社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

WFST-駆動G2Pシステムの構築と評価

Novak　Josef†　峯松　信明†　広瀬　啓吉†

† 東京大学大学院情報理工学研究科
E-mail: †novakj@gavo.t.u-tokyo.ac.jp

あらまし　書記素列に対して音素列を推定する課題（書記素音素変換）は、音声認識や音声合成システム構築において重要な位置を占める。特に英語やフランス語などでは、書記素対音素の対応が複雑であるため、OOV に対して音素列を推定するために、高精度な書記素音素変換システムが必要である。本研究では、EM 駆動の多対多アラインメント手法と統計的 *N*-gram モデルを統合し、重み付き有限状態トランスデューサー（WFST）に基づいた書記素音素変換手法を提案する。そして、この手法を用いた新しいオープンソースの WFST 駆動書記素音素変換システムを構築する。本システムは実用的かつ教育的な支援を目的に、WFST フレームワークの上に構築されたものであり、第三者より提供されたコンポーネントにも対応可能である。本稿では、本システムの精度を実験的に検証し、先行研究とほぼ同様の精度を、より短時間の学習時間で実現している。

キーワード　WFST、 G2P、 アラインメント、 N-gram

Evaluations of an Open Source WFST-based Phoneticizer

Josef NOVAK†, Nobuaki MINEMATSU†, and Keikichi HIROSE†

† University of Tokyo, EEIC Department
E-mail: †novakj@gavo.t.u-tokyo.ac.jp

Abstract　This paper describes in detail some recent experiments for an Open-Source, WFST-based Grapheme-to-Phoneme system, Phonetisaurus. The system comprises several loosely coupled components and includes implementations of several G2P alignment algorithms, and simple 3-gram LMs, as well as support for several third-party components. Standard G2P evaluations were also performed on widely available test sets for English. In particular, on the standard Jiang NetTalk test sets, using an EM-based multiple-to-multiple alignment and a standard 6-gram language model with modified Kneser-Ney smoothing, Phonetisaurus performs favorably compared to state-of-the-art benchmarks. We also note that combination of reverse N-gram and forward N-gram models results in a modest performance gain on all the evaluated test sets with respect to just one or the other, and further that the reverse N-gram models consistently outperform the forward N-gram models on all the data sets.

Key words　G2P, WFST, N-gram, alignment

1. Introduction

Grapheme-to-Phoneme conversion (G2P) is an area that has received much attention over the years, particularly for languages like English and French where there is no general one-to-one correspondence between graphemes and phonemes, or the way that words are written compared to how they are pronounced. This inconsistency is largely due to the practice, common in these languages whereby foreign loan words are adopted with their original spelling, or corresponding romanization, while the pronunciation is adapted to the phonetic and phonotactic constraints of the target language. The G2P problem is important in these languages for both Text-To-Speech synthesis (TTS), and Automatic Speech Recognition (ASR). In the former case G2P systems are used to produce likely pronunciation candidates for synthesis systems, while in the latter they are used to provide dynamic vocabulary support for Out Of Vocabulary (OOV) words, and in reverse to provide likely romanizations or spellings for novel phoneme sequences. Although relevant to most languages, the problem is fairly tractable for many, such as Finnish, Korean or even Spanish where the orthographic conventions are consistent and loan word orthography is often forced to adopt the orthographic conventions of the target language rather than the original language of the loan word in question. For example, the Spanish words “béicon” (bacon) and “cóctel” (cocktail),

which are both English loan words, have since been adopted into Spanish, but with spelling conventions that better reflect general grapheme-to-phoneme correspondences in the Spanish language. In contrast, English words like "rodeo" and "adobe" which have been adopted from Spanish maintain their original Spanish-language orthography, but their pronunciation has been adapted to the constraints of English.

In summary, these issues in English have proven to be challenging, and a unified general solution that provides accuracy similar to that achieved by these other more coherent languages has so far proven elusive. This paper introduces another viable approach to this problem, which we believe may also be naturally extendable to similar problems in other languages, for example accent prediction in Japanese. The remainder of the paper is structured as follows: Section 2. summarizes some of the related research in this area, and Section 3. describes the alignment sub-problem and our chosen solution. Section 4. explains the N-gram based pronunciation model and WFST conversion system that we employ. Section 5. provides information on several sets of G2P experiments that we conducted. Finally, Section 6. concludes the paper.

2. Related Work

Grapheme-to-phoneme conversion is the term applied to the process of automatically generating pronunciation hypotheses given an input orthography. It is an important issue for both speech synthesis and automatic speech recognition for English and many other languages. Due to the relative seriousness of the problem in English, much of the related literature has been devoted to evaluating systems on English data, under the assumption that a solution that performs well on English data should also be robust for other languages. The simplest, most low-tech approach to building a G2P conversion system is to manually create a map between phonemes and graphemes, and to add additional manual rules where necessary. While this may be sufficient for some special cases such as the Japanese Kana alphabets, the approach is extremely unwieldy for languages like English or French, both of which exhibit highly irregular orthographic patterns. Much research has focused on data-driven methods for training G2P systems, for example [1], [2], [3], [4], and [5].

The approach of Galescu in [1] presents one of the first examples based on joint grapheme/phoneme units. An advantage of this model is that it may be applied both to G2P problems as well as P2G problems with little modification. In [1] the authors employ a basic 1-to-1 EM-based alignment procedure to align the grapheme and phoneme pairs and a joint N-gram model to model higher level correspondences. This joint N-gram model is also the model that we employ, and it is discussed in greater detail in Section 4..

In [2] the authors compare the performance of a conditional maximum entropy model with a joint maximum entropy N-gram model and a joint maximum-entropy N-gram model augmented with syllabification and found that the simple joint maximum entropy N-gram model provides several advantages.

In [3] the authors employ a more advanced EM-based alignment algorithm which allows the mapping of multiple-grapheme clusters to multiple-phoneme clusters. They combine this with a traditional Hidden Markov Model (HMM) to solve the G2P problem. The same authors followed this up with a further advances in [4] where they employ an online discriminative training framework to model grapheme-to-phoneme correspondences. We employ the multiple-to-multiple alignment algorithm that they describe in the present work.

Finally, in [5] the authors propose a novel algorithm for performing joint sequence estimation as applicable to the G2P problem, and show state-of-the-art results on many of the standard test data sets used in this area. The results from [5] are utilized as a baseline for our G2P experiments.

The basic G2P problem is succinctly formulated in [2] as follows: given a grapheme sequence G, find the phoneme sequence P^* that maximizes $Pr(P|G)$:

$$P^* = \underset{P}{argmax}\ Pr(P|G) = \underset{P}{argmax}\ Pr(G, P) \qquad (1)$$

One approach to modeling this is via a joint source channel model such as that described in [1]. The current work builds on this approach and also employs the WFST framework in a manner similar to [8]. Specifically, given an orthography, $G = (g_1, g_2, ..., g_N)$, and a pronunciation $P = (p_1, p_2, ..., p_N)$ the model is trained to compute:

$$\prod_{k=1}^{N} Pr(< g, p >_k \mid < g, p >_{1,k-1}) \qquad (2)$$

Three improvements have been made to the basic idea outlined in [8]. First the approach has been extended to support the multiple-to-multiple G2P alignment procedure described in [3]. Second, it has been extended to support reverse N-gram models, and third it has been re-written as a modular, open-source project [12].

3. Alignment

In most cases pronunciation dictionaries do not contain grapheme-to-phoneme alignments, thus it is necessary to first align the grapheme and phoneme sequences in a pronunciation dictionary, prior to building a pronunciation model. One approach is to use a simple dynamic programming algorithm such as Needleman-Wunsch [6]. In most previous literature, including [8], a 1-to-1 alignment procedure has been utilized. Instead, in this work we utilized the EM-based multiple-to-multiple alignment procedure detailed in [3] that supports alignments from digraphs such as "SH" to a single phoneme, or the reverse case. This should be advantageous for languages like English, where such mappings occur frequently. Specifically this means that we can model problematic alignment sequences like,

in a way that reflects the actual associations for these particular words. This overcomes the problems presented by double letters

TH	*O*	*R*	*A*	*X*
|	|	|	|	|
θ	O	R	Æ	**KS**

and double phonemes, both of which occur frequently in languages like English and French. The algorithm itself is described in detail in [3], and is reproduced here for the sake of completeness. This algorithm is an extension of a more basic 1-to-1 stochastic transducer also trained with a variation of the EM algorithm and proposed in [7]. A summary of the algorithm from [3] is presented in Algorithm 1.

Algorithm 1 EM-based Many-to-Many Alignment

Input: $x^T, y^V, maxX, maxY$
Output: γ
for all Mapping operations **do**
  $\gamma(z) \coloneqq 0$
for Sequence pair (x^T, y^V) **do**
  $\alpha \coloneqq$ Forward Many-to-Many$(x^T, y^V, maxX, maxY)$
  $\beta \coloneqq$ Backward Many-to-Many$(x^T, y^V, maxX, maxY)$
  if $(\alpha_{T,V} = 0)$ **then**
    return
  for $t = 0...T$ **do**
    for $v = 0...V$ **do**
      if $(t > 0 \wedge DELX)$ **then**
        for $i = 1...maxX$ **such that** $t - i \geqq 0$ **do**
          $\gamma(x^t_{t-i+1}, \epsilon) \mathrel{+}= \frac{\alpha_{t-i,v}\delta(x^t_{t-i+1}, \epsilon)\beta_{t,v}}{\alpha_{T,V}}$
      if $(v > 0 \wedge DELY)$ **then**
        for $j = 1...maxY$ **such that** $v - j \geqq 0$ **do**
          $\gamma(\epsilon, y^v_{v-j+1}) \mathrel{+}= \frac{\alpha_{t,v-j}\delta(\epsilon, y^v_{v-j+1})\beta_{t,v}}{\alpha_{T,V}}$
      if $(v > 0 \wedge t > 0)$ **then**
        for $i = 1...maxX$ **such that** $t - i \geqq 0$ **do**
          for $j = 1...maxY$ **such that** $v - j \geqq 0$ **do**
            $\gamma(x^t_{t-i+1}, y^v_{v-j+1}) \mathrel{+}= \frac{\alpha_{t-i,v-j}\delta(x^t_{t-i+1}, y^v_{v-j+1})\beta_{t,v}}{\alpha_T, V}$
Maximization step(γ)

In Algorithm 1 the approach works as follows. The expectation step counts all possible grapheme-to-phoneme mappings for each grapheme/phoneme sequence pair (x, y). The partial counts are stored in the γ table, and the associated probabilities are stored in the δ table. Here T and V refer to the lengths of x and y respectively and $maxX$, $maxY$ refer to the maximum allowable subsequence length for the graphemes and phonemes respectively. For example $maxX = 3$ would imply that the maximum allowable grapheme/letter subsequence length would be 3 letters. Similarly the $DELX$ and $DELY$ variables determine whether or not deletions or null transitions are allowed for the grapheme and phoneme sequences respectively. Various settings were explored for the $maxX$, $maxY$, $DELX$, and $DELY$ variables, and it was determined experimentally that in most cases a $maxX = 2$, $maxY = 2$, $DELX = True$, $DELY = False$ produced the optimal alignments. One exception

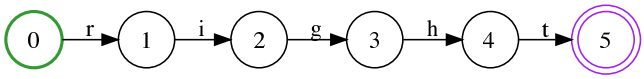

図 1 An example word FSA for the test word 'right', suitable for a system using 1-to-1 alignment.

to this was the NETtalk database and all the test sets based on this database. Here $maxX = 1$ and $maxY = 1$ produced optimal alignment results. The reason for this is most likely due to the much smaller size of the database, as well as the fact that the NETtalk phoneme inventory already includes several diphones.

4. Pronunciation Model

The pronunciation model followed the same general approach that was described in [8]. The pronunciation dictionary was first aligned using the EM-based multiple-to-multiple alignment procedure described in the previous section. The actual pronunciation model was then constructed via the following steps:

（1） Convert each aligned sequence, $(g_1, g_2, ..., g_n)$, $(p_1, p_2, ..., p_n)$ to a sequence of aligned pairs, $(g_1 : p_1, g_2 : p_2, ..., g_n : p_n)$.

（2） Generate an N-gram model from the result of (1)

（3） Convert the N-gram model to a Weighted Finite-State Acceptor

（4） Re-separate the individual grapheme/phoneme pairs into input and output labels respectively, thereby turning the WFSA into a WFST.

Note that in the case of Step 1, a single g_i or p_i may actually correspond to more than one grapheme or phoneme depending on the alignment parameters and either g_i or p_i could conceivably be the empty symbol. However, it is always true that the length of g is equal to the length of p for each sequence pair following alignment. In the case of Step 2, the mit-lm language modeling toolkit was used to generate an N-gram model with $N = 6$. The LMs were generating using modified Kneser-Ney smoothing. In Step 3, the ARPA format language model generated by mit-lm was converted to a WFSA utilizing a modified version of the standard conversion algorithm described in [9]. The algorithm was modified to split the combined grapheme-phoneme input labels into grapheme input labels and and output phoneme labels, thereby constructing a WFST rather than a WFSA. A trivial example of such a pronunciation model WFST is depicted in Figure 3.

Generating a pronunciation for a new word is achieved by compiling the word into a Finite-State Acceptor (FSA) and composing it with the pronunciation model. An example of such a word FSA is depicted in Figure 1 for the test word 'right', in the case of a one-to-one alignment. The case for multi-to-multi alignment is slightly more complicated. In this case, a table was first generated containing all grapheme clusters that were generated by the alignment process. These clusters were then utilized to generate alternative

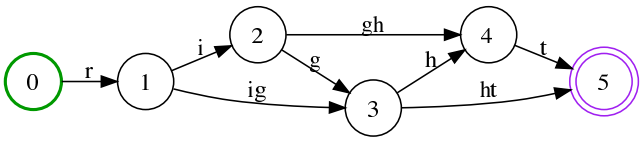

図 2 Another example word FSA for the word 'right', this time suitable for a system using multiple-to-multiple alignment. Whether or not to use a clustered arc, e.g., $i \rightarrow g \rightarrow h$ versus $i \rightarrow gh$ will be determined at runtime by the WFST encoding the pronunciation model.

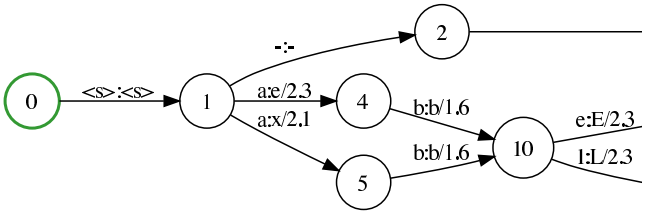

図 3 An example of a WFST pronunciation model. The WFST accepts graphemes (letters) as input and outputs corresponding phoneme sequences. The figure has been simplified to fit the page.

paths through the test FST, where appropriate. The maximum size of the clusters is determined by the alignment parameters, however it is important to note that not all possible clusters will be generated. A possible example of such a test FST for the same word, 'right', is shown in Figure 2.

When the test model is combined with the WFST encoding the joint N-gram model, the model will automatically determine whether it is least costly to opt for the clustered arcs or the non-clustered arcs, based on the context in which they appear and the previously seen training data used to train the model.

4.1 Decoding

The decoding procedure is a multi-step process consisting of a cascade of standard WFST operations and a final formatting step. The process is described in Equation 3.

$$H_{list} = ShortestPath(Det(Proj_o(W \circ M))) \qquad (3)$$

where H_{list} refers to the weighted list of pronunciation hypotheses. W refers to the FSA constructed from the input test word, and M refers to the WFST constructed from the joint G2P N-gram model. The $\circ$ operator denotes composition, the $Proj_o$ operator indicates that the output labels only are projected, thereby creating an FSA containing just the hypothesized phoneme arcs corresponding to the input FSA constructed from the test word. The Det operator refers to determinization, and $ShortestPath$ denotes the global shortest path, or N-shortest paths. Here the best hypothesis is just the shortest path through the composed WFST. A major advantage of this approach is that each component of the final model can usually be trained in a matter minutes. This is in contrast to [5] which often requires many hours to train a model using the same data.

表 1 Results for 6 G2P test sets. Proposed vs. Bisani [5]. Figures in %; (m) means m2m alignment, while (R), (F), and (C) refer to Reverse N-gram, Forward N-gram and Combination system respectively.

Test set	Author	PER	WER
Celex	Bisani [5]	2.5	11.4
	Proposed(Fm)	2.6	12.4
	Proposed(Rm)	2.7	12.2
	Proposed(Cm)	2.6	12.3
OALD	Bisani [5]	3.5	17.5
	Proposed(Fm)	3.6	18.8
	Proposed(Rm)	3.7	18.9
	Proposed(Cm)	3.7	19.0
Pronlex	Bisani [5]	6.8	27.3
	Proposed(Fm)	6.9	28.4
	Proposed(Rm	7.0	28.0
	Proposed(Cm)	6.9	28.0
*NETtalk*15*k*	Bisani [5]	8.3	33.7
	Proposed(F)	***7.3***	34.3
	Proposed(R)	***7.2***	33.9
	Proposed(C)	***7.2***	***33.6***
*NETtalk*18*k*	Bisani [5]	7.8	31.8
	Proposed(F)	***6.8***	31.9
	Proposed(R)	***7.1***	***31.5***
	Proposed(C)	***6.8***	***31.4***
*NETtalk*19*k*	Bisani [5]	7.7	31.0
	Proposed(F)†	***6.6***	***31.0***
	Proposed(R)†	***7.3***	***30.0***
	Proposed(C)†	***6.6***	***31.0***

5. Grapheme-to-Phoneme Experiments

We conducted 7 sets of experiments with the system, utilizing several well-known test sets from the G2P literature. The first set of experiments evaluated Phonetisaurus alone on the popular NETtalk-15k test set, using 3 alignment methods and N-gram orders from 2 to 7. The results of these experiments are depicted in Fig. 4. $n > 6$ resulted in over-fitting, thus n was set to 6 for the remaining experiments. The multiple-to-multiple alignment supported up to 2-2 alignments, but for some test-sets 1-1 alignment performed best.

Further experiments were carried out to compare the proposed system to existing results on other standard test sets. The results from these experiments are summarized in Table 1, where PER refers to Phoneme Error Rate and WER refers to Word Error Rate. PER was calculated according to the standard formula used to compute Word Error Rate for large vocabulary speech recognition: $WER = \frac{S+D+I}{N}$, where S refers to substitions, D refers to deletions, and I refers to insertions. Pronunciation WER on the other hand simply reflects a count of exactly correct hypotheses divided by the

number of test items. Wherever possible the exact same testing conditions as described in [5] and other literature were replicated, however in some cases the exact testing and training data partitions were not available. In the latter case a † indicates that the test partitions were not identical, but that the testing conditions were yet replicated in a manner faithful to that described in the original literature. The performance for the proposed system and [5] system is clearly very similar across all of the experiments. Our experiments also showed that the appropriate alignment procedure depends on the test set. The proposed approach has several advantages from a development standpoint however; first it is highly modular, and second it is very fast. The proposed system required 2m 55s training time for the NETtalk-15k data set whereas the [5] approach required many hours. The proposed system also supports multiple-to-multiple alignment. Finally, the WFST framework ensures a compact representation of the results, including n-best.

5.1 Reverse N-gram models

Following the experiments described above, which we first reported in the forthcoming [13], a simple alternative to the forward N-gram approach was evaluated. This is a backward N-gram training procedure. The motivation for this approach was the simple idea, as yet untested in the related literature as far as we are aware, that grapheme-to-phoneme correspondences, at least in English, depend more heavily on subsequent history features, than on past evidence. The WFST framework made it quite trivial to train and test this idea. The training procedure was set up by simply reversing the grapheme and phoneme sequences in the original training data and running the same two-step multiple-to-multiple alignment procedure followed by the N-gram model generation on this reversed training data. Testing was then conducted by by building and then reversing each FSA for the test data. This yielded a small but significant improvement in the G2P WER performance on all test sets, which is also illustrated in Table 1, by the results marked with the (*R*) identifier. In particular on the NETtalk experiments this pushed the proposed method past that reported in [5] on all counts. Interestingly the improvements in WER were also accompanied by a much smaller but still consistent degradation in the PER scores for all but the *NETtalk15k* experiment. That is, the number of correct pronunciation hypotheses increased, but the average number of phoneme errors in incorrect hypotheses, which includes insertions, deletions and mismatches, increased very slightly. One conceivable explanation for the reduction in WER with the reverse *N*-gram models is that they are more apt at modeling initial vowels. Further analysis is required to determine the underlying cause for the slight increase in PER.

5.2 Model combination

Upon further investigation it was revealed that the errors produced by the forward and reverse N-gram models tend to be complementary. That is, there is a significant subset of words which only one of the two approaches produces correct pronunciation hypotheses. Not surprisingly, the forward N-gram approach tends to be slightly better on average at predicting the *ends* of words, e.g. the "x" (schwa) in "abide (/x b aI d/)" while the reverse N-gram model tends to be more effective at predicting the *beginnings* of words, e.g. the "aI" in "malign (/m x l aI n/)".

A simple linear combination approach, described in Equation 4 was applied to the N-best results for the two model types, with $N = 5$.

$$SC = \frac{1}{S_{Rf}} * S_f + \frac{1}{S_{Rr}} * S_r \quad (4)$$

Here the combined score for a pronunciation hypothesis, SC was calculated as it's inverse rank, denoted by subscript R, within its associated model, denoted by subscript f for forward and subscript r for reverse. The inverse rank was then multiplied by the posterior score attributed to the hypothesis by the pronunciation model, S_f and S_r respectively. The scores for the pronunciation models were not normalized. The pronunciation hypothesis with the best score was then chosen as the answer.

Using this basic linear combination approach the overall WER on the NETtalk-15k test set was further reduced 0.3% absolute, to 33.6%, which is also an improvement over the state-of-the-art joint sequence approach reported in [5] for this dataset. We further note that if only the correct pronunciation hypotheses from the Forward and Reverse models are counted, the WER could potentially be further reduced in the best case to 30.7%. Thus it is reasonable to suppose we can further reduce WER by employing a more intelligent approach to model combination.

6. Conclusion and Future work

In this work we introduced a new, modular open-source phoneticizer, based on the WFST-framework. This G2P system performs favorably in comparison to standard state-of-the-art results on many standard test sets for this research domain. We showed that the WFST paradigm provides for an easy means to rapidly test different ideas, and for applying a G2P system to P2G problems as well. The system is flexible and also supports not just one-to-one alignment methods but also supports multiple-to-multiple alignments a key feature which significantly improves performance on most larger data sets.

In future we plan to extend it with larger array of native alignment implementations and modeling techniques. Preliminary combination experiments showed that there is a clear advantage to combining Reverse and Forward N-gram models, and furthermore that there is still significant room for improvement over our baseline linear combination model. This implies that the method still has further applications. We also note that it should be applicable to accent prediction in Japanese and plan to investigate this in future.

文　献

[1] L. Galescu, et. al., "Bi-directional Conversion Between Graphemes and Phonemes Using a Joint N-gram Model", Proceedings ISCA Tutorial on TTS, 2001.

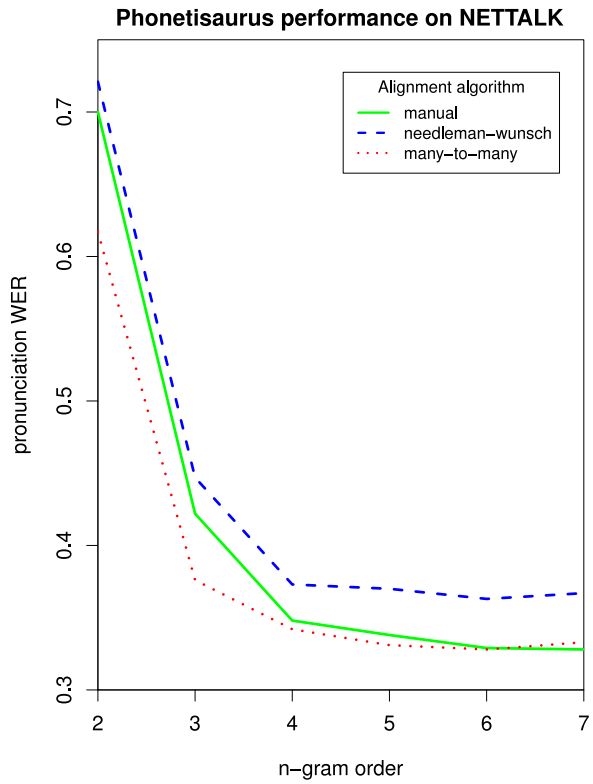

図 4 Results on the NETtalk-15k test set for 3 alignment approaches and n-gram orders 2-7.

[2] S. Chen, et. al., “Conditional and Joint Models for Grapheme-to-Phoneme Conversion”, Eurospeech, 2003, pages 2033-2036.

[3] S. Jiampojamarn, et. al., “Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion”, NAACL HLT, 2007, pages 372-379.

[4] S. Jiampojamarn, et. al., “Online Discriminative Training for Grapheme-to-Phoneme Conversion”, Interspeech 2009, 2009, pages 1303-1306.

[5] M. Bisani, et. al., “Joint-sequence models for grapheme-to-phoneme conversion”, Speech Communication 50, 2008, pages 434-451.

[6] S. Needleman, et. al., “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, Journal of Molecular Biology 48 (3), 1970, pages 443?53.

[7] S. Ristad, et. al., “Learn- ing string-edit distance”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, pages 522-532.

[8] D. Yang, et. al., “Rapid development of a G2P system based on WFST framework”, ASJ 2009 Autumn session, 2009, pages 111-112.

[9] C. Allauzen, et. al., “Conversion of ARPA LMs to WFST”, 2004.

[10] C. Allauzen, et. al., “OpenFST: A General and Efficient Weighted Finite-State Transducer Library”, Proceedings CIAA 2007, pages 11-23.

[11] S. Jiampojamarn, http://code.google.com/p/m2m-aligner

[12] J. Novak, http://code.google.com/p/phonetisaurus

[13] J. Novak, et. al., “Initial Evaluations of an Open-Source WFST-based Phoneticizer”, ASJ 2011 Spring session, 2011, *submitted*.