A hybrid approach to grapheme-phoneme conversion

Kay-Michael Würzner and **Bryan Jurish**
Berlin-Brandenburg Academy of Sciences
Jägerstrasse 22-23 · 10117 Berlin · Germany
{wuerzner,jurish}@bbaw.de

Abstract

We present a simple and effective approach to the task of grapheme-to-phoneme conversion based on a set of manually edited grapheme-phoneme mappings which drives not only the alignment of words and corresponding pronunciations, but also the segmentation of words during model training and application, respectively. The actual conversion is performed with the help of a conditional random field model, after which a language model selects the most likely string of grapheme-phoneme segment pairs from the set of hypotheses. We evaluate our approach by comparing it to a state-of-the-art joint sequence model with respect to two different datasets of contemporary German and one of contemporary English.

1 Introduction

Grapheme-to-phoneme conversion (g2p) is the process of converting graphematic representations of words into corresponding phonetic transcriptions. The chief difficulty associated with this task stems from the ambiguity of graphemes with respect to pronunciation. In German for example, the letter 'e' can be realized as stressed, closed, long /eː/ (e.g. ***Me****t*, engl. 'mead'), as stressed, open, short /ɛ/ (e.g. *k****e****ss*, engl. 'perky'), or as unstressed /ə/ (e.g. *Ramp****e***, engl. 'ramp'). In addition, 'e' occurs in diphthongs (e.g. *eu, ei*) or as a length marker (e.g. *ie*) without being overtly pronounced.

Automated g2p is used most prominently in text-to-speech (TTS) systems such as that described by Black et al. (2001), where phonetic transcriptions are estimated from input text to enable subsequent synthesis of a speech signal. More recently, approaches to automatic canonicalization of historical writing systems have made use of phonetic transcriptions as a normal form for identifying spelling variants of a modern word (Jurish, 2010; Porta et al., 2013).

2 Previous Work

g2p implementations can be roughly divided into two types: systems using manually constructed rules and systems based on some statistical model automatically induced from training data.

2.1 Rule-based Approaches

Beginning with the well-known *The Sound Pattern of English* (Chomsky and Halle, 1968, SPE), phonology has been a favorite topic for grammarians, resulting in a large number of phonological descriptions based on transformational rule systems. With Johnson's (1972) proof that rule systems such as those used in SPE are equivalent in power to regular grammars and rewriting systems as long as they do not require cyclic application of rules, finite-state machines became the standard data structure for implementing phonological grammars.

g2p converters based on a manually designed grammar exist for many languages. They have been successfully used in various TTS systems, including `MITalk` (Allen et al., 1987), `gnuspeech` (Hill et al., 1995), and `festival` (Taylor et al., 1998). The biggest problem with hand-written, grammar-based g2p approaches is the expertise and effort required for their production and maintenance. Consider for example `TETOS` (Wothke, 1993), a German g2p system developed at IBM: its grammar consists of about 1,460 rules and the authors admit, "It may also occur that special rules will never be applied" (Heinecke and Wothke, 1992, p. 16).

2.2 Statistical Approaches

Statistical or data-driven approaches to g2p are based on the assumption that regularities in the

correspondence between a word's spelling and its pronunciation can be automatically inferred from a set of word+transcription pairs if sufficient appropriate data are available. The main advantage of such approaches is the fact that creating training data by (manually) transcribing word pronunciations is a much simpler task than creating a formal model of word pronunciation rules, and can be performed by non-experts.

Starting with Sejnowski and Rosenberg (1987), a great number of data-driven g2p techniques have been proposed. The interested reader is referred to Reichel et al. (2008) for a competitive comparison of various techniques. Of particular interested in the current context are the works of Bisani and Ney (2008), who present a joint-sequence model which has been praised as "the gold standard in this area" (Novak et al., 2012b, p. 1); and Jiampojamarn and Kondrak (2009), who were the first to use conditional random field models (CRFs, see Sec. 3.2) as an underlying statistical framework.

3 Our Approach

`gramophone` combines a small set of manually constructed rules with a statistical model induced from a training set of pre-transcribed words. The manual contribution constrains the alignment of the grapheme and phoneme levels, with the aim of allowing only transparent and linguistically motivated alignments by for example foregoing free deletion of either grapheme- or phoneme-symbols and reducing the number of errors due to inadmissible alignments produced by "pure" statistical approaches.[1] The remainder of the procedure is similar to existing approaches: grapheme strings are converted to phoneme strings and the transcription pairs are rated according to their probabilities as estimated from frequency distributions extracted from the training set.

3.1 Alignment

Usually, phonetic transcriptions in corresponding data sets are associated with entire words instead of being explicitly aligned at the grapheme (substring) level. Grapheme-phoneme alignment is therefore a fundamental preprocessing step for training a g2p system. The relation between the grapheme- and phoneme-levels is of type $n : m$ with $n, m \in \mathbb{N}$. Many automatic alignment procedures make use of some Levenshtein-like mechanism (Levenshtein, 1966) to simplify the aforementioned relation to the more tractable case of $n, m \in \{0, 1\}$ (Reichel, 2012; Novak et al., 2012a). Alternatively, string-to-string alignment estimation algorithms have been proposed (Jiampojamarn et al., 2007; Bisani and Ney, 2008).

The alignment scheme proposed here is inspired by Black et al. (1998), and may be described as *constraint-based alignment*: given a grapheme alphabet Σ_{G}, a phoneme alphabet Σ_{P}, and a finite set $M \subset (\Sigma_{\mathrm{G}}^{+} \times \Sigma_{\mathrm{P}}^{+})$ relating grapheme substrings and their potential phonemic realizations, (a) words and their transcriptions are aligned for subsequent model training, and (b) admissible segmentations of words into grapheme-substrings are generated for runtime transcription. The alignment step is implemented using finite-state transducers (FSTs). An FST is a labeled directed graph $T = \langle Q, \Sigma, \Gamma, q_0, F, \delta \rangle$ with a set of states Q, the input alphabet Σ, the output alphabet Γ, the initial state q_0, a set of final states F, and a transition relation $\delta \subseteq Q \times Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \cup \{\varepsilon\}$. Given δ, we may define an extended transition relation δ^* such that $\delta \subseteq \delta^*$, $\forall q \in Q((q, q, \varepsilon, \varepsilon) \in \delta^*)$ and $\forall q, r, s \in Q\ \forall x, y \in \Sigma^*\ \forall a, b \in \Sigma((q, r, x, y) \in \delta^* \wedge (r, s, a, b) \in \delta) \rightarrow (q, s, xa, yb) \in \delta^*)$. Elements from δ^* are called *paths* in T.

Starting from the given set of admissible mappings M, we create an FST E, which we henceforth call the *editor* for M. For each mapping $(g, p) \in M$, a path $(q_0, q_0, g \cdot |, p \cdot \_)$ is added to E with q_0 serving as the initial as well as the only final state of E. '|' and '_' are reserved delimiter symbols. Next, we construct FSTs I_{G} and I_{P} which insert the respective delimiter symbol between grapheme and phoneme segments from M into words and phonetic transcriptions, respectively. I_{G} contains a path $(q_0, q_0, g, g \cdot |)$ for every g in the domain of M. It generates all admissible segmentations by inserting the delimiter symbol at the appropriate location(s) on its output tape. Here again, q_0 is the initial as well as the only final state. I_{P} is defined analogously, and contains a path for every element in the codomain of M. Finally, we construct simple letter FSTs W for a word and T for its phonetic transcription. The alignment $A_{W,T}$ is then the result of a series of composition opera-

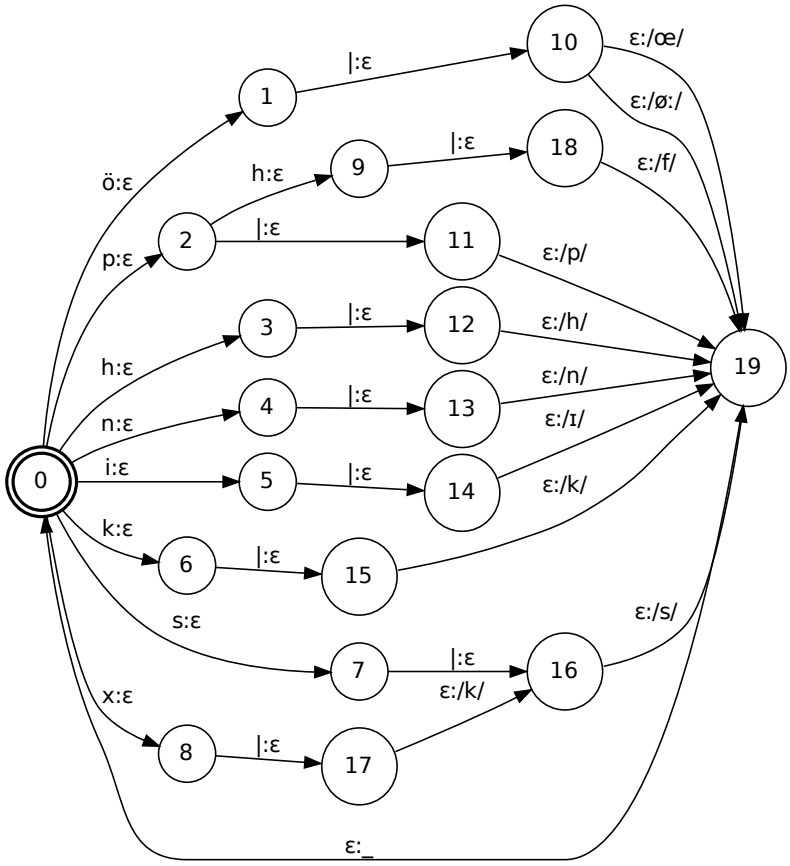
[1] The influence of alignment on the overall performance of g2p systems has been investigated for example by Lehnen et al. (2011).

(a) $M = \left\{ \begin{array}{llllll} \text{p} : /\text{p}/, & \text{h} : /\text{h}/, & \text{ph} : /\text{f}/, & \text{ö} : /\text{ø:}/, & \text{ö} : /\text{œ}/ \\ \text{n} : /\text{n}/, & \text{i} : /\text{ɪ}/, & \text{k} : /\text{k}/, & \text{s} : /\text{s}/, & \text{x} : /\text{ks}/ \end{array} \right\}$

(b) Grapheme Segmentations = { p\|h\|ö\|n\|i\|x\|, ph\|ö\|n\|i\|x\| }

(c) Phoneme Segmentations = { f_ø:_n_ɪ_ks_, f_ø:_n_ɪ_k_s_ }

(d)



(e) Alignment = { ph\|ö\|n\|i\|x\| : f_ø:_n_ɪ_ks_ }

Figure 1: `gramophone` alignment sketch for *Phönix* (engl. 'phoenix'), pronounced /føːnɪks/. Phonemic symbols on the output tape are quoted with slashes '/'. (a) grapheme-to-phoneme segment mapping, (b) grapheme segmentations generated by composition with I_{G}, (c) phoneme segmentations generated by composition with I_{P}, (d) grapheme-to-phoneme segment mapping editor E, and (e) resulting alignment.

tions:[2]

$$A_{W,T} = \pi_2(W \circ I_{\mathrm{G}}) \circ E \circ \pi_2(T \circ I_{\mathrm{P}}) \qquad (1)$$

If multiple admissible alignments were possible – if $A_{W,T}$ contains more than one successful path – then a unique alignment path was chosen randomly.

To generate the admissible segmentations S of a word for subsequent transcription during the application stage, we make use of the first subexpression of Equation 1, repeated below as Equation 2:

$$S = \pi_2(W \circ I_{\mathrm{G}}) \qquad (2)$$

A simple example of a g2p editor is sketched in Figure 1e.

[2]The expression $\pi_2(X)$ in Equations 1 and 2 denotes the 2nd projection (output tape) of the transducer X. We assume that the returned acceptors are treated as identity-transducers in the subsequent compositions.

3.2 Transcription

We treat the transcription stage as an instance of a label assignment task: from the set of known phoneme segments (labels) from M, select for each grapheme segment in an admissible word segmentation (observation) the most likely phoneme segment in the given context. In our case, labeling is performed using a CRF (Lafferty et al., 2001). Such models have become extremely popular in natural language processing (NLP) and have for example been applied to morphological analysis, tokenization, and part-of-speech tagging, in addition to g2p. CRFs can be considered a generalization of probabilistic finite-state automata in the sense that they relax the requirement that each label (state) may depend only on a fixed number of previous labels (states) and the current observation (Rabiner, 1989). In contrast to modeling the *joint* probability of a state and an observation sequence, the *conditional* probability of a state given an observation sequence is modeled (Wallach, 2004).

The CRF is inferred from aligned "graphone" strings – strings of (*grapheme-substring*, *phoneme-substring*) pairs – given a set of features. These features can be understood as random variables expressing the characteristics of an observation. The selection of useful features is a non-trivial task. In the present case, we chose to rely only on the (observable) grapheme context. We treat the size of the available context as a free parameter N, which we refer to as the "order" of the resulting `gramophone` model. Each position i in the input grapheme string $o = o_1 \dots o_n \in \Sigma_{\mathrm{G}}^*$ is assigned a feature for each substring of o of length $m \leq N$ within a context window of $N-1$ characters relative to position i. Formally, a `gramophone` model of order N has $2N^2 - \sum_{m=1}^{N} m$ distinct feature functions f_j^k, where $-N < j \leq k < N$ and $k - j < N$, with $f_j^k(o_i) = o_{i+j} o_{i+j+1} \cdots o_{i+k-1} o_{i+k}$.

The training process is essentially the optimization of the influence (i.e. weights) of the features by maximum likelihood learning. During runtime application, the labeling by the CRF selects the b most probable transcription(s) for each admissible segmentation.

3.3 Rating

The segmented and labeled transcription candidates returned by the CRF transcription phase are then rated using an N-gram language model to determine a univocal "best" transcription for each input word. The model is defined over strings of grapheme-phoneme segment pairs ("graphones"), defining a joint probability for each such string as a product of conditional probabilities under the appropriate Markov independence assumptions. N-gram language models are a standard tool in NLP and can be implemented with (weighted) finite-state techniques (Pereira and Riley, 1997).

The conditional probabilities of the graphone segment N-grams are determined by simply counting graphone substrings of length N and computing their relative frequencies. To ameliorate sparse data problems, some smoothing technique has to be applied. In the present case, we use interpolation (Jelinek and Mercer, 1980) of all k-gram distributions with $1 \leq k \leq N$ in combination with Kneser-Ney discounting (Kneser and Ney, 1995) for treatment of out-of-vocabulary items. Effectively, the rating step selects the most probable word-transcription pair from the set of all previously generated candidates for a given word, as estimated by the graphone N-gram model induced from the training set.

4 Evaluation

We evaluated the system described above on the task of grapheme-to-phoneme conversion for contemporary German and English. German has a rich morphology in terms of word formation processes which are also applicable to foreign words and named entities (e.g. *Versaillesdiktat* /vɛʁzaɪ̯dɪktaːt/, engl. 'Versailles diktat'). Elements of German and foreign pronunciation may thus occur within a single word, which causes finite list-based exception strategies for handling such material to fail. We report the influence of model order on both word and phoneme error rates (WER and PER, respectively) for `gramophone` in comparison to `sequitur`,[3] a freely available implementation of Bisani and Ney's (2008) approach.

4.1 Implementation

Alignment and segmentation procedures were implemented with the help of `OpenFST` (Allauzen et al., 2007). For training and run-time application of CRFs we used the `wapiti` toolkit (Lavergne et al., 2010), employing only unigram feature tem-

[3] `http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html`

plates as described in Section 3.2 and allowing the CRF labeling phase to generate $b = 3$ candidate transcriptions for each admissible segmentation. The final rating of candidate hypotheses was performed using the `OpenGRM` N-gram library (Roark et al., 2012).

4.2 Data

We employ three different data sets for the comparative evaluation of `gramophone` and `sequitur`. The first is the *Bielefeld Lexicon Database VM-II* (Gibbon and Lüngen, 2000, "LexDb"), which contains 76,936 entries. We converted all words to lowercase, removed duplicates and in cases of multiple pronunciation variants for a word selected the first record. 71,481 words and their corresponding phonological representations remained for evaluation purposes.[4]

The second dataset we used for evaluation is based on the pronunciations provided by the German part of the wiktionary project.[5] These pronunciations were created manually using common guidelines,[6] and are encoded using the *International Phonetic Alphabet* (International Phonetic Association, 1999, "IPA"). From the wiktionary XML dump retrieved on 23rd April, 2014, we extracted 148,279 entries which were flagged as German and include a phonetic transcription. After conversion to lower-case, removal of duplicates, and exclusion of incomplete pronunciations (e.g. when only an inflectional suffix was transcribed), 147,359 entries remained. Again, we selected the first available record in cases of multiple pronunciation variants for a word. In addition, we performed an extensive manual revision of the data, addressing phenomena such as incorrect IPA realizations, pseudo-words or mis-transcriptions of inflected forms possibly due to "copy & paste" editing. The complete revised list is available for download at `http://kaskade.dwds.de/~moocow/gramophone/de-wiktionary.data.txt`.

Additionally, we used a subset of 73,736 words from the English part of the CELEX database (Baayen et al., 1995) to test `gramophone`'s performance on English data, which has a reputation for being especially problematic for g2p systems. The principle challenge for g2p on English data lies in its comparative lack of phonological transparency compared to languages such as German or Spanish. Our implementation included for example 25 different grapheme patterns associated with the unstressed mid central vowel /ə/, compared to only 5 for German.

4.3 Method

For each dataset, we manually prepared a set of grapheme-phoneme segment mappings M as described in Section 3.1. For the SAMPA-encoded LexDb dataset we enumerated 277 distinct mapping pairs, *versus* 589 such pairs for the IPA-encoded wiktionary dataset and 463 pairs for the CELEX dataset. Each dataset was randomly partitioned into ten chunks of approximately equal size and evaluated by 10-fold cross-validation. For each of the ten training subsets and for each model order N with $1 \leq N \leq 5$, we trained a `sequitur` model of order N and a `gramophone` model using a context window of N grapheme segments for CRF model features and a language model of order N for candidate rating as described in Sections 3.2 and 3.3, respectively. 5% of the training subset was reserved as a development set for testing model convergence criteria for both `sequitur` and CRF model training. Each trained model was applied to the respective disjoint test subset, and both word- and phoneme-error rates were computed for the concatenation of all test subsets.

4.4 Results & Discussion

Evaluation results for `sequitur` and `gramophone` are given in Table 1. The `gramophone` system outperformed `sequitur` for all conditions tested, although the differences between the two systems became less pronounced as model order increased. On the English CELEX dataset with model order $N = 5$ for example, `gramophone` and `sequitur` differed by only 48 phoneme errors and 50 word errors, rendering the methods effectively indistinguishable in this case.

[4] LexDb was also chosen by Bisani and Ney (2008) to evaluate their approach on German. The authors reported excellent results (WER 1.75 %, PER 0.28 %), far better than those reported for other languages or datasets of comparable size. Such remarkable differences may be attributed to a greater transparency of German's orthography (*versus* for example that of English), but may also be due at least in part to the fact that pronunciations in LexDb were to a large extent automatically generated (Lüngen, p.c.), a property which make them a dubious choice for purposes of g2p evaluation.

[5] `http://de.wiktionary.org`

[6] `http://de.wiktionary.org/wiki/Wiktionary:Deutsch/Lautschrift`

		sequitur		gramophone		Δ	
Dataset	**N**	**WER%**	**PER%**	**WER%**	**PER%**	**WER%**	**PER%**
de-LexDB	1	98.21	34.88	86.26	21.37	12.17	38.73
de-LexDB	2	33.12	5.26	27.94	4.18	15.64	20.53
de-LexDB	3	7.26	1.00	4.19	0.55	42.29	45.00
de-LexDB	4	1.80	0.26	1.36	0.19	24.44	26.92
de-LexDB	5	1.18	0.16	1.02	0.14	13.56	12.50
de-wiktionary	1	98.62	43.21	89.77	29.68	8.97	31.31
de-wiktionary	2	59.46	12.40	51.72	9.97	13.02	19.60
de-wiktionary	3	22.78	4.13	18.78	3.31	17.56	19.85
de-wiktionary	4	12.29	2.22	11.03	1.96	10.25	11.71
de-wiktionary	5	9.61	1.74	9.05	1.66	5.83	4.60
en-CELEX	1	98.53	45.18	87.50	31.45	11.19	30.39
en-CELEX	2	67.02	17.77	51.44	12.21	23.25	31.29
en-CELEX	3	30.74	6.71	22.80	4.85	25.83	27.72
en-CELEX	4	13.58	2.81	11.36	2.37	16.35	15.66
en-CELEX	5	8.98	1.88	8.91	1.87	0.78	0.53

Table 1: Evaluation results for `sequitur` and `gramophone`. Relative error-rate reduction values in the rightmost two columns are computed as $\Delta_r = (r_{\text{sequitur}} - r_{\text{gramophone}})/r_{\text{sequitur}}$ for $r \in \{\text{WER}, \text{PER}\}$.

For both `sequitur` and `gramophone`, error rates were substantially higher for wiktionary and CELEX than for LexDB. Taken together with the unusually high accuracy rates for LexDB reported by Bisani and Ney (2008), this phenomenon suggests that the grapheme-phoneme correspondences encoded in LexDB are themselves particularly amenable to machine learning techniques. Given that these data were to a large extent automatically generated, this is not surprising. Nonetheless, it may also be the case that the raw wiktionary data – due to the distributed and collaborative nature of their creation – display less internal consistency than single-source datasets typically created in the context of academic projects. Although we attempted to address and remove such inconsistencies as part of the data preparation process described in Section 4.2, some degree of noise is likely to remain in the wiktionary data.

In general, `gramophone` models learned more quickly than their `sequitur` counterparts of the same order, but the relative improvement tends to decrease as the order of the model increases, particularly with regard to phoneme error-rates. Indeed, the observed differences in transcription accuracy between the two approaches on all three datasets becomes negligible for $N = 5$. In light of these trends, it may well be the case that `sequitur` will "overtake" `gramophone` as model order grows beyond $N = 5$, since `gramophone`'s reliance on a set of manual alignment heuristics would prevent it from discovering a correct transcription whenever the necessary segment mappings are not encoded in its editor, effectively setting an upper bound for `gramophone` transcription accuracy. Lacking any such alignment constraints, `sequitur` would be free to learn the proper transcriptions in such cases.

5 Conclusion & Outlook

We have presented `gramophone`, a hybrid system for grapheme-to-phoneme conversion using a simple set of manually constructed alignment mappings to provide a grapheme-level segmentation of each input word. Each segment is assigned a phoneme-segment label by a conditional random field model, and the resulting graphone strings are passed to an N-gram language model to select the optimal transcription. We tested our approach by comparing it to the `sequitur` system described by Bisani and Ney (2008) on two independent datasets of contemporary German and one of contemporary English.

Our approach outperformed `sequitur` on all conditions tested, although decreasing absolute and relative error reduction rates for `gramophone` with respect to `sequitur` lead in general to only minimal observable differences

for model order $N = 5$. Future work should investigate whether the upper bound imposed by `gramophone`'s reliance on explicit heuristics to provide all admissible segmentations counteracts its performance benefits with respect to pure statistical approaches such as `sequitur` for higher model orders.

We are also interested in determining to what extent the `gramophone` architecture can be simulated using purely (weighted) finite-state means, in particular with the aim of reducing memory-, I/O-, and computation overhead incurred by overgeneration in the alignment phase by means of "lazy" best-path search in weighted transducer cascades (Mohri, 2002; Jurish, 2010a). While CRFs cannot in general be represented as WFSTs, the CRF employed by `gramophone` uses only observable features and thus contains no overt feature-dependency cycles; it is currently unknown to the authors whether a WFST equivalent exists in this case. Similarly, the shift from a transducer-like representation during the labeling phase to a string-of-pairs representation for the rating phase cannot in general be implemented using traditional (W)FSTs, since these do not admit intersection in the general case (Roche and Schabes, 1997). We speculate that since the maximum length of an alignment mapping is finite and determined at compile-time by the finite set of mapping heuristics, an efficient WFST approximation may be possible.

References

Cyril Allauzen, Michael D. Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Twelfth International Conference on Implementation and Application of Automata*, volume 4783 of *LNCS*, pages 11–23. Springer.

Jonathan Allen, M. Sharon Hunnicutt, and Dennis Klatt. 1987. *From Text to Speech: the MITalk system*. Cambridge University Press.

R. Harald Baayen, Richard Piepenbrock, and Léon Gulikers. 1995. *The CELEX Lexical Database (Release 2) [CD-ROM]*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Alan W. Black, Kevin Lenzo, and Vincent Pagel. 1998. Issues in building general letter to sound rules. In *Third ESCA/COCOSDA Workshop on Speech Synthesis*, pages 77–80. International Speech Communication Association.

Alan Black, Paul Taylor, Richard Caley, Rob Clark, Korin Richmond, Simon King, Volker Strom, and Heiga Zen. 2001. The festival speech synthesis system version 1.4.2. Software, Jun.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper and Row.

Dafydd Gibbon and Harald Lüngen. 2000. Speech lexica and consistent multilingual vocabularies. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer.

Johannes Heinecke and Klaus Wothke. 1992. Letter-to-phone rules for German taking into account the morphological structure of words. Technical Report 75.92.03, Heidelberg Scientific Center.

David Hill, Leonard Manzara, and Craig Schock. 1995. Real-time articulatory speech-synthesis-by-rules. In *Proceedings of AVIOS*, volume 95, pages 27–44.

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.

Frederick Jelinek and Robert L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In Edzard S. Gelsema and Laveen N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–397. North-Holland.

Sittichai Jiampojamarn and Grzegorz Kondrak. 2009. Online discriminative training for grapheme-to-phoneme conversion. In *INTERSPEECH-2009*, pages 1303–1306.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 372–379. ACL.

Douglas C. Johnson. 1972. *Formal Aspects of Phonological Descriptions*. Mouton.

Bryan Jurish. 2010. More than words: Using token context to improve canonicalization of historical German. *JLCL*, 25(1):23–40.

Bryan Jurish. 2010a. Efficient online k-best lookup in weighted finite-state cascades. In Thomas Hanneforth and Gisbert Fanselow, editors, *Language and Logos: Studies in Theoretical and Computational Linguistics*, volume 72 of *Studia grammatica*, pages 313–327. Akademie Verlag, Berlin.

Reinhard Kneser and Herman Ney. 1995. Improved backing-off for m-gram language modeling. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513. ACL.

Patrick Lehnen, Stefan Hahn, Andreas Guta, and Hermann Ney. 2011. Incorporating alignments into conditional random fields for grapheme to phoneme conversion. In *International Conference on Acoustics, Speech and Signal Processing*, pages 4916–4919. IEEE.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(1966):707–710.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321–350.

Josef R. Novak, Paul R. Dixon, Nobuaki Minematsu, Keikichi Hirose, Chiori Hori, and Hideki Kashioka. 2012a. Improving WFST-based g2p conversion with alignment constraints and RNNLM n-best rescoring. In *INTERSPEECH-2012*.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012b. WFST-based grapheme-to-phoneme conversion: Open source tools for alignment, model-building and decoding. In *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, pages 45–49. ACL.

Fernando C. Pereira and Michael D. Riley. 1997. Speech recognition by composition of weighted finite automata. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, chapter 15, pages 433–453. MIT Press.

Jordi Porta, José-Luis Sancho, and Javier Gómez. 2013. Edit transducers for spelling variation in Old Spanish. In *Proceedings of the workshop on computational historical linguistics at NoDaLiDa 2013*, NEALT Proceedings Series 18 / Linköping Electronic Conference Proceedings 87, pages 70–79.

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285.

Uwe Reichel, Hartmut R Pfitzinger, and Horst-Udo Hain. 2008. English grapheme-to-phoneme conversion and evaluation. *Speech and Language Technology*, 11:159–166.

Uwe D. Reichel. 2012. PermA and Balloon: Tools for string alignment and text processing. In *INTERSPEECH-2012*.

Brian Roark, Richard Sproat, Cyril Allauzen, Michael D. Riley, Jeffrey Sorensen, and Terry Tai. 2012. The OpenGrm open-source finite-state grammar software libraries. In *Proceedings of the ACL 2012 System Demonstrations*, pages 61–66. ACL.

Emmanuel Roche and Yves Schabes. 1997. Introduction. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, chapter 1, pages 1–65. MIT press.

Terence J. Sejnowski and Charles R. Rosenberg. 1987. Parallel networks that learn to pronounce English text. *Complex Systems*, 1(1):145–168.

Paul Taylor, Alan W. Black, and Richard J. Caley. 1998. The architecture of the Festival speech synthesis system. In *Proceedings of the Third International Workshop on Speech Synthesis*.

Hanna M. Wallach. 2004. Conditional random fields: An introduction. Technical Report MS-CIS-04-21, University of Pennsylvania, Department of Computer and Information Science.

Klaus Wothke. 1993. Morphologically based automatic phonetic transcription. *IBM Systems Journal*, 32(3):486–511.