Text Preprocessing for Speech Synthesis

Conference Paper · January 2006

2 authors, including:

Some of the authors of this publication are also working on these related projects:

Project   Longitudinal study of infant-directed speech using multimodal methods (NKFI-134775) View project

Project   Form and Function of prosodic structure in Hungarian and German View project

Text Preprocessing for Speech Synthesis

Uwe D. Reichel, Hartmut R. Pfitzinger

Department of Phonetics and Speech Communication
University of Munich, Schellingstr. 3, 80799 Munich, Germany
{reichelu|hpt}@phonetik.uni-muenchen.de

Abstract

In this paper we describe our text preprocessing modules for English text-to-speech synthesis. These modules comprise rule-based text normalization subsuming sentence segmentation and normalization of non-standard words, statistical part-of-speech tagging, and statistical syllabification, grapheme-to-phoneme conversion, and word stress assignment relying in parts on rule-based morphological analysis.

1. Introduction

Text preprocessing for English text-to-speech (TTS) synthesis in general consists of the following steps:

Text Normalization This first step subsumes sentence segmentation, tokenizing, and normalization of non-standard words.

For **sentence segmentation** the main problem is the ambiguity of the period, marking sentence boundaries or abbreviations, sometimes even simultaneously (*it is 5 p.m.*). For period disambiguation an identification of abbreviations is needed as well as a disambiguation of capitalized words (proper names vs. sentence initial words, thus words following a sentence boundary period). Complications arise from abbreviations that do not differ from ordinary sentence final words (*no.* also being an abbreviation of *number*) and from the fact that also proper names can occur in sentence initial position. Rule-based systems for heuristic period disambiguation operate on local grammars containing abstract contexts for within-sentence periods and sentence boundaries (Cherry and Vesterman, 1991; Aberdeen et al., 1995). Mikheev's (2002) rule-based segmentation is preceded by capitalized word disambiguation. Machine learning approaches as the decision tree classifier in Riley (1989) use context features such as word lengths, capitalization, and word occurrence probabilities on both sides of the period in question. Current systems achieve an error rate down to less than 1%.

Tokenizing in its simplest form is achieved by splitting the text at white spaces and at punctuation marks, that do not belong to abbreviations identified in the preceding step.

Non-standard words are tokens to be expanded to an appropriate orthographic form before grapheme-to-phoneme conversion. Their normalization includes amongst others number conversion, homograph disambiguation (*Henry* **X**, *Scene* **X**, *Mr.* **X**), expansion of abbreviations and symbols, and appropriate treatment of acronyms (some have to be spelled, others not), and email and URL addresses. A token might be split into several words by these operations. Normalization is a difficult task, since creation of the non-standard word types mentioned above is arbitrarily productive and therefore not to be solved solely by table lookup. Furthermore phonetic realization is highly context dependent, examples are the homographs above or digit strings which can be realized either as numbers, phone numbers or

years. While most of the normalization systems tackle this problem by heuristic disambiguation and expansion rules, e.g. Black et al. (1999), there are also some language modeling and machine learning approaches for normalization subtasks. For example in Sproat et al. (2001) word normalization is amongst others formulated in terms of maximizing the conditional probability of a normalized word sequence given an observed token sequence.

Part-of-Speech Tagging Part-of-Speech (POS) tagging means word class assignment to each token. Its input is given by the tokenized text. Taggers have to cope with unknown words (OOV problem) and ambiguous word-tag mappings. Rule-based approaches like ENGTWOL (Voutilainen, 1995) operate on a) dictionaries containing word forms together with the associated POS labels and morphologic and syntactical features and b) context sensitive rules to choose the appropriate labels during application. In statistical approaches (Jelinek, 1985) generally the most probable tag sequence given the observed word sequence is estimated. In transformation-based tagging (Brill, 1995) a hybrid approach can be found, where disambiguation rules are derived by statistical means.

Grapheme-to-Phoneme Conversion Since hand-crafted rule generation for language processing is very time-consuming and corresponding systems are highly language dependent, most of the current G2P systems are purely data-driven (see e.g. Yvon (1994) for an overview over some machine learning approaches to G2P conversion). Accounting for the influence of morphology and syllable structure can improve performance of G2P conversion (Reichel and Schiel, 2005).

Word Stress The assignment of English word stress relies on phonological, morphological, and word class features. The crucial phonological feature is syllable weight: heavy syllables rather attract stress than weak ones. Amongst the morphological features are affix types (stressed vs. unstressed vs. pre-stressed) and the position within a compound. Word class and word syllable length determine default stress patterns. Metrical phonology (Liberman and Prince, 1977) accounts for lots of these factors and is a useful framework for rule-based approaches to word stress assignment. Among the data-driven approaches are neural networks (Gupta and Touretzky, 1994) predicting stress patterns given syllable weight patterns and

instance-based learning (Daelemans and van den Bosch, 1997) which matches new words against words with an already known stress pattern.

In the following sections our TTS text preprocessing modules are presented.

2. Text Normalization

2.1. Identification of Proper Names, Acronyms and Abbreviations

Since retrieval of proper names, acronyms and abbreviations is crucial for appropriate sentence segmentation and normalization of non-standard words, this task is carried out prior to text normalization. Due to the high productivity of these word classes simple table lookup is insufficient and has to be augmented by following procedures.

Proper names   All those tokens are considered as proper names that occur only and at least twice in capitalized form. Only occurrences in unambiguous environments are counted, that means not behind a period except for periods of prepositional titles like *Mr., Dr.*, etc.

Abbreviations   Token t is identified as an abbreviation, if 1) it has not been classified as a proper name and 2) it ends with a period and 3) one of the following conditions is fulfilled:

- t contains another period (*e.g.*), or
- the string of t preceding the period consists of just one small letter, or
- t contains no vowel (exception *qu.*) and at least one small letter (vs. acronyms, numbers), or
- the letter sequence of t indicates a violation of phonotactics (see below).

Acronyms   Token t is identified as an acronym, if it has not been classified as an proper name or abbreviation and has not been classified as a roman number (using local grammars) and if one of the following conditions holds:

- t consists entirely of consonants, or
- t consists entirely of capitals (except *I*), or
- t is preceded by the article *an* and does not start with a vowel, or
- t is preceded by the article *a* and starts with a vowel (except *u*), or
- the letter sequence of t indicates a violation of phonotactics.

Violation of Phonotactics   The phonotactics exploited here is related to the sonority-based syllable definition according to which a syllable is characterized by a sonority peak facultatively preceded by a rise and followed by a decline of sonority (in case of presence of head and coda, respectively). A letter sequence of a token indicates a violation of phonotactics if 1) the first (resp. last) letter can be associated with a phoneme of higher sonority than that of a fricative (which can occur as a syllable appendix), and 2) the sonority of that phoneme is higher than the phoneme associated with the following (resp. preceding)

letter, and 3) none of the two letters in focus can be associated with a syllable nucleus. For word beginnings vowel letters are associated with syllable nuclei, for word endings also $<m>, <n>$; $<l>$ is not treated as a nucleus associate since syllabic /l/ is represented by *le* in English orthography. Examples: *incl.* is identified as an abbreviation while *wrists.* and *fascism.* are treated as standard words followed by a period.

2.2. Sentence Segmentation

The hand-crafted binary decision tree in Figure 1 guides the decision whether or not token t_i is followed by a sentence boundary. i is ranging over the tokens of the present tokenization of the text at white spaces and unambiguous punctuation. The sentence segmentation completes the tokenization process.
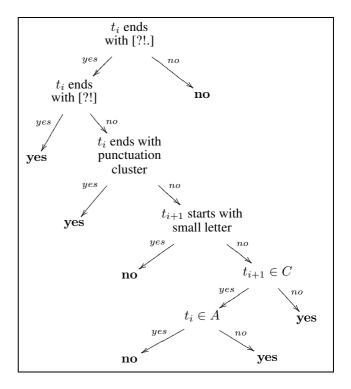

Figure 1: Decision tree for sentence boundary detection (**yes** vs. **no**). A *punctuation cluster* is for example: **".**; $C :=$ {names, titles, numbers, roman numbers}; $A :=$ {abbreviations, postpositional titles}.

2.3. Normalization of non-standard words

For space reasons we present just a selection of our normalization procedures here.

Numbers   In general the following number transformations are carried out: roman numbers are converted to arabic numbers by calculation and arabic numbers are converted to letters by finite state transducers for cardinal and ordinal numbers. The identification of roman numbers and the distinction of cardinals and ordinals is guided by local grammars.

Cardinal numbers are disambiguated whether to be pronounced as one number, as a date, or digit by digit through

pattern matching and examination of the text environment regarding e.g. date-related or phone number cues.

Dates are further completed by prepositions and articles accordingly. E.g. *12 Feb.* becomes *on the twelve of February*, but *on* being omitted if a preposition is already given.

Abbreviations and Acronyms Unknown abbreviations are spelled. Unknown acronyms are spelled if indicated by a preceding indefinite article or by violation of phonotactics (incl. lack of vowels; see above). Otherwise they are pronounced as standard words. This acronym examination also takes place for each part of a hyphenated compound (*CD-Rom*) and within URLs and email addresses.

3. Part-of-Speech Tagging

Our approach for POS tagging described in more detail in Reichel (2005) is statistical and can be seen as a generalization of the classical Markov tagger presented by Jelinek (1985). The $P(w|t)$ emission probabilities of word w given tag t are replaced by a linear interpolation of tag emission probabilities given a list of representations of w, that are connected to automatically derived word suffixes. Since in English language suffixes also store word class information and are observed in the training data with a high probability, the OOV problem can be reduced this way. However, no linguistic knowledge is needed, hence our approach is language independent.

3.1. Basic Form of a Markov POS Tagger

The aim is to estimate the probable tag sequence $\hat{T}$ given word sequence W:

$$\hat{T} = \arg\max_{T} \Big[P(T|W) \Big] \qquad (1)$$

To estimate $P(T|W)$ first a reformulation is needed by applying Bayes Formula, which leads to:

$$\hat{T} = \arg\max_{T} \Big[P(T)P(W|T) \Big] \qquad (2)$$

given that the denominator $P(W)$ is constant. Further two simplifying assumptions are to be made to get reliable counts for the probability estimations:

- Probability of word w_i depends only on its tag t_i.
- Probability of tag t_i depends only on a limited tag history.

The resulting formula is thus:

$$\hat{T} = \arg\max_{t_1 \dots t_n} \Big[\prod_{i=1}^{n} P(t_i|\text{t-history}_i)P(w_i|t_i) \Big] \qquad (3)$$

$\hat{T}$ is retrieved using the Viterbi algorithm (Viterbi, 1967).

3.2. Generalizations of the basic model

First $P(t_i|\text{t-history}_i)$ is replaced by a linearly interpolated trigram model

$$\sum_{j} u_j P(t_i|\text{t-history}_{ij}),$$

j ranging from unigram to trigram tag history. Further w_i is replaced by a list of word representations leading to a reformulation of $P(w_i|t_i)$:

$$\frac{P(w_i)}{P(t_i)} \sum_{k} v_k P(t_i|\text{w-representation}_{ik})$$

applying again Bayes Formula and linear interpolation. Our model is thus given by:

$$\hat{T} = \arg\max_{t_1 \dots t_n} \Big[\prod_{i=1}^{n} \frac{1}{P(t_i)} \sum_{j} u_j P(t_i|\text{t-history}_{ij}) \\ \sum_{k} v_k P(t_i|\text{w-representation}_{ik}) \Big]. \qquad (4)$$

The interpolation weights u_j and v_k are calculated via the EM algorithm (Dempster et al., 1977).

In order to reduce calculation effort in application, just for unknown words the probabilities are calculated for all POS tags. For known words just the POS tags co-occurring with them in the training corpus are taken into consideration.

Our training data comprises 620000 tokens (including punctuation) taken from prose of the 20th century and pre-tagged by the TnT tagger (Brants, 2000) using the Penn tag set (Marcus et al., 1995).

3.3. Word representations

The representation of words seen in the training data is simply the word form. For OOV cases the representation is given by two string suffixes which are determined by Normalized Backward Successor Variety (NBSV). The Successor Variety (SV) of a string is defined as the number of different characters that follow the string in a given lexicon. This concept is adopted from stemming procedures like the Peak and Plateau algorithm of Nascimento and da Cunha (1998). Backward SV means that the SVs are calculated from reversed strings in order to increase the probability to separate linguistically meaningful *suffixes*. In our approach the SVs are weighted with respect to the mean SV at the corresponding string position to eliminate positional effects. The mean SV is highest in the beginning and declines continuously while moving forward in the word string.

The lexicon of reversed words is represented in the form of a trie (cf. Figure 2), in which the SV at a given state is the number of all outgoing transitions. NBSV peaks are treated as morpheme boundaries. Since this method is knowledge free, of course not all of the obtained segments necessarily correspond to linguistic meaningful entities as might be suggested by Figure 2.

4. Grapheme-to-Phoneme Conversion

Our G2P approach is data-driven; as a classifier we use the C4.5 decision tree (Quinlan, 1993). We treat the conversion as a one-to-one mapping from the set of graphemes to the set of phonemes (UK SAMPA). To cope with any n-to-n relation the phoneme set also comprises the empty phoneme as well as phoneme clusters. A canonical pronunciation dictionary containing 61340 entries is used for training and lookup at application time.
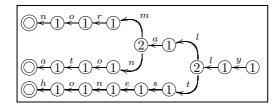
Figure 2: Lexicon trie reversely storing the entries *normally, atonally* and *honestly*. The nodes are labelled according to their SV (not normalized here). The SV peaks correspond to the boundaries of the morphemes *al* and *ly*, respectively.

4.1. Alignment

The first step for creating the grapheme-to-phoneme converter was to align the phoneme string and the orthographic string of each pronunciation dictionary entry. Inspired by the work of Daelemans and van den Bosch (1997) an initial co-occurrence matrix between letters and phonemes was estimated. This was done by diagonally aligning the letters and phonemes of each entry (see Figure 3).
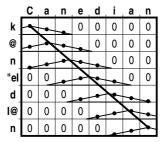

Figure 3: Initial estimation of co-occurrence values for the letters and phonemes of the word Canadian / k @ n "eI d I@ n /. The triangular windows are used to spread the co-occurence probability to adjacent letters.

For each phoneme a triangular window with an area of 1 and a width of 5 letters was centered at the diagonal in order to spread the probability of co-occurence to adjacent letters. The values of the initial co-occurrence matrix are converted into probabilities and used in a Dynamic Programming (DP) algorithm to find the most likely alignment for each pronunciation dictionary entry. The DP algorithm is designed to align either the empty phoneme, ore one phoneme, or a phoneme cluster to each letter.

In order to get a left-aligned phoneme string which is necessary for its alignment with morphologic segments (see below), heuristic post-processing was applied.

4.2. Syllable Segmentation

Since syllable structure influences G2P conversion and is furthermore needed for word stress assignment (see below), syllable segmentation is carried out in advance. Also for syllable segmentation a C4.5 decision tree is trained deciding for each letter whether or not a syllable boundary follows. The current letter as well as the surrounding letters within a window of length 9 are used as features. For model development the same dictionary is used as above, 80% of the entries taken for training and 20% for testing. The resulting decision tree yields a letter error rate of 1.2% and a word error rate of 8.6% on the test data.

4.3. Features

To map a grapheme g on the corresponding phoneme, the decision tree is supplied with 24 features:

- graphemes within a window of length 9 centered at g
- information whether or not a syllable boundary follows for each grapheme within that window
- position of g within the current syllable (head, nucleus, coda)
- type of the current syllable (onset/null onset, open/closed)
- relative position of g within the word
- phoneme history of length 3.

5. Word stress assignment

In our approach word stress is assigned again by a C4.5 decision tree deciding for each syllable whether or not being stressed. Since English word stress is governed by phonology, morphology, and word class (see above) the classifier should be provided by features of all three domains. The phonological features are derived from syllabification and G2P conversion, word class features from POS tagging. To obtain morphologic features some morphologic analysis has to be carried out.

5.1. Morphologic segmentation

The segmentation algorithm we used here is a simplified version of the procedure presented in Reichel and Weilhammer (2004). It consists of two stages: lexicon construction and segmentation. Since it requires some knowledge about affixation it is applicable for different languages just in combination with language dependent stemmers and affix lexica.

5.1.1. Lexicon construction

The lexicon initially comprises English prefixes and suffixes and the linking morpheme '-'. It is then augmented by stems and prefix-stem concatenations of nouns, verbs, adjectives, and adverbs resulting from the application of a slightly modified Porter stemmer (Porter, 1980) for suffix separation. Table 1 shows the morpheme classes of the lexicon entries.

morpheme class	symbol
prefix	prfx
suffix	sfx
linking morpheme	lm
unstemmed word	w
stem	s

Table 1: Morpheme classes. w: word left unchanged by the Porter stemmer.

5.1.2. Segmentation

Each word w is stemmed by the Porter stemmer. Then the stem and the suffix string are further segmented by the function *segmentation* (see Figure 4) the following way:

```
global list morphs := [ ]
function segmentation(str) ≡
    for i:=2 to length(str)-1
        [ prfx, sfx ] := split(str) at position i
        if (prfx ∈ lexicon)
            if (segmentation(sfx) and
                morphotactics_ok(class(prfx), class(first_sfx)))
                    morphs := [prfx, morphs]
                    return 1
            elseif (sfx ∈ lexicon and
                morphotactics_ok(class(prfx), class(sfx))
                    morphs := [prfx, sfx]
                    return 1
            endif
        endif
    endfor
    return 0
```

Figure 4: Algorithm for morphological segmentation

Each input s is recursively divided into string prefixes and suffixes from left to right until a permitted segmentation is achieved or until the end of s is reached. In the course of the recursion, a boundary dividing the current string into prefix and suffix is accepted if 1) the prefix is found in the lexicon, 2) there exists a permitted segmentation for the suffix or (if not) the suffix is found in the lexicon, and just for stem segmentation, 3) the sequence 'prefix class + class of first suffix segment' is not in conflict with simplified English morphotactics as represented by the automaton in Figure 5.
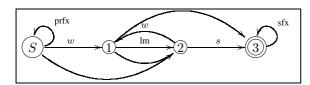

Figure 5: Automaton for simplified English morphotactics. The morpheme classes are explained in Table 1.

On a random test sample of 1000 word types our system yields a word accuracy of 79.6% for completely correct morphologic analysis. Future improvements can be achieved by modifying the Porter stemmer in order to cope with short *ly*-adverbs and comparative adjectives.

5.2. Features

For each syllable s the following features are used for word stress assignment:

- word class
- syllable features
    - syllable weight (reduced, light, heavy)
    - syllable type (onset/null onset, open/closed)
    - word syllable length

- morphologic features (and features derived from morphologic segmentation)
    - class of the morpheme containing the nucleus of s (cf. Table 1). Prefixes and suffixes are further divided into stressed and unstressed affixes (suffixes: also pre-stressed).
    - index of current compound part
    - absolute and relative position of s within whole word and respective compound part
    - only stressable syllable (binary; nucleus in stressed affix or in only stressable morpheme)

Syllable weight is extracted within a 5 syllable window centered on s, morpheme class within a 3 morpheme window centered on the morpheme containing the nucleus of s.

6. Results

Evaluation data taken from the "European Parliament Plenary Session" (EPPS) corpus was provided by ELDA. ELDA also carried out the evaluations, but due to some convention differences (see below) we had to revise the results.

6.1. Text Normalization

Sentence Segmentation End-of-sentence detection was evaluated for 500 sentences. Given two errors the error rate amounts to 0.4%.

Word Normalization The normalization of non-standard words was evaluated for acronyms, for number, time, date, year, and money expressions, as well as for hybrid word forms like e.g. letter-digit combinations. The word error rate for non-standard words adds up to 28.9%.

6.2. Part-of-Speech Tagging

The evaluation data comprises 10000 words extracted randomly from 100000 running words.

Tagset Mapping Different tagsets were used for training and evaluation. Evaluation was carried out using the UK TC-STAR Grammatical POS tagset, but since no appropriate training material was available we worked with the standard PENN tagset (Marcus et al., 1995).
The problem to map from our tagset to the one of TC-STAR was not solely solvable by simple table lookup but was also connected to disambiguation of adjectives and ordinal numbers, of prepositions and subordinating conjunctions, and of auxiliaries and full verbs. Disambiguation was carried out by local grammars. Note that disambiguation was not possible in some cases.

Results After POS mapping and removal of further systematic tagset differences the word error rate amounts 6.5%. Since more tagset inconsistencies are likely, this result has to be taken preliminarily.

6.3. Grapheme-to-Phoneme Conversion

Evaluation was carried out for common words (3808 types), geographic locations (1870 types), and English proper names (2237 types). Due to different treatment of syllabic consonants (marked by "=" by ELDA) we recalculated the error rates after having marked the syllabic consonants from our G2P output accordingly, which is allowed

due to the redundancy of this marking. The overall results including syllable segmentation and word stress placement can be found in Table 2.

Task	Error Rate
Sentence Segmentation	0.4%
Normalization of Non-Standard Words	28.9%
POS Tagging	6.2%
G2P Conversion	
Common Words	6.5%
Geographic Locations	21.1%
Proper Names	17.9%

Table 2: Error rates for the text processing tasks; sentence error rate for sentence segmentation, word error rate otherwise.

7. Discussion

Our submodules for TTS text preprocessing presented here are partly data-driven as for POS tagging, syllable segmentation, and grapheme-to-phoneme conversion and partly rule-based as for text normalization. For word stress assignment we have chosen a hybrid approach using a statistical classifier fed by features partially derived by a rule-based morphologic analysis. In order to improve the modules' adaptabilities to other languages the amount of needed linguistic knowledge should be reduced. Concerning morphology we intend to adopt the automatic induction method used to derive word representations for POS tagging for a complete morphological analysis.

Furthermore it is to investigate if this morphologic analysis could be helpful not only for word stress assignment but also for G2P conversion, for which – being provided with morphological information – an improvement had already been shown for German (Reichel and Schiel, 2005).

Special effort is to be invested in the conversion of geographic location and proper names, for which the results are far away from satisfying.

Due to the tagset inconsistencies, the POS tagging results should be regarded rather as preliminary and recalculated given a unique tagset used for both training and testing.

For G2P conversion it should also be tested if training and test material are created following the same conventions, which is not clear per se due to their different origins.

8. Acknowledgments

9. References

J. Aberdeen, D. Burger, L. Hirschman, P. Robinson, and M. Vilain. 1995. MITRE: Description of the alembic system used for muc-6. In *Proc. MUC-6*, pages 141–155, Columbia, Maryland.

A. Black, P. Taylor, and R. Caley. 1999. The festival speech synthesis system. http://www.cstr.ed.ac.uk/projects/festival.html.

T. Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proc. ANLP-2000*, pages 224–231, Seattle, WA.

E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–566.

L.L. Cherry and W. Vesterman. 1991. Writing tools – the STYLE and DICTION programs. In *4.3 BSD UNIX System Documentation*. University of California, Berkeley.

W. Daelemans and A. van den Bosch. 1997. Language-Independent Data-Oriented Grapheme-to-Phoneme Conversion. In J.P.H. van Santen, R.W. Sproat, J.P. Olive, and J. Hirschberg, editors, *Progress in Speech Synthesis*, pages 77–89. Springer, New York.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statistical Society*, 39(1):1–21.

P. Gupta and D. Touretzky. 1994. Connectionist models and linguistic theory: Investigations of stress systems in language. *Cognitive Science*, 18(1):1–50.

F. Jelinek. 1985. Markov source modeling of text generation. In *The Impact of Processing Techniques on Communications*, NATO ASI series, pages 569–598. Dordrecht: M. Nijhoff.

M. Liberman and A. Prince. 1977. On stress and linguistic rhythm. *Linguistic Inquiry*, 8(2):249–336.

M.P. Marcus, B. Santorini, and M.A. Marcikiewicz. 1995. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330.

A. Mikheev. 2002. Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318.

M.A. Nascimento and A.C.R. da Cunha. 1998. An experiment stemming non-traditional text. In *Proc. SPIRE'98*, pages 74–80, Santa Cruz de La Sierra, Bolivia.

M.F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo.

U.D. Reichel and F. Schiel. 2005. Using morphology and phoneme history to improve grapheme-to-phoneme conversion. In *Proc. Eurospeech*, pages 1937–1940, Lisbon, Portugal.

U.D. Reichel and K. Weilhammer. 2004. Automated Morphological Segmentation and Evaluation. In *Proc. LREC*, pages 503–506, Lisbon, Portugal.

U.D. Reichel. 2005. Improving data driven part-of-speech tagging by morphologic knowledge induction. In *Proc. Advances in Speech Technology AST*, Maribor.

M.D. Riley. 1989. Some applications of tree-based modelling to speech and language indexing. In *Proc. DARPA Speech and Natural Language Workshop*, pages 339–352. Morgan Kaufman.

R. Sproat, A.W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. 2001. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333.

A.J. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

A. Voutilainen. 1995. A syntax-based part of speech analyser. In *Proc. of the Seventh Conference of the European Chapter of the Association for Computation al Linguistics*, pages 157–164, Dublin. Association for Computational Linguistics.

F. Yvon. 1994. Self-learning techniques for grapheme-to-phoneme conversion. Onomastica Research Colloquium, London.