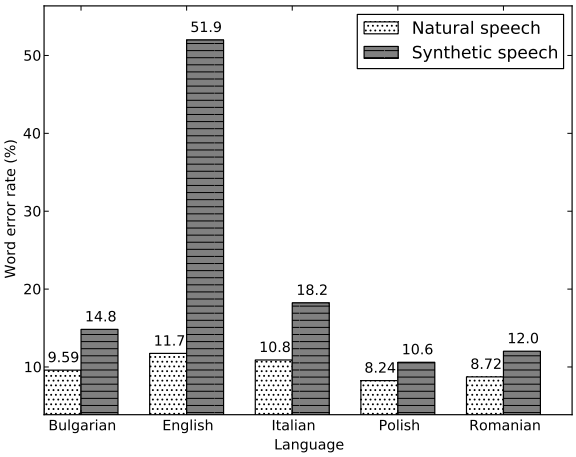

Figure 3: Word error rates for TTS systems and natural speech for 5 of the 14 systems built from the Tundra corpus.

than those for natural sentences: 24.8% and 69.4% for Romanian and English, respectively.

5. Conclusions

We have presented tools for building TTS front-ends in a way that exploits unsupervised learning techniques to side-step the need for language-specific expert knowledge and resources such as pronunciation lexicons, phoneme inventories and part of speech taggers. We have shown how the tools were applied to the languages of the Tundra corpus to produce TTS systems in 14 languages. As we had previously built the Tundra corpus from found data using minimal supervision and language specific knowledge, these TTS systems represent the output of our entire pipeline of tools, and show the type of voice which any interested developer should be able to build using our toolkit (which will be made freely available) despite a lack of language-specific or speech technology expertise, if a source of speech and text data can be found. Five of the voices were evaluated in a listening test for intelligibility, which we consider to show that systems of reasonable quality can be built by applying our tools to publicly available audiobook data, assuming orthographies of similar transparency to those of Bulgarian, Italian, Polish and Romanian. While evaluation of the remaining systems that can be heard in the demo is still ongoing, the results for five languages published here – having been obtained from a standardised, publicly available corpus – are intended to be useful benchmarks against which future work can be compared.

6. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement Nº 287678.

The research presented here has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF: `http://www.ecdf.ed.ac.uk`). The ECDF is partially supported by the eDIKT initiative (`http://www.edikt.org.uk`).

Thanks to Vasilis Karaiskos for setting up the webpages for the listening test.

7. References

[1] A. Stan, O. Watts, Y. Mamiya, M. Giurgiu, R. A. J. Clark, J. Yamagishi, and S. King, "TUNDRA: A Multilingual Corpus of Found Data for TTS Research Created with Light Supervision," in *Proc. of Interspeech (accepted)*, 2013.

[2] J. Kominek, T. Schultz, and A. W. Black, "Voice building from insufficient data – classroom experiences with web-based language development tools," in *Proc. 6th ISCA Speech Synthesis Workshop*, Bonn, Germany, 2007, pp. 322–327.

[3] R. Tucker and K. Shalonova, "Supporting the creation of TTS for local language voice information systems," in *Proc. Interspeech 2005*, Lisbon, Portugal, Sep. 2005, pp. 453–456.

[4] A. Black and A. Font Llitjos, "Unit selection without a phoneme set," in *IEEE TTS Workshop 2002*, 2002.

[5] G. Anumanchipalli, K. Prahallad, and A. Black, "Significance of early tagged contextual graphemes in grapheme based speech synthesis and recognition systems," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-April 4 2008, pp. 4645–4648.

[6] J. Černocký, "Speech processing using automatically derived segmental units: Applications to very low rate coding and speaker verification," Ph.D. dissertation, Universite Paris-Sud, Dec 1998.

[7] M. P. Aylett, S. King, and J. Yamagishi, "Speech synthesis without a phone inventory," in *Interspeech*, 2009, pp. 2087–2090.

[8] P. Mermelstein, "Automatic segmentation of speech into syllabic units," *The Journal of the Acoustical Society of America*, vol. 58, no. 4, pp. 880–883, 1975.

[9] Z. Xie and P. Niyogi, "Robust acoustic-based syllable detection," in *Proceedings of the ICSLP, International Conference on Spoken Language Processing*, 2006.

[10] J. Kominek, "Tts from zero: Building synthetic voices for new languages," Ph.D. dissertation, Carnegie Mellon University, 2009.

[11] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Trans. Speech Lang. Process.*, vol. 4, no. 1, pp. 3:1–3:34, Feb. 2007.

[12] C. Christodoulopoulos, S. Goldwater, and M. Steedman, "Two decades of unsupervised POS induction: How far have we come?" in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, October 2010, pp. 575–584.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, March 2003.

[14] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.

[15] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology, Miami, Florida, USA*, 2012.

[16] A. Stan, P. Bell, J. Yamagishi, and S. King, "Lightly Supervised Discriminative Training of Grapheme Models for Improved Sentence-level Alignment of Speech and Text Data," in *Proc. of Interspeech (accepted)*, 2013.

[17] Y. Mamiya, J. Yamagishi, O. Watts, R. A. Clark, S. King, and A. Stan, "Lightly Supervised GMM VAD to use Audiobook for Speech Synthesiser," in *Proc. ICASSP*, 2013.

[18] N. Braunschweiler and S. Buchholz, "Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality," in *Proc. Interspeech*, Florence, Italy, Aug. 2011, pp. 1821–1824.

[19] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*.

[20] G. Murray, S. Renals, and M. Taboada, "Prosodic correlates of rhetorical relations," in *Proceedings of HLT/NAACL ACTS Workshop, 2006, New York City, USA*, Jun. 2006.

[21] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[22] O. Watts, "Unsupervised learning for text-to-speech synthesis," Ph.D. dissertation, University of Edinburgh, 2012.

[23] J. Lorenzo-Trueba, O. Watts, R. Barra-Chicote, J. Yamagishi, S. King, and J. M. Montero, "Simple4All proposals for the Albayzin Evaluations in Speech Synthesis," in *Proc. Iberspeech 2012*, 2012.

[24] R. San-Segundo, J. M. Montero, V. Lopez-Ludeña, and S. King, "Detecting acronyms from capital letter sequences in Spanish," in *Proc. Interspeech*, Portland, Oregon, USA, Sep. 2012.

[25] J. Yamagishi and O. Watts, "The CSTR/EMIME HTS System for Blizzard Challenge," in *Proc. Blizzard Challenge 2010*, Sep. 2010.

[26] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 153–165, Jan. 2011.

[27] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[28] C. Benoit, M. Grice, and V. Hazan, "The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences," *Speech Communication*, vol. 18, no. 4, pp. 381 – 392, 1996.

[29] S. King and V. Karaiskos, "The blizzard challenge 2011," in *Proc. Blizzard Challenge 2011*, sep 2011.

[30] M. Bisani and H. Ney, "Bootstrap estimates for confidence intervals in ASR performance evaluation," in *Proc. ICASSP '04*, vol. 1, 2004, pp. 409–12.

The Simple4All entry to the Blizzard Challenge 2013

O. Watts[1]*, A. Stan*[2]*, Y. Mamiya*[1]*, A. Suni*[3]*, J.M. Burgos*[4]*, J.M. Montero*[4]

[1]Centre for Speech Technology Research, University of Edinburgh, UK
[2]Communications Department, Technical University of Cluj-Napoca, Romania
[3]Institute of Behavioural Sciences, University of Helsinki, Finland
[4]Speech Technology Group, ETSIT, Universidad Politécnica de Madrid, Spain
owatts@inf.ed.ac.uk, adriana.stan@com.utcluj.ro, Antti.Suni@helsinki.fi,
jose.martin@die.upm.es, juancho@die.upm.es

Abstract

We describe the synthetic voices entered into the 2013 Blizzard Challenge by the SIMPLE4ALL consortium. The 2013 Blizzard Challenge presents an opportunity to test and benchmark some of the tools we have been developing to address two problems of interest: 1) how best to learn from plentiful 'found' data, and 2) how to produce systems in arbitrary new languages with minimal annotated data and language-specific expertise on the part of the system builders. We here explain how our tools were used to address these problems on the different tasks of the challenge, and provide some discussion of the evaluation results.

Index Terms: statistical parametric speech synthesis, speech alignment, speech segmentation, style diarisation, unsupervised learning, vector space model, audiobook data, glottal inverse filtering, glottal flow pulse library

1. Introduction

This paper describes the synthetic voices entered into the 2013 Blizzard Challenge by the SIMPLE4ALL consortium. SIMPLE4ALL is a European speech synthesis project focused on creating speech synthesis technology that learns from data with little or no expert supervision.[1] The 2013 Blizzard Challenge provides a good opportunity to test and benchmark some of the techniques we have been developing within the project. Two problems of central importance for SIMPLE4ALL are 1) how best to learn from plentiful 'found' data, and 2) how to produce systems in arbitrary new languages with minimal annotated data and language-specific expertise on the part of the system builders. We here explain how the different tasks of the challenge relate to the problems of interest, and give an overview of how we applied four parts of the SIMPLE4ALL toolkit to the tasks.

Obtaining and transcribing the speech data for training a corpus-based text-to-speech (TTS) system in a new language requires considerable time and expert knowledge. Typically, this speech data is collected during a specially-arranged recording session, for which a recording script has to be prepared, a suitable studio must be found, a voice talent must be recruited and speech recording must be carefully supervised. SIMPLE4ALL aims to ease the building of new voices by developing and distributing tools which allow the reuse of speech data produced for other purposes. A prime example of such 'found' data is freely available audiobook recordings which have been released into the public domain. In [1] we presented a part of our toolkit for segmenting and aligning such recordings, allowing us to circumvent the need to engineer purpose-recorded speech corpora where existing recordings are available. Task EH1 of the challenge lets us test tools addressing this problem, as it involves building a voice from a very large set of audiobook data which is provided as approximately 300 hours of chapter-sized mp3 files.

As well as obtaining a segmentation and alignment for audiobook data, it is also important to deal with the heterogeneity of such data. To this end, another part of the SIMPLE4ALL toolkit was used to provide diarisation of the automatically obtained corpora. If audio from radio broadcasts are to be used for training a TTS system, for example, it is crucial to diarise audio into speech and non-speech (e.g. music, applause, laughter). When pure speech has been obtained, it is further necessary to diarise it into separate speakers, and it may also be desirable to diarise a single speaker's speech into different emotions or speaking styles [2]. Ultimately the goal of the latter would be to build a synthesiser capable of producing speech in a variety of styles. A more short-term approach is to exclude more unusual speaking styles to produce a subset of relatively homogeneous and neutral speech. This gives a set of training data which is as much like a conventional TTS database as possible, but which doesn't incur the associated costs. This is the approach taken here.

A third part of the SIMPLE4ALL toolkit used for our Blizzard Challenge entry is designed to enable the construction of systems in languages where we have access to little or no linguistic expertise or expert-annotated data. We think it is valuable for speech technology to venture beyond the handful of the world's languages where resources such as text normalisers, lexicons and part-of-speech taggers already exist. Thus, part of the SIMPLE4ALL toolkit includes tools for constructing TTS front-ends which make as few implicit assumptions about the target language as possible, and which can be configured with minimal effort and expert knowledge to suit arbitrary new target languages. To this end, the modules rely on resources which are intended to be universal, such as the Unicode character database, and employ unsupervised learning so that unlabelled text resources can be exploited without the need for costly annotation. Task IH1 lets us test tools addressing this problem, as it involves building voices for four Indian languages (Hindi, Bengali, Kannada and Tamil) for which the consortium members have no language-specific expertise or resources.

The fourth and final part of the SIMPLE4ALL toolkit used for our Blizzard Challenge entry is an implementation of new

[1] www.simple4all.org/

speech signal models capable of modelling a large variety of speaking styles and vocal emotions [3].

We note that an initial public version of tools for this whole pipeline of tools is due to be released in November 2013.

2. System Description

2.1. Data preparation

As already mentioned, the training data for task EH1 of the challenge was provided without a sentence-level speech segmentation and text alignment. Therefore one of the sub-tasks was to obtain the correct alignment, prior to building the synthetic voices. Our previous work on automatic alignment of speech with imperfect transcripts [4, 5, 6] has developed tools to perform the alignment without the use of high-level language expertise or existing acoustic models. The method involves two major steps: *1) a sentence-level segmentation of the speech data*, and *2) automatic alignment of speech and text at sentence-level*. Both steps are lightly supervised and require only a minimum amount of manually labelled data, also called *initial training data*. The following paragraphs describe them in more detail.

Step 1. Speech segmentation is performed using a 16 Gaussian Mixture Model (GMM)-based voice activity detection algorithm [6]. Two GMMs are trained, one for silence and one for speech, from 10 minutes of manually-labelled data, in which the inter-sentence silences are marked. Feature vectors consist of energy, 12 dimensional MFCCs, their deltas and the number of zero crossings. After training the GMMs, for each frame within the manually-labelled data, we compute the the log likelihood ratio, followed by a median filter smoothing. This process also detects short intra-sentence silences. In order to discriminate between inter- and intra- sentence silence frames, two Gaussian probability distribution functions are fitted onto the histogram of silence durations. Their intersection represents the threshold for sentence boundary silence duration. The GMMs are then run on the entire speech resource. Results showed over 96% accuracy in sentence boundary detection.

Step 2. The speech alignment step starts from the same 10 minutes of initial training data, which is now segmented and needs to be orthographically transcribed. A first set of poor initial grapheme-level acoustic models is built from it. The models are then used to recognise the entire speech resource with the help of a highly restricted word network built from the full text transcript (see [4] for more details). To determine the correctly recognised utterances, the recognition is run over the speech data with various degrees of freedom within the word networks, and the obtained acoustic scores are compared. Confident data is then used to re-train the acoustic models, and the process repeats. A final step in the alignment is the re-estimation of the acoustic models using tri-graphemes, and this increases the aligned data by over 40% relative. However, for short speech resources, this step might be unfavourable, as the number of tri-graphemes is too large to obtain satisfactory statistics for them. Previous results obtained with an English audiobook showed an average 75% confident data with a 7% SER and 0.5% WER [5].

For the Blizzard Challenge task EH1 , each audiobook was segmented and aligned individually, aligned percentages being similar to our previous results.

2.2. Data selection

The speaker diarization system described in [7] was used to cluster the segmented utterances obtained as described in section 2.1 for a single audiobook. As we are clustering the speech of a single speaker, the result is a set of 'pseudo-speakers', each corresponding to some automatically detected speaking style as in [2]. A difference in the current case is that we seek only a single cluster of neutral style speech to use, and discard the other clusters. 12 such clusters were produced by an iterative process of speaker segmentation and agglomerative clustering of segments. For each sentence, the system output the dominant 'speaker' of the sentence and the purity of the sentence (fraction of the sentence spoken by the dominant speaker). A single cluster accounted for 90% of the sentences processed – informal listening suggested that this corresponded well with the speaker's neutral style of reading. Taking only the completely pure utterances reduced this to 89%.

For the EH1 voice acoustic models, a 5 hour subset of this pure neutral data was selected. Note however that the whole of the data for which a confident alignment was obtained (section 2.1) was used for the pause prediction model (see section 2.4).

2.3. Text processing

The tools used for building TTS front-ends for entries to all parts of the challenge are based on ideas outlined in [8], applied to Spanish TTS in [9], and to 14 different languages in [10]. We summarise the tools here, drawing heavily on descriptions given in those previous publications.

Input to the system consists of the audio of utterances together with their text transcription. For the EH1 voice, these utterances made up 5 hours of the neutral speech extracted as described in Section 2.2. For each of the Indian languages of task IH1, 950 of the available 1000 sentences and their plain orthography UTF-8 transcriptions were used as input; 50 sentences were set aside for use as an internal development set.

As well as the training speech data and its transcripts, our tools exploit the large amount of unannotated text data which is available for many languages on the web. For the task IH1 voices, this consisted of approximately 13.4, 2.2, 4.4 and 6.4 million tokens of text for Hindi, Bengali, Kannada and Tamil, respectively, which we obtained from Wikipedia. For the English voice for Task EH1, we used only the transcripts of the full audiobook training corpus only as we wanted to experiment with using only in-domain data. For all languages, these unannotated text data were used for construction of the word- and letter-representations described below.

Text which is input to the system is assumed to be UTF-8 encoded: given UTF-8 text, text processing is fully automatic and makes use of a theoretically universal resource: the Unicode database. Unicode character properties are used to tokenise the text and characterise tokens as words, whitespace, punctuation etc. Our front-ends currently expect text without abbreviations, numerals, and symbols (e.g. for currency) which require expansion; however, the lightly supervised learning of modules to expand such non-standard words is an active topic of research [11], and we hope to integrate such modules into our toolkit in the near future.

A letter-based approach is used, in which the names of letters are used directly as the names of speech modelling units (in place of the phonemes of a conventional front-end). This has given good results for languages with transparent alphabetic orthographies such as Romanian, Spanish and Finnish, and can give acceptable results even for languages with less transparent orthographies, such as English [8, 12, 13, 14]. We decided to submit letter-based systems for both the EH1 and IH1 tasks, even though high-quality lexicons are available for English. Al-

though the complicated letter-to-sound relations of English orthography mean that we expect this to severely degrade synthesis quality, we wished to make use of the opportunity presented by the Blizzard Challenge to evaluate this naive approach using many listeners against state-of-the-art systems. In this way, we have a useful benchmark against which to compare the results of ongoing attempts to tackle the same problem in a less naive way.

The induced front-ends make use of no expert-specified categories of letter and word, such as phonetic categories (vowel, nasal, approximant, etc.) and part of speech categories (noun, verb, adjective, etc.). Instead, features that are designed to stand in for such expert knowledge but which are derived fully automatically from the distributional analysis of unannotated text (speech transcriptions and Wikipedia text) are used. The distributional analysis is conducted via vector space models (VSMs); the VSM was originally applied to the characterisation of documents for purposes of Information Retrieval. VSMs are applied to TTS in [8], where models are built at various levels of analysis (letter, word and utterance) from large bodies of unlabelled text. To build these models, co-occurrence statistics are gathered in matrix form to produce high-dimensional representations of the distributional behaviour of e.g. word and letter types in the corpus. Lower-dimensional representations are obtained by approximately factorising the matrix of raw co-occurrence counts by the application of slim singular value decomposition. This distributional analysis places textual objects in a continuous-valued space, which is then partitioned by decision tree questions during the training of TTS system components such as acoustic models for synthesis or decision trees for pause prediction. For the present voices, a VSM of letters was constructed by producing a matrix of counts of immediate left and right co-occurrences of each letter type, and from this matrix a 5-dimensional space was produced to characterise letters. Token co-occurrence was counted with the nearest left and right neighbour tokens (excluding whitespace tokens); co-occurrence was counted with the most frequent 250 tokens in the corpus. A 20-dimensional space was produced to characterise word tokens.

2.4. Pause Prediction

Phrase-break prediction is an essential part in text-to-speech synthesis because it determines the rhythm, as well as prominence in the output synthetic speech. As previously stated, our system tries to avoid supervised and language-dependent modules. Hence, our phrase-break prediction step is also lightly supervised, and we treat silences detected from the acoustics as surrogate phrase-breaks. We exploit the large amount of speech data made available for task EH1, and extract a training set from the forced alignment of the audio and its corresponding orthographic transcripts obtained in the alignment step (see Section 2.1). (The same approach was used for the IH1 voices, except in those cases the training corpus was much smaller and a sentence segmentation was already available.) To discriminate between the short inter-word pauses and pauses which might signal actual phrase-breaks, we plotted the histogram of all the silence segments within the available data. This lead to a separation threshold of 200 ms. Silences below this threshold were discarded and added to the no-pause (NP) set. A list of all the consecutive pairs of words from the text and the length, and existence of a phrase break constitutes our training data. This method works under the assumption that the test data will be part of the same domain as the training one (i.e. audiobooks), and the phrase break durations would be similar, which also means that the method is corpus-dependent.

But, as the surface form of the words does not inherently contain enough information to predict the phrase breaks, we rely on the vector representations of words mentioned in section 2.3. The vectors for each pair of consecutive words from the training data, along with their pause indicator constitute the input for a classification and regression tree. Results showed an overall 0.9 F-measure, but only an 0.4 F-measure for *pause* instances (P). This is mostly due to the unbalanced training data set (i.e. there are more NP word pairs than P). Even when the set was artificially built from equal amounts of and NP pairs, the results remained similar. This might be caused by the VSMs not being able to capture the essential features required for the pause prediction, and hence a more elaborate set of features would be beneficial in future work.

Punctuation is also an important pause indicator, and so we included the punctuation marks as word-pair constituents. This lead to an increase of 0.1 in the F-measure of the P class. Still the results are below expectation, but we estimate that they are caused by the poor alignment of speech with its orthographic transcripts, especially for English which is known to have a high letter-to-sound complexity.

To estimate the phrase breaks in the testing data, we converted the sentences into word pairs, extracted their corresponding vectors and predicted the P/NP class with the previously trained CART.

2.5. Acoustic Modelling

As mentioned previously, a five-hour subset of the available training corpus for EH1 was used to train acoustic models. The inconsistent recording conditions and small amounts of training data for the IH1 tasks meant that extra robustness for acoustic parameterization and training was required. The 4 IH1 voices were each built in an identical fashion, except that half of the Bengali training data was discarded due to being recorded in excessively reverberant conditions. Various other inconsistencies were present too. Style-adaptive training and the use of extra contextual labels were considered for distinguishing these different recording conditions, but our tools for unsupervised recording quality classification are not yet ready.

2.5.1. Parameterisation

For the EH1 voice, the training data were parameterised using STRAIGHT, almost as described in [15]. The only difference is that instead of the committee of different pitch-trackers used in the earlier work, pitch tracks obtained with GlottHMM (using a glottal source signal estimated by glottal inverse filtering [3]) were used for their greater accuracy.

For the IH1 voices, full GlottHMM parameterisation [3] was used after initial denoising of the training speech. 24 vocal tract LSF coefficients and 10 voice source LSF coefficients were extracted as well as harmonic-to-noise ratio with 5 bands, energy and F0. Pulse libraries [16] were extracted from 10 utterances for each voice.

Some alterations to the parameterization scheme described in [3] were made to increase robustness. First, the iterative adaptive inverse filtering method was replaced with direct inverse filtering using a pre-emphasis filter only. Second, the pre-emphasis filter was added to unvoiced analysis, to ensure continuous LSF trajectories across voicing boundaries, thus reducing the audible distortion of voicing errors.

Notably, we did not use the vocal tract LSF parameters directly in the training, but instead converted the parameters to mel-cepstral representation via LPC spectrum. As mel-cepstral coefficients are decorrelated, focus on perceptually relevant frequencies and provide smooth trajectories, they might be more suitable than LSFs for HMM training, especially on difficult material such as the current challenge. Further investigation on this topic would be needed to verify this.

2.5.2. Training and synthesis

A rich set of contexts was created using the results of the analysis described in section 2.3 for each letter token in the training data for all languages. Features used include the identity of the letter and the identities of its neighbours within a window of given length. A 5-letter window was used for the IH1 voices, and a 9-letter window for the EH1 voice. Some informal experiments suggested this to be an appropriate size for the 5 hour subset of the EH1 data we used. Additional features were the VSM values of each letter in the window, and the distance from and until a word boundary, pause, and utterance boundary.

For the EH1 voice, speaker-dependent acoustic models were built from the parameterised speech data and labelling using the speaker-dependent model-building recipe described in [17].

For the IH1 voices, the HMM models were trained with the standard HTS 2.0 [18] recipe, modified for additional GlottHMM streams, but using three iterations of decision tree clustering instead of two. MGE training was also applied. Parameter generation was performed considering global variance, with stream-dependent thresholds. Generated mel-cepstral coefficients were converted back to the LSF form for stability checking and vocoding purposes. Excitation was generated using the PCA-mean pulse approach [19].

Informal listening by the authors and feedback from several native speakers suggested that the denoised GlottHMM version performed better than previous SIMPLE^{4}ALL voices built on the same data using the STRAIGHT vocoder, but detailed analysis of the exact reasons for this improvement remains to be done.

3. Results

The identifier for our system in the published results is P.

On Task EH1 ours was consistently the worst-performing system of all entries. On the intelligibility sections of the evaluation, there was a c.10% gap in WERs between our system and the second worst performing one. This gap was higher among the paid subset of listeners, and lower among online volunteers and speech expert listeners, where it dropped to c.5–6%.

Performance relative to the other systems in the IH1 tasks was much better. For the speaker similarity and naturalness sections of the evaluation for all 4 languages, our system tends to score somewhere in the middle of all TTS systems. The intelligibility results published for Hindi and Kannada follow a similar pattern. In the Hindi test, 4 TTS systems achieved lower WERs than ours, 1 was worse, and 1 scored within 1% WER of our system; in the paid listener subset, our system achieves precisely the middle rank in the Hindi intelligibility results. In both listener group sections of the Kannada intelligibility test, our system also achieves precisely the middle rank.

4. Conclusions

The poor performance of our system in EH1 was anticipated due to the difficulty of TTS from the surface orthographic forms of English words, and to the high level of expertise that has been accumulated for doing TTS in English where there is no self-imposed limit on the amount of target-language expertise that can be used in a system. However, we wished to know exactly how much the lack of a lexicon would set us back in an extensive evaluation with many listeners. Furthermore, these results are envisaged as being useful for on-going improvements to our system, where light supervision and unsupervised lexicon induction techniques are exploited. Because Blizzard stimuli are released after the challenge, it is possible to evaluate improved systems by re-running the evaluation locally on a smaller scale, using a subset of 'landmark' systems from the challenge which allow new results for improved systems to be placed among existing Blizzard results. Having our own baseline among the original results is useful for sanity-checking when projecting results for new systems into the space of existing results.

We regard the middling performance of our system on the IH1 tasks as a success, given that the system makes no use of expert script knowledge, while we assume that other systems probably all make use of at least the phonetic annotation distributed for the challenge. This is the first formal evaluation of our letter-based front-end as applied to a non-alphabetic script: we regard its reasonable performance on the four alphasyllabic scripts of IH1 as a validation for the unsupervised approach for our main target domain of under-resourced languages.

5. References

[1] A. Stan, O. Watts, Y. Mamiya, M. Giurgiu, R. A. J. Clark, J. Yamagishi, and S. King, "TUNDRA: A Multilingual Corpus of Found Data for TTS Research Created with Light Supervision," in *Proc. of Interspeech (accepted)*, 2013.

[2] J. Lorenzo, B. Martinez, R. Barra-Chicote, V. LopezLudena, J. Ferreiros, J. Yamagishi, and J. Montero, "Towards an unsupervised speaking style voice building framework: Multistyle speaker diarization."

[3] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, and P. Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 153–165, 2011.

[4] A. Stan, P. Bell, and S. King, "A grapheme-based method for automatic alignment of speech and text data," in *Proc. IEEE Workshop on Spoken Language Technology, Miami, Florida, USA*, 2012.

[5] A. Stan, P. Bell, J. Yamagishi, and S. King, "Lightly Supervised Discriminative Training of Grapheme Models for Improved Sentence-level Alignment of Speech and Text Data," in *Proc. of Interspeech (accepted)*, 2013.

[6] Y. Mamiya, J. Yamagishi, O. Watts, R. A. Clark, S. King, and A. Stan, "Lightly Supervised GMM VAD to use Audiobook for Speech Synthesiser," in *Proc. ICASSP*, 2013.

[7] J. Pardo, R. Barra-Chicote, R. San-Segundo, R. de Cordoba, and B. Martinez-Gonzalez, "Speaker Diarization Features: The UPM Contribution to the RT09 Evaluation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 426–435, 2012.

[8] O. Watts, "Unsupervised learning for text-to-speech synthesis," Ph.D. dissertation, University of Edinburgh, 2012.

[9] J. Lorenzo-Trueba, O. Watts, R. Barra-Chicote, J. Yamagishi, S. King, and J. M. Montero, "Simple4All proposals for the Albayzin Evaluations in Speech Synthesis," in *Proc. Iberspeech*, 2012.

[10] O. Watts, A. Stan, R. Clark, Y. Mamiya, M. Giurgiu, J. Yamagishi, and S. King, "Unsupervised and lightly-supervised learning for rapid construction of TTS systems in multiple languages from 'found' data: evaluation and analysis," in *Proc. of 8th ISCA Workshop on Speech Synthesis*, 2013.

[11] R. San-Segundo, J. M. Montero, V. Lopez-Ludeña, and S. King, "Detecting acronyms from capital letter sequences in Spanish," in *Proc. Interspeech*, Portland, Oregon, USA, Sep. 2012.

[12] A. Black and A. Font Llitjos, "Unit selection without a phoneme set," in *IEEE TTS Workshop 2002*, 2002.

[13] G. Anumanchipalli, K. Prahallad, and A. Black, "Significance of early tagged contextual graphemes in grapheme based speech synthesis and recognition systems," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-April 4 2008, pp. 4645–4648.

[14] M. P. Aylett, S. King, and J. Yamagishi, "Speech synthesis without a phone inventory," in *Proc. of Interspeech*, 2009, pp. 2087–2090.

[15] J. Yamagishi and O. Watts, "The CSTR/EMIME HTS System for Blizzard Challenge," in *Proc. Blizzard Challenge 2010*, Sep. 2010.

[16] T. Raitio, A. S. H. Pulakka, M. Vainio, and P. Alku, "Utilizing glottal source pulse library for generating improved excitation signal for HMM-based speech synthesis," in *Proc. ICASSP*, 2011.

[17] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[18] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of 7th ISCA Workshop on Speech Synthesis*, 2007, pp. 294–299.

[19] T. Raitio, A. Suni, M. Vainio, and P. Alku, "Comparing glottal-flow-excited statistical parametric speech synthesis methods," in *Proc. ICASSP*, 2013.

Wavelets for intonation modeling in HMM speech synthesis

Antti Suni[1]*, Daniel Aalto*[1]*, Tuomo Raitio*[2]*, Paavo Alku*[2]*, and Martti Vainio* [1]

Institute of Behavioural Sciences (SigMe Group), University of Helsinki, Finland
[2]Department of Signal Processing and Acoustics, Aalto University, Finland
antti.suni@helsinki.fi, daniel.aalto@helsinki.fi, tuomo.raitio@aalto.fi
paavo.alku@aalto.fi, martti.vainio@helsinki.fi

Abstract

The pitch contour in speech contains information about different linguistic units at several distinct temporal scales. At the finest level, the microprosodic cues are purely segmental in nature, whereas in the coarser time scales, lexical tones, word accents, and phrase accents appear with both linguistic and paralinguistic functions. Consequently, the pitch movements happen on different temporal scales: the segmental perturbations are faster than typical pitch accents and so forth. In HMM-based speech synthesis paradigm, slower intonation patterns are not easy to model. The statistical procedure of decision tree clustering highlights instances that are more common, resulting in good reproduction of microprosody and declination, but with less variation on word and phrase level compared to human speech. Here we present a system that uses wavelets to decompose the pitch contour into five temporal scales ranging from microprosody to the utterance level. Each component is then individually trained within HMM framework and used in a superpositional manner at the synthesis stage. The resulting system is compared to a baseline where only one decision tree is trained to generate the pitch contour.

Index Terms: HMM-based synthesis, intonation modeling, wavelet decomposition

1. Introduction

The fundamental frequency (f_0) contour of speech contains information about different linguistic units at several distinct temporal scales. Likewise prosody in general, f_0 is inherently hierarchical in nature. The hierarchy can be viewed in phonetic terms as ranging from segmental perturbation (i.e., microprosody) to a levels that signal phrasal structure and beyond (e.g., utterance level downtrends). In between there are levels that signal relations between syllables and words (e.g., tones and pitch accents). Consequently, the pitch movements happen on different temporal scales: the segmental perturbations are faster than typical pitch accents, which are faster than phrasal movements and so on. These temporal scales range between several magnitudes from a few milliseconds to several seconds and beyond.

In HMM-based speech synthesis paradigm, all modeling is based on phone sized units. In principle, slower intonation patterns are more difficult to model than segmentally determined ones. Moreover, the statistical procedure of decision tree clustering highlights instances that are more common, resulting in a good reproduction of microprosody and overall trends (such as general downtrends) and relatively poor reproduction of prosody at the level of words and phrases. This shortcoming calls for methods that take into account the inherent hierarchical nature of prosody.

Traditionally the problem has been approached by using superpositional models which separate syllable and word level accents from phrases [2, 7]. On feature extraction side, discrete cosine transform parameterization of f_0 has been investigated, providing compact representation of the pitch contour [12]. Typically, each voiced segment or syllable and phrase are parameterized with a constant number of DCT coefficients, statistical clustering is performed based on contextual features, and synthesis is performed in additive fashion [11]. However, the constant number of coefficients is problematic for variable length units, and natural continuity between units is difficult to achieve.

In HMM framework, decomposition of f_0 to its hierarchical components during acoustic modeling has been investigated [4, 15]. These approaches rely on exposing the training data to a level-dependent subset of questions for separating the layers of the prosody hierarchy. The layers can then be modeled separately as individual streams [4], or jointly with adaptive training methods [15]. Results indicate that syllable level modeling improves prosody whereas higher levels do not provide benefits.

In HMM-based speech synthesis, f_0 is modeled jointly with voicing decision. The unit of modeling is typically a phone HMM with five states. For each state, predefined contextual questions concerning phones, syllables, words and phrases are used to form a set of possible splits in a decision tree. The splitting decisions are made in a greedy fashion based on likelihood increase. Thus the hierarchical nature of intonation is only implicitly addressed by questions on different levels of hierarchy. With multiple levels, including voicing decision, modeled by a single set of trees, the rare or slow events can not be modeled robustly, due to fragmentation of the training data by previous, more urgent splits for the short time scale of the model.

In this paper, we present a solution to the problems outlined above based on continuous wavelet transform (CWT). The CWT is used to decompose the f_0 contour into several temporal scales that can be used to model the levels ranging from microprosody to the utterance level separately. As well as separating the contour into meaningful temporally assigned levels – ranging from microprosody to utterance level prosody – the CWT produces a continous f_0 contour which has further merits. Earlier, wavelets have been used in speech synthesis context for parameter estimation [3, 6, 10].

We chose four f_0 modeling methods for comparison: (1) The normal HTS method using the MSD stream, and two

wavelet-based setups modeling the f_0 contour on several distinct levels: (2) one with a joint model and (3) one where five separate CWT based levels are modeled separately. In addition, (4) a continous interpolated f_0 stream model was added. The fourth method was added in order to evaluate the wavelet based methods against another model using continuous trajectories since interpolation alone has been reported to improve f_0 modeling [14].

Objective comparison of the proposed methods is presented against single-stream baselines using two GlottHMM [9] Finnish voices trained from a male and a female corpus.

2. Pitch decomposition and wavelets

2.1. Extraction and preprocessing of f_0

GlottHMM vocoder was used for estimating the fundamental frequency (f_0) of speech. GlottHMM is a physiologically oriented vocoder that uses glottal inverse filtering for separating speech into the glottal source signal and the vocal tract filter. The iterative adaptive inverse filtering (IAIF) method is used for the separation, and the f_0 is estimated from the glottal source signal that is free from the distracting vocal tract resonances [9].

The autocorrelation method [8] was used to estimate the f_0. A range of possible f_0 values is defined based on the speaker's f_0 range in order to reduce gross errors. The voiced-unvoiced decision is made based on the energy of the low frequency band (0–1 kHz) and the number of zero-crossings in the frame. The length of the frame from which the f_0 is estimated is longer than the speech analysis frame in order to estimate the lowest possible f_0 values, as low as 30 Hz. The frames determined as unvoiced are marked as zeros. Parabolic interpolation was used in order to reduce the estimation error due to finite sampling period; a quadratic function is fitted to the peak of the autocorrelation function (ACF) to find the refined f_0 value.

Finally, post-processing is applied to the estimated f_0 trajectory. A repetitive process is applied which consists of 3-point median filtering, filling small unvoiced gaps and removing outlier voiced sections, and detection of unnatural discontinuities based on weighted linear estimation of each individual f_0 estimate from previous and following samples. If the difference between the estimated and the actual values is greater than a specific threshold (based on the mean and variance of the f_0 trajectory), the original value may be replaced with a secondary f_0 estimate from the ACF. This replacement depends on the goodness of the fitting and the relative jump of the original f_0 estimate. An example of extracted f_0 is shown in the top pane of Figure 1.

2.2. Completion of f_0 over unvoiced passages

The wavelet method is sensitive to the gaps in the f_0 contour and therefore, the f_0 contour is completed to yield a continuous f_0 trajectory. Since the wavelet approach aims at connecting the signal to the perceptually relevant information, the linear frequency scale is transformed to the logarithmic semitone scale. A simple linear interpolation method is used. First, smoothed version of the original f_0 was created, and then interpolated over unvoiced passages. The smoothed unvoiced parts are then added to the original f_0 with 3 point median smoothing to reduce discontinuities in voicing boundaries. In addition, to alleviate edge artifacts, constant f_0 was added prior to and after the utterance. The pre-utterance f_0 value was set to the mean f_0 value calculated over the first half (in seconds) of the utterance; the post-utterance f_0 was set to the respective minimum. Finally, the interpolated $\log f_0$ contour is normalized to zero mean, unit variance as required by wavelet analysis. An example of an interpolated pitch contour is depicted in the second pane of Figure 1.

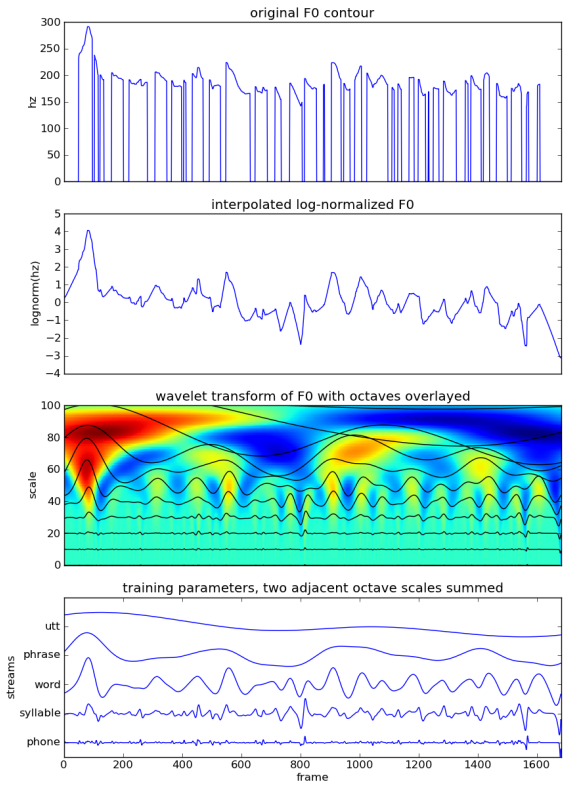

Figure 1: Example of f_0 parameterization. Top pane depicts the baseline method, *base*, in linear frequency scale; the second pane shows the interpolated baseline, *contf0*; third pane shows the continuous wavelet transform of the f_0 signal with the ten chosen scales separated by an octave (method *wave1*); the bottom pane shows the five scales that are merged from the continuous wavelet picture forming the basis of *wave5*

2.3. Wavelet based decomposition of f_0 contour

Wavelet transforms can be used to decompose a signal into frequency components similar to the Fourier transform. Although several alternatives exist, here we have chosen to use continuous wavelet transforms for f_0 decomposition. To define the wavelet transform, consider a (bounded) pitch contour f_0. The continuous wavelet transform $W(f_0)(\tau, t)$ of f_0 is defined by

$$W(f_0)(\tau, t) = \tau^{-1/2} \int_{-\infty}^{\infty} f_0(x) \psi \left(\frac{x - t}{\tau} \right) dx$$

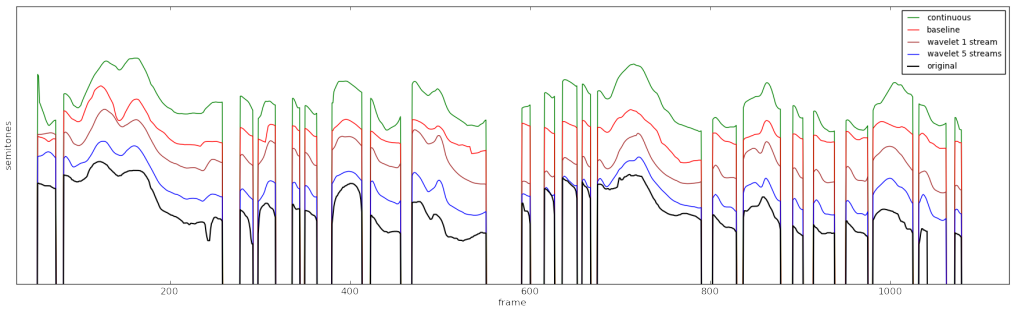

Figure 2: Example of synthesized f_0 contours with evaluated methods on a female corpus test utterance, overlaid three semitones apart.

where ψ is the Mexican hat mother wavelet. The original signal f_0 can be recovered from the wavelet representation $W(f_0)$ by inverse transform (for the proof, see [1, 5]):

$$f_0(t) = \int_{-\infty}^{\infty} \int_{0}^{\infty} W(f_0)(\tau, x) \tau^{-5/2} \psi \left(\frac{t-x}{\tau} \right) dx d\tau.$$

However, the reconstruction is incomplete, if all information on $W(f_0)$ is not availabe. Here, the decomposition and reconstruction is approximated by choosing ten scales, one octave apart. f_0 is represented by the wavelets as ten separate streams given by

$$W_i(f_0)(t) = W(f_0)(2^{i+1}\tau_0, t)(i + 2.5)^{-5/2} \quad (1)$$

where $i = 1, \ldots, 10$ and $\tau_0 = 5$ ms, and the original signal is approximately recovered by

$$f_0(t) = \sum_{i=1}^{10} W_i(f_0)(t) \; + \; \epsilon(t) \quad (2)$$

where $\epsilon(t)$ is the reconstruction error. The reconstruction formula (2) is *ad hoc* and no attempts were made in this stage to optimize the computational efficiency. The accuracy of the reconstruction was evaluated by decomposing and reconstructing ten utterances spoken by a male and a female. The correlation between the original and the reconstructed f_0 signal was $99.7\,\%$ with root mean square reconstruction error of $1.03\ Hz$.

The continuous wavelet transform and ten distinct scales are shown in the third pane of the Figure 1. The scales 0 and 1 correspond to phone level (50 and 25 Hz), scales 2 and 3 correspond to syllable level (6 and 13 Hz), scales 4 and 5 show word level (1.6–3 Hz), scales 6 and 7 correspond to phrase level (0.4–0.8 Hz), and scales 8 and 9 correspond to utterance level. The adjacent scales are combined and shown in the bottom pane of the Figure 1. These five broad scales are separated by two octaves from each other. The correspondance of the prosodic levels of hierarchy and the wavelet scales is approximative and the wavelet scales are not adjusted to optimize the fit. Hence, e.g., not all the syllables have a duration that would fall in the "syllable scale".

3. Constructing the synthesis

3.1. Speech material

In order to carry out evaluation of the proposed f_0 modeling methods, two Finnish HMM-voices were trained, a male and a female one. The male database (MV) used is a traditional synthesis corpus, with rather carefully articulated set of 692 isolated sentences, while the female one (HK) is more diverse, consisting of 600 phonetically rich sentences as well as continuous prosodically rich read speech; 266 long sentences of fact and 607 sentences of diverse prose. 92 sentences of the male database was left out for evaluation purposes and 60 utterances of prose for the female. Both corpora have been tagged for word prominence on discrete scale ranging from 0 to 3, using acoustic features [13]. The prominence labels were used in both training and evaluation as contextual features. Thus the evaluation was not affected by TTS symbolic prosody prediction errors. In addition to word prominence, full context labels were generated with conventional features: quinphones with positional and length features of phones, syllables, word and phrases. Notably, more enriched labeling above word level would have been preferable for the current topic of modeling the prosodic hierarchy.

3.2. Parameterization of f_0 contours

Four different HMM-based statistical models for f_0 generation were compared. Synthesized f_0 contours based on these four and the original sentence f_0 are depicted in Figure 2.

3.2.1. *base*

A standard MSD model for f_0 is trained where each continuous f_0 passage between unvoiced segments is independently generated.

3.2.2. *wave5*

In the model *wave5*, five different f_0 components $w_1, \ldots, w_5$, defined by

$$w_i(t) = W_{2i-1}(f_0)(t) + W_{2i}(f_0)(t),$$

are independently trained by HMMs.

3.2.3. *wave1*

The different time scales correlate especially with their neighbors, so a plausible alternative would be to jointly model all the scales. This is done in *wave1* where one vector $V(t) = \{W_i(f_0)(t)\}_{i=1}^{10}$ contains the time scales.

3.2.4. *contf0*

Since the wavelet based methods *wave5* and *wave1* generate a continuous f_0 trajectory, and since interpolating the pauses in the training data improves the synthesized contours [14], an alternative, *contf0*, is offered where the unvoiced segments are interpolated in the same way as in the preprocessing of the wavelets.

3.3. HMM-training

The speech was parameterized with GlottHMM vocoder [9], yielding a 5-stream HMM structure: vocal tract spectrum LSFs and Gain (31 parameters), voice source spectrum LSFs (10), Harmonic-to-noise ratio (5) and $\log f_0$ (1). f_0 was then processed as described in the previous chapter. 5 streams (1 parameter each) for method wave5, 1 stream (10) for wave1 and one stream for continuous $\log f_0$. The baseline f_0 method was modeled as an MSD stream, others as continuous streams. With dynamic features further added, HMM training was perfomed in a standard fashion using HTS [16]. Stream weights affecting model alignment were set to zero for all streams except vocal tract spectrum LSFs and $\log f_0$. Decision tree clustering was perfomed individually for each stream without stream-dependent contextual question sets. Using the MDL criterion on decision tree building, the wave5 trees tended to become very large compared to baseline. Attempts were made to control the tree size with minimum leaf occupancy count, which was set to 10 on baseline MSD $\log f_0$ stream and 20, 25, 30, 60 and 70 for respective wave5 streams. In addition, MDL factor was set to 0.6 for $\log f_0$ stream and 1.5 for wave5 streams.

4. Evaluation

4.1. Evaluation data

The fundamental frequency parameters of the test utterances were generated from HMMs using original time alignments. For wavelet methods, the f_0 trajectories were constructed from generated scales using Equation (1). Voicing decision for continuous f_0 methods was based on the base MSD stream as well as mean and variance of f_0 for normalized wavelet methods.

The alignments were aquired by force-alignment method with the monophone models estimated during synthesis training. The synthesized sentences were checked manually for gross timing errors, and bad ones were excluded. The final MV test data consisted of 41 isolated utterances, spoken in the same formal style as the training data. By contrast, the HK test utterances consisted of 60 sentences of expressive prose.

4.2. Performance measures

The synthesized f_0 contours were compared to the original f_0 contours, estimated with GlottHMM, by measuring the correlation between the two curves and by calculating the root mean square error for each test utterance. Within an utterance, only the frames that were voiced with all methods were included. Also, due to frequent creaky voice with erratic pitch on original trajectories, the frames where the distance between original and at least one of the synthesized trajectories was more than 8 semitones, were excluded as outliers. It should be noted that these frames were completely excluded from the evaluation so that the comparisons were performed on exactly the same data sets. For the error calculation, the f_0 was converted to semitone scale with base 40 Hz. A Wilcoxon signed rank test was used to assess the statistical significance of the results.

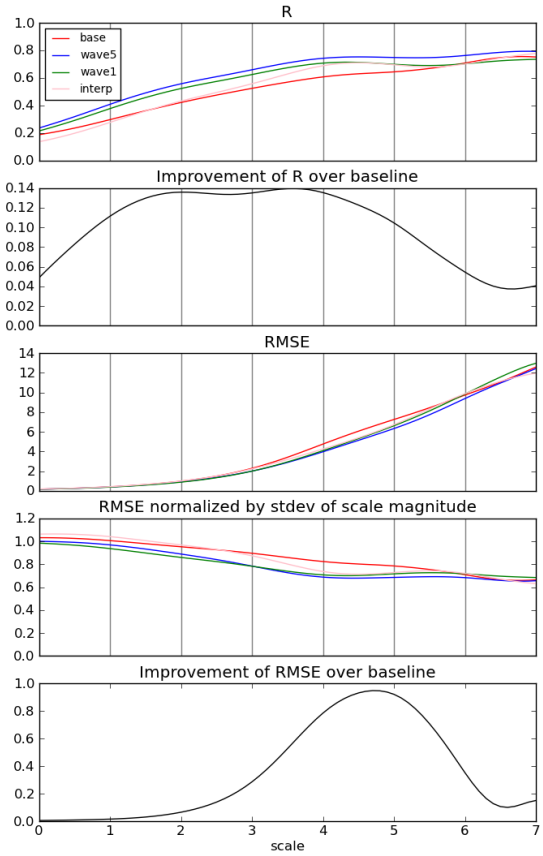

Figure 3: Evaluation results shown scale by scale. The top pane shows the correlations between the four synthesized contours and the original; second pane depicts the difference between the wavelet method *wave5* and *base*; third pane shows the absolute RMSE; in the fourth pane, the values are normalized by the variation at the scale; the bottom pane shows the difference in RMSE between the *wave5* and *base*.

4.3. Performance results

The correlations between the generated f_0 values and original contours showed significantly better performance for wavelet methods than for the baseline for both speakers. For the female data, the correlations over the test utterances were 0.76, 0.72, 0.72, and 0.68 for *wave5, wave1, contf0*, and *base*, respectively, as shown in Table 1. The *wave5* was better than *wave1* ($V = 1298$, $p < 0.05$), better than *contf0* ($V = 1324$, $p < 0.05$) and *base* ($V = 1445$, $p < 0.005$). In addition, the *wave1* was better than *base* ($V = 1329$, $p < 0.05$) but not significantly different

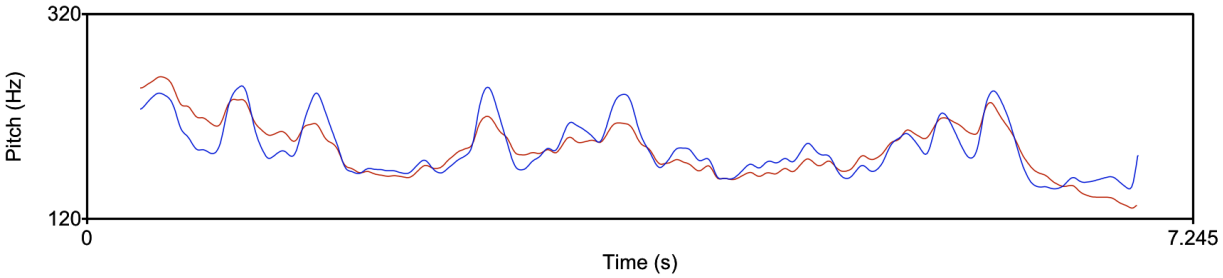

Figure 4: The reconstruction can be weighted to enhance the word level (blue curve) or the phrase level (red curve) intonation.

from *contf0* ($V = 1064$, $p > 0.1$). The *contf0* was marginally better than the *base* ($V = 702$, $p < 0.1$).

The male data showed similar patterns. The correlations over the test utterances were 0.85, 0.84, 0.81, and 0.81, respectively. The *wave5* was marginally better than *wave1* ($V = 288$, $p < 0.1$), better than *contf0* ($V = 129$, $p < 0.001$) and *base* ($V = 88$, $p < 0.001$). In addition, the *wave1* was better than *base* ($V = 136$, $p < 0.001$) and *contf0* ($V = 196$, $p < 0.005$). The *contf0* and the *base* were not significantly different ($V = 439$, $p > 0.1$).

Table 1: *A summary of the performance results of the syntheses. The means of the performance measures for each of the two data sets (female, male).*

	wave5	wave1	contf0	base
corr (F)	0.76	0.72	0.72	0.68
corr (M)	0.85	0.84	0.81	0.81
RMSE (F)	1.38	1.44	1.48	1.53
RMSE (M)	1.57	1.60	1.75	1.76

The root mean square error patterns are similar to the correlation results of the previous paragraphs. For the female data, the root mean square errors were 1.38, 1.44, 1.48, and 1.53 semitones for *wave5, wave1, contf0* and *base*, respectively. The *wave5* outperformed the *wave1* ($V = 1551$, $p < 0.001$), the *contf0* ($V = 1666$, $p < 0.001$, and the *base* ($V = 1781$, $p < 0.001$). The *wave1* and the *contf0* were statistically not different ($V = 1085$, $p > 0.1$), but the *wave1* was better than the *base* ($V = 1419$, $p < 0.005$). The *contf0* was better than *base* ($V = 599$, $p < 0.01$). For the male data, the root mean square error was 1.57, 1.60, 1.75, and 1.76 semitones for *wave5, wave1, contf0* and *base*, respectively. The *wave5* was not different from the *wave1* ($V = 307$, $p > 0.1$) but was better than the *contf0* ($V = 143$, $p < 0.001$) and the *base* ($V = 96$, $p < 0.001$). The *wave1* outperformed both the *contf0* ($V = 206$, $p < 0.005$) and the *base* ($V = 145$, $p < 0.001$). Finally, the *contf0* and *base* did not differ significantly ($V = 433$, $p > 0.1$).

4.4. Temporal scale analysis of the results

In Figure 3, the performance measures over the female test sentences are decomposed to the scale-wise components. Overall, the *wave5* is better than the baselines at all scales. However, the difference is pronounced for the middle scales.

5. Discussion and conclusions

The results of the objective evaluation are in line with previous research. Continuous f_0 modeling is found significantly better than the standard HTS method. On male voice, the synthesis of f_0 is very accurate, suggesting that existing methods are capable of modeling higher level structures to an adequate degree, given consistent style and accurate labels of word prominence. Consequently, the differences between evaluated methods are rather small, though the wavelet based methods provide some gains. As expected, the performance of all evaluated methods is lower on female voice due to difficult test utterances of continuous expressive prose, and also possibly due to more errors in f_0 estimation during analysis. Here, the individually modeled wavelet scales provide a large improvement. However, subjective evaluation is still required for final conclusions.

Overall, the results suggest that the proposed method largely solves the fragmentation problem caused by simultaneous decision tree clustering of all levels of prosodic hierarchy. Yet, somewhat contrary to expectations the improvements seem larger on word level and syllable level than on phrase level. Although technical problems of higher scales affected by boundary effects on wavelet analysis may have an effect, this mainly highlights the need for new contextual features on supra-word level, beyond position and number. With the proposed method the features representing for instance constituent structure, phrase type and utterance modality could actually have an effect on the synthesized prosody.

The wavelet decomposition offers a possibility of adjusting the weights of individual scales prior to reconstruction. This could have potential applications in speaking style modification. For example, informal listening suggested that increasing the weight of the word level makes the synthesized speech sound more resolute and perhaps more intelligible, while listening longer passages is less displeasing when phrase level is emphasized. Moreover, moderate modifications do not seem to have adverse effect on naturalness. Figure 4 presents an example of this type of modification. Local weighting within utterance could also be applied for e.g. emphasis reproduction. Rapid adaptation of speaking style based on transform of the scale weights alone could also be considered.

The current paper has presented a novel method of f_0 modeling based on wavelet decomposition. Many open questions remain. Selection of scales and model structure were made based on intuition alone, no other wavelets beyond mexican hat were considered, neither more popular discrete wavelet transform.

Also, while the proposed method seems quite suitable for the current HMM-synthesis framework, it is deeply unsatisfying to model utterance level f_0 contour with inherently sub-segmental models, when the discrete cosine transform or discrete wavelet transform could represent the level with only a few coefficients.

6. Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement n^{o} 287678 and the Academy of Finland grants 128204 and 125940.

7. References

[1] Daubechies, I., "Ten lectures on wavelets", Philadelphia, SIAM, 1992.

[2] Fujisaki, H., Hirose, K., Halle, P., and Lei, H., "A generative model for the prosody of connected speech in Japanese", Ann. Rep. Eng. Research Institute 30: 75–80, 1971.

[3] Kruschke, H. and Lenz, M., "Estimation of the parameters of the quantitative intonation model with continuous wavelet analysis", in Proc. Eurospeech'03, 4, pp. 2881-2884, Geneva, 2003.

[4] Lei, M., Wu, Y. J., Ling, Z. H., and Dai, L. R., "Investigation of prosodic F_0 layers in hierarchical F_0 modeling for HMM-based speech synthesis", Proc. IEEE Int. Conf. Signal Processing (ICSP) 2010, 613–616.

[5] Mallat, S., "A wavelet tour of signal processing", Academic Press, San Diego, 1998.

[6] Mishra, T., van Santen, J., and Klabbers, E., "Decomposition of pitch curves in the general superpositional intonation model", Speech Prosody, Dresden, Germany, 2006.

[7] Öhman, S., "Word and sentence intonation: a quantitative model", STLQ progress status report, 2–3:20–54, 1967.

[8] L. Rabiner, "On the use of autocorrelation analysis for pitch detection", IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 25, no. 1, pp. 24–33, 1977.

[9] Raitio, T., Suni, A., Yamagishi, J., Pulakka, H., Nurminen, J., Vainio, M., and Alku, P., "HMM-based speech synthesis utilizing glottal inverse filtering", IEEE Trans. on Audio, Speech, and Lang. Proc., 19(1):153–165, 2011.

[10] van Santen, J. P. H., Mishra, T., and Klabbers, E., "Estimating phrase curves in the general superpositional intonation model", Proc. 5th ISCA speech synthesis workshop, Pittsburgh, 2004.

[11] Stan, A. and Giurgiu, M., "A Superpositional Model Applied to F0 Parameterization using DCT for Text-to-Speech Synthesis", Proceedings of 6th Conference on Speech Technologoy and Human-Computer Dialogue (SpeD), 2011, 1–6, Brasov.

[12] Teutenberg, J., Watson, C. I., and Riddle, P., "Modelling and synthesising F0 contour with the discrete cosine transform", ICASSP 2008, 3973–3976, 2008.

[13] Vainio, M., Suni, A., and Sirjola, P., "Accent and prominence in Finnish speech synthesis", Proc. 10th Int. Conf. Speech and Computer (Specom 2005), 309–312.

[14] Yu, K. and Young, S., "Continuous F0 Modeling for HMM based statistical parpametric speech synthesis", Trans. Audio, Speech and Lang. Proc., 19:5, 1071–1079, 2011.

[15] Zen, H. and Braunschweiler, N., "Context-dependent additive log F0 model for HMM-based speech synthesis", Proc. Interspeech 2009: 2091–2094.

[16] Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A., Tokuda, K., 2007. The HMM-based speech synthesis system (HTS) version 2.0. In: SSW6. pp. 294–299.

Continuous wavelet transform for analysis of speech prosody

Martti Vainio, Antti Suni, and Daniel Aalto

Institute of Behavioural Sciences (SigMe Group), University of Helsinki, Finland
martti.vainio@helsinki.fi, antti.suni@helsinki.fi, daniel.aalto@helsinki.fi

Abstract

Wavelet based time frequency representations of various signals are shown to reliably represent perceptually relevant patterns at various spatial and temporal scales in a noise robust way. Here we present a wavelet based visualization and analysis tool for prosodic patterns, in particular intonation. The suitability of the method is assessed by comparing its predictions for word prominences against manual labels in a corpus of 900 sentences. In addition, the method's potential for visualization is demonstrated by a few example sentences which are compared to more traditional visualization methods. Finally, some further applications are suggested and the limitations of the method are discussed.

Index Terms: continuous wavelet transform; speech prosody; intonation analysis; prominence

1. Introduction

The assumption that prosody is hierarchical is shared by phonologists and phoneticians alike. There are several accounts for hierarchical structure with respect to speech melody: In the tone sequence models which interpret the f_0 contour as a sequence of tonal landmarks of peaks and valleys (e.g. [15]) the hierarchy is mainly revealed at the edges or boundaries of units whereas in superpositional accounts (e.g., [13, 6]) it is seen as a superposition of different levels at each point of the contour. The problem with the tone sequence models stems from their phonological nature which requires a somewhat discretized view of the continuous phonetic phenomena. The superpositional accounts suffer, conversely, from the lack of signal based categories that would constrain the analysis in a meaningful way. Both models suffer from being disjointed from perception and require *a priori* assumptions about the utterances.

Wavelets emerged independently in physics, mathematics, and engineering, and are currently a widely used modern tool for analysis of complex signals including electrophysiological, visual, and acoustic signals [5]. In particular, the wavelets have found applications in several speech prosody related areas: The first steps of the signal processing by the auditory periphery are well described by models that rely on wavelets [23, 22, 17]; they are used in a robust speech enhancement in noisy signals with unknown or varying signal to noise ratio, in automatic speech segmentation, and in segregation along various dimensions of speech signal in a similar way as mel-cepstral coefficients [2, 1, 8, 9]; the multiscale structure of the wavelet transform has been taken advantage of in musical beat tracking [19]. The quantitative analysis of speech patterns through wavelets might also be relevant for understanding the cortical processing of speech (e.g. [3, 14, 7]).

In the present paper, we apply the wavelet methods to recorded speech signals in order to extract prosodically important information automatically. Here, only the fundamental frequency of the speech signal is analyzed by wavelets although similar analysis could be performed to any prosodically relevant parameter contour (e.g., the intensity envelope contour or a speech rate contour) or even the raw speech signal itself.

The analysis of intonation by wavelets is not a new idea. Discrete wavelet analysis with Daubechies mother wavelets was the key component in automatically detecting the correct phrasal components of synthesized f_0 contours of the Fujisaki model further developed under the name general superpositional model for intonation proposed by van Santen et al. [21, 12]. Continuous wavelet transforms with Mexican hat mother wavelet have been used for Fujisaki accent command detection by Kruschke and Lenz [10]. Overall, previous work with wavelets and f_0 have been mainly concerned with utilizing wavelets as a part of model development or signal processing algorithm, instead of using the wavelet presentation itself.

In Finnish, the prosodic word is an important hierarchical level and the prominence at that level reveals much of the syntactically and semantically determined relations within the utterances. We have successfully used a four level word prominence in text-to-speech synthesis in both Finnish and English [20] and the automatic detection of word prominence is a prerequisite for building high quality speech synthesis. In relation to both a tone sequence and superpositional accounts the succesfull detection of word prominence would be related to distinguishing the accentedness of the unit as well as the magnitude of the accent.

Using an inherently hierarchical analysis we can do away with a fixed model and try to directly link acoustical features of an utterance to the perceived prominences within the utterance. In order to evaluate the wavelet analysis we calculated CTW based prominences for about 7600 separate words in 900 utterances previously annotated by human labelers and compared various wavelet and f_0 based features with each other. In this paper we first discuss the CWT and its application to f_0 and then show the quantitative evaluation followed by discussion and conclusion.

2. Continuous wavelet transform

The continuous wavelet transform (CWT) can be constructed for any one-dimensional or multidimensional signal of finite energy. In addition to the dimensions of the original signal, CWT has an additional dimension, scale, which describes the internal structure of the signal. This additional dimension is obtained by convolving the signal by a mother wavelet which is dilated to cover different frequency regions [5]. The CWT is similar to the windowed Fourier transform: the CWT describes the time-

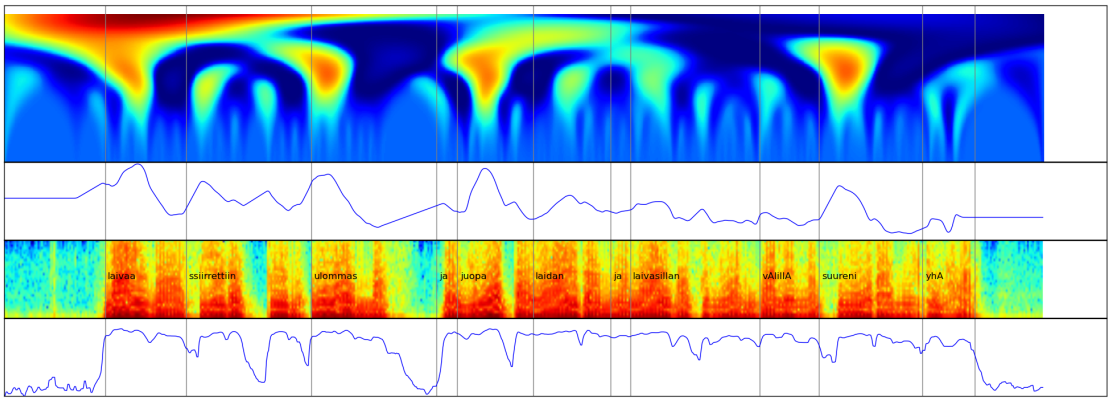

Figure 1: Different analyses aligned temporally. Top pane depicts the continuous wavelet transform with Mexican hat mother wavelet of f_0, second pane shows the interpolated f_0 contour; third pane shows spectrogram of the speech signal; the bottom pane shows gain. The light gray vertical lines show the word boundaries. The text superposed to the third pane transcribes the uttered words (The ship was moved outwards and the gap between the board of the ship and the gangplank got wider, still.)

frequency behaviour of the signal and the signal can be reconstructed from the CWT by inverse wavelet transform. We use here a Mexican hat shaped mother wavelet which corresponds formally to the second derivative of the Gaussian, see pages 76–78 in [11]. In the Figure 1, the top pane shows the CWT of the f_0 contour shown in the second pane. The peaks in f_0 curve show up in the CWT as well, but the size of the peaks in the wavelet picture depends on the local context: the higher at the picture, or in other words, the coarser the scale, the slower the temporal variations and the larger the temporal integration window. Although several hierarchical levels emerge, the quantitative evaluation of the suitability of the CWT to prosodic analysis is only performed on word level. Note that in Finnish, content words have a fixed stress on the first syllable, clearly visible in the Figure 1. The third and fourth panes show the spectrogram and the intensity envelope of the same utterance. The time scales in the wavelet picture range from the 67 Hz as finest to less than 1 Hz as coarsest.

3. Quantitative evaluation

A visualization tool cannot be evaluated quantitatively as a whole. However, if the different temporal scales reflect perceptually relevant levels of prosodic hierarchy, the representation of f_0 at any scale should correlate with judgements of the relative prominence at that particular level. This hypothesis is tested at the level of prosodic word. Although word prominence is signaled by f_0, it is, to large extent, signaled by other means as well including intensity, duration, word order, and morphological marking. Hence, the f_0 based prominence annotation is compared to a simple baseline f_0 prominence annotator and to the labels obtained from phonetically trained listeners.

3.1. Recorded speech data

The evaluation data consisted of 900 read sentences by a phonetically trained, native female speaker of Finnish. Linguistically, the sentences represented three different styles: modern standard scientific Finnish, standard Finnish prose, and phonetically rich sentences covering the Finnish phonemes. The sentences were recorded using high quality condenser microphone in a sound proof studio, digitized, and stored on a computer hard drive. The mean durations of the sentences had average durations of 6.1 s, 3.5 s, and 3.8 s. The total duration amounted to 1h 1 min. Acoustic features were extracted of the utterances with GlottHMM [16], and then the utterances were aligned with the text.

3.2. Fundamental frequency extraction

The fundamental frequency of the test utterances were extracted by GlottHMM speech analysis and synthesis software. In GlottHMM analysis, the signal is first separated to vocal tract and glottal source components using inverse filtering, and the f_0 is then extracted from the differentiated glottal signal using autocorrelation method. Parameters concerning voicing threshold and admissible range of f_0 values were tuned manually for the current speaker. While GlottHMM performs some post-processing on analyzed f_0 trajectories, deviations from perceived pitch remain, particularly in passages containing creaky voice. Thus, f_0 values were first transformed to logarithm scale and then all values lower than 2 standard deviations below the mean of log f_0 were removed.

The unvoiced segments of the speech and the silent intervals make the direct wavelet analysis impossible since f_0 is not well defined for these segments. Hence, the unvoiced gaps were filled using linear interpolation. Additionally, to alleviate edge artifacts, the continuous f_0 contour was extended over the silent beginning and end intervals by replacing the former by the mean f_0 value (logarithmically scaled) over the first half of the completed f_0 contour, and the latter by the mean over the second half. Then the f_0 curve was filtered by a moving average Hamming window of length 25 ms and finally normalized to zero mean and unity variance.

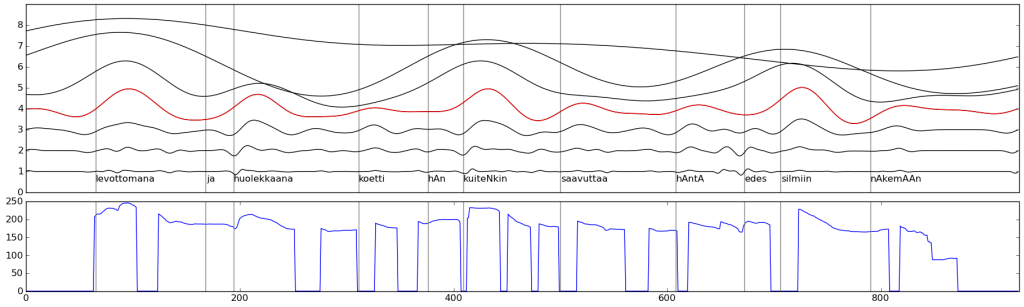

Figure 2: The word prosody scale is chosen from a discrete set of scales with ratio 2 between ascending scales as the one with the number of local maxima as close to the number of words in the corpus as possible. The upper pane shows the representations of f_0 at different scales. The word level (4.2 Hz; see text) is drawn in red. The lower pane shows the f_0 curve. The abscissa shows the frame count from the beginning of the utterance (5 ms frame duration).

3.3. Baseline annotation based on f_0 signal

For each word in the evaluation data, we extracted two common measurements from the preprocessed and normalized f_0 signal, the maximum value observed during word (BMax) and the maximum minus minimum (BRange). The measurements were not further processed, despite the scale differences compared to manual annotation, as only correlation was being tested.

3.4. CWT annotation based on f_0 signal

The CWT transform was first perfomed with one scale per octave, with finest scale being 3 frames or 15 ms. Then, the scale of interest for word prominence was selected as the one with positive peak count closest to the number of words (see Figure 2; the word scale corresponds to 4.2 Hz in the current data). This is intuitively suitable for Finnish, with relatively few unaccented function words. Three wavelet based measurements were then extracted for each word, height of the first local maximum (WPeak) as well as the same two measurements as in f_0 baseline (WMax, WRange). If the word contained no maxima, then the prominence of the word was set to zero. Note that the peak method is not applicable to raw F0, as the noisier contour contains many peaks. More complex measurements were experimented with, such as averaging over multiple scales, but with only moderate success.

3.5. Prominence labeling

Ten phonetically trained listeners participated in prominence labeling. The listeners were instructed to judge the prominence of each word in a categorical scale: 0 (unaccented, reduced); 1 (perceivably accented but no emphasis); 2 (accented with emphasis); 3 (contrastive accent). The listeners reported to have based their judgements mainly on listening and secondarily to the available Praat analyses of pitch, intensity, and spectrogram. Every listener labeled 270 sentences in such a way that every sentence was labeled by three listeners. The prominence of a word was set to the average of the three judgements.

3.6. Statistical analysis

The two baseline annotations and the three wavelet based annotations were compared to the listeners' judgements of word prominence by linear regression analysis. The amount of variance explained (R squared) by the regression model was used as an indicator for the goodness of the used measure.

3.7. Results

The baseline measure $BMax$ has a strong correlation to the prominence judgements with 37 % of the variance explained. The other baseline measure $BRange$ explained 36 % of the variance. The wavelet based measures fitted better to the data: $WMax$ and $WRange$ explained 47 % and 39 % of the variance, respectively. The more involved measures $WPeak$ explained 53 % of the variance.

4. Discussion

The results of the evaluation show that it is fairly straightforward to extract prosodically relevant information form the CWT analysis. In this case it was at the level of prosodic word (which in Finnish correponds well with the grammatical word). As can be seen in Figures 1 and 2, there are other levels both above and below the word that are relevant and if discretized, form a hierarchical tree which can be further exploited for instance in text-to-speech synthesis. However, such an analysis is not free of problems. For instance, the temporal scale corresponding to syllables becomes coarser (higher levels in the Figure 1) when the speech slows down, as is the case in e.g. pre-pausally.

What is important to notice here is that the CWT analysis – as applied to the pitch contour – takes into account both the f_0 level and its temporal properties as cues for prominence. Although we only used one level it is the analysis as a whole that we are interested in. As mentioned earlier, the wavelet analysis can be done on any prosodically relevant signal either alone or jointly – although multidimensional may no longer be easily visualizable.

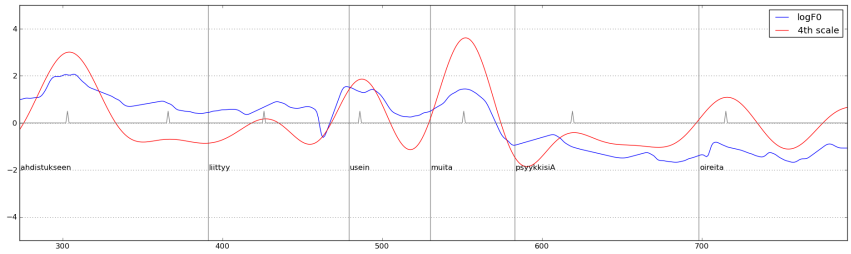

Figure 3: Comparison of selected word scale and original f_0 contour with detected peaks marked with gray triangles. Observe that the wavelet contour is free of noise and declination trend.

5. Conclusion

Continuous wavelet transfrom, a standard mathematical tool for simultaneous analysis and visualization of various temporal scales of a signal, is applied to f_0 signal of recorded speech. At the temporal scale corresponding to prosodic word, the local maxima correlate strongly with the listeners' judgements on the perceived word prominence. This is taken as evidence that the small and large scale contributions induced by segmental micro-prosody and phrasal intonation components are effectively removed by the analysis. Moreover, a hierarchical structure emerges which is easily visible and has similarities with the classical description of prosodic structure through a prosodic tree. Unlike other hierarchical models of prosody, the structure rises directly from the signal with no assumptions on the f_0 model.

Some interesting future directions could include building a 'spectrogram of prosody' -visualization tool combining spectrogram and prosody in the same picture, attempting to discretize the hierarchical structure for higher level applications, applying the decomposed prosodic features for TTS prosody models, studying other prosodic features such as energy by CWT, and, finally, exploring the relationship between the CWT analyses and human auditory processing.

6. Acknowledgments

The research leadgin to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement n^o 287678 and the Academy of Finland (projects 135003 LASTU programme, 1128204, 128204, 125940). We would also like to thank Heini Kallio for collecting the prominence data.

7. References

[1] Alani, A. and Deriche, M., "A novel approach to speech segmentation using the wavelet transform", 5th Int. Symposium on Signal Processing and its Applications, Brisbane, 1999.

[2] Bahoura, M., "Wavelet speech enhancement based on the Teager energy operator", IEEE Signal Processing Letters, 8(1):10–12, 2001.

[3] Bradley, A. P. and Wilson, W. J., "On wavelet analysis of auditory evoked potentials", Clinical neurophysiology, 115:1114-1128, 2004.

[4] Chi, T., Ru, P., and Shamma, S. A., "Multiresolution spectrotemporal analysis of complex sounds", J. Acoust. Soc. Am. 118(2):887–906, 2005.

[5] Daubechies, I., "Ten lectures on wavelets", Philadelphia, SIAM, 1992.

[6] Fujisaki, H., Hirose, K., Halle, P., and Lei, H., "A generative model for the prosody of connected speech in Japanese", Ann. Rep. Eng. Reserach Institute 30: 75–80, 1971.

[7] Giraud, A. and Poeppel, D., "Cortical oscillations and speech processing: emerging computational principles and operations", Nature Neuroscience, 15:511–517, 2012.

[8] Hu, G. and Wang, D., "Segregation of unvoiced speech from nonspeech interference", J. Acoust. Soc. Am., 124(2): 1306–1319, 2008.

[9] Irino, T. and Patterson, R. D., "Segregating information about the size and shape of the vocal tract using a time-domain auditory model: the stabilised wavelet-Mellin transform", Speech Communication, 36:181–203, 2002.

[10] Kruschke, H. and Lenz, M., "Estimation of the parameters of the quantitative intonation model with continuous wavelet analysis", in Proc. Eurospeech'03, 4, pp. 2881-2884, Geneva, 2003.

[11] Mallat, S., "A wavelet tour of signal processing", Academic Press, San Diego, 1998.

[12] Mishra, T., van Santen, J., and Klabbers, E., "Decomposition of pitch curves in the general superpositional intonation model",

[13] Öhman, S., "Word and sentence intonation: a quantitative model", STLQ progress status report, 2–3:20–54, 1967.

[14] Petkov, C. I., O'Connor, K. N., and Sutter, M., L., "Encoding of illusory continuity in primary auditory cortex", Neuron, 54: 153–165, 2007.

[15] Pierrehumbert, J., "The phonology and phonetics of English intonation", PhD Thesis, MIT, 1980.

[16] Raitio, T., Suni, A., Yamagishi, J., Pulakka, H., Nurminen, J., Vainio, M., and Alku, P., "HMM-based speech synthesis utilizing glottal inverse filtering", IEEE Trans. on Audio, Speech, and Lang. Proc., 19(1):153–165, 2011.

[17] Reimann, H. M., "Signal processing in the cochlea: the structure equations", J. Mathematical Neuroscience, 1(5):1–50, 2011.

[19] Smith, L. M. and Honing, H., "Time-Frequency representation of musical rhythm by continuous wavelets", J. Mathematics and Music, 2(2):81–97, 2008.

[20] Suni, A., Raitio, T., Vainio, M., and Alku, P., "The GlottHMM entry for Blizzard Challenge 2012 – hybrid approach", in Blizzard Challenge 2012 Workshop, Portland, Oregon, 2012.

[21] van Santen, J. P. H., Mishra, T., and Klabbers, E., "Estimating phrase curves in the general superpositional intonation model", Proc. 5th ISCA speech synthesis workshop, Pittsburgh, 2004.

[22] Yang, X., Wang, K., and Shamma, S., "Auditory representation of acoustic signals", IEEE Trans. Information theory, 38:824–839, 1992.

[23] Zweig, G., "Basilar membrane motion", Cold Spring Harbor Symposia on Quantitative Biology, 40:619–633, 1976.

End of Appendix.