INFERENCE OF LETTER-PHONEME CORRESPONDENCES USING GENERALISED STOCHASTIC TRANSDUCERS

Robert W.P. Luk[1]

Department of Computing, Hong Kong Polytechnic
Hong Kong

Robert I. Damper

Department of Electronics and Computer Science
University of Southampton, UK

ABSTRACT

This paper describes a special Mealy machine called a generalised stochastic transducer (GST) used to infer letter-phoneme correspondences from a large set of word spelling and their associated phonemic forms. The main idea is to use the transducer to define the most likely alignment for each word and based on the alignment, correspondences are obtained according to the four neighbourhood connectivity. We compared the performance of the inferred correspondences with the manually-derived correspondences, with the inferred correspondences using the delimiting and dynamic programming (DP) techniques and with the GST used directly for translation.

1. MOTIVATION

The delimiting and dynamic programming (DP) techniques [1] (DD) can infer correspondences that can yield similar performance with the manually-derived ones. However, the estimate of the relative index n of the association indices [2] are inaccurate for long words although the delimiting algorithm reduces the length of the word for the DP algorithm before inference. In addition, there is no formal basis for using the delimiting algorithm and the delimited part of the word is defined by the minimum Euclidean distance that has no relation to the translation model. An alternative is to assume that stochastic phonographic transduction [3] is a valid translation model and a stochastic transducer similar to the one in [4] is built to align and subsequently to infer correspondences from the training data. In this way, correspondences are obtained consistent with the translation model.

2. GENERALISED STOCHASTIC TRANSDUCER

The basic idea is to define a simple finite-state transducer (FST) that can align *any* word spelling with the associated pronunciation. Since there are more than one possible alignment, statistical modelling is introduced to define the best alignment as the most likely one. The fast implementation of the Viterbi algorithm determines the ML alignment [5] and using the alignment of the word, we cluster individual letters and phonemes along the ML path to form letter-phoneme correspondences according to the four-neighour connectivity [6]. The inferred correspondences are checked with the existing set of correspondences. If there is identical correspondence in the set, the inferred correspondence is discarded; otherwise it is included in the set.

To infer correspondences, the first task is to estimate the probabilities of GST which may require several passes through the training data. After the probabilities are estimated, each word in the training data is aligned by the GST and correspondences are inferred. When the entire training data are processed, a set of distinct correspondences are obtained.

2.1. Definition

A GST is a stochastic version of a special Mealy machine [7] defined as a sept-tuple $(Q, \Sigma_o, \Sigma_p, \lambda, \mu, q_o, F, \pi)$ where $Q = \{q_o\}$ is a set of states, Σ_o is the input alphabet, Σ_p is the output alphabet, δ is the state transition function, λ is the input-output mapping function, $F = \{q_o\}$ is the set of acceptor states and π is the set of probabilities. The state transition function of GST is defined as:

$$\delta(q_o, a) = q_o$$

where $a \in \{\Sigma_o \cup \{\epsilon\}\}$ and the mapping function is defined as:

$$\lambda(q_o, a) = b$$

where $b \in \{\Sigma_p \cup \{\epsilon\}\}$ given that $\lambda(q_o, \epsilon) \neq \epsilon$. Effectively, we are using a set of correspondences Σ:

$$\{(a, b) | \neg(a = b = \epsilon)\}$$

as the terminals of the phonographic grammar. Building this set of correspondences only need the input and output alphabet which are known in any grapheme-to-phoneme conversion whereas phoneme models [8] have to be specified manually.

2.2. Statistical Models

There are three statistical models that define the probabilities of the most likely alignment. These models are called the independent, hidden Markov and Markov models (first order). The difference between these models is how the conditional probability:

$$p(R^j | R^{j-1}, ..., R^0) = p(\delta^j, \mu^j | \delta^{j-1}, ..., \delta^0, \mu^{j-1}, ..., \mu^0) \quad (1)$$

in the sentential derivations [3] are simplified to. For the independent, hidden Markov and Markov model, equation (1) is reduced to the following three equations respectively:

$$p(R^j | R^{j-1}, ..., R^0) = p(R^j) \quad (2)$$

$$p(R^j | R^{j-1}, ..., R^0) = p(\delta^j | \mu^j) \times p(\mu^j | \mu^{j-1}) \quad (3)$$

$$p(R^j | R^{j-1}, ..., R^0) = p(R^j | R^{j-1}) \quad (4)$$

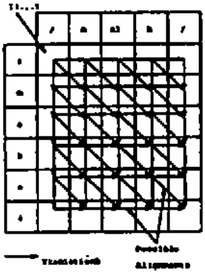

Figure 1: Enumerating all possible alignments on a table. Each alignment is a path on the table from $I(0,0)$ to $I(|\alpha_i|+1,|\beta_i|+1)$.

For the hidden Markov model, we assume the following two conditional independences hold in order to simplify equation (4) to (3):

$$\begin{aligned} p(\delta^{j-1},\mu^j|\delta^{j-1}) &= p(\mu^j|\mu^{j-1}) \times p(\delta^{j-1}|\mu^{j-1}) \\ p(\delta^j,\delta^{j-1},\mu^{j-1}|\mu^j) &= p(\delta^j|\mu^j) \times p(\delta^{j-1},\mu^{j-1}|\mu^j) \end{aligned}$$

We found that the number of correspondences inferred by the GST using the hidden Markov model is small enough for evaluation (i.e. 1082 correspondences) whereas the other two statistical models infer over 1800 correspondences which cannot be evaluated. Thus, in the following discussion, we refer to the GST-inferred correspondences as the set inferred using the hidden Markov model.

2.3. Estimating Probabilities

The conditional probabilities in equation (2,3,4) are estimated from the training data. According to the frequency interpretation of probabilities, we can count the number of times correspondences have been used in the alignments of all the words in the training data, assuming that all possible alignments of a word is equally likely. However, the number of possible alignments of a word N_a grows in a factorial manner with respect to $|\alpha_i|$ and $|\beta_i|$:

$$N_a = \sum_{p=0}^{|\beta_i|} \left(\begin{array}{c} |\alpha_i| \\ |\beta_i| - p \end{array} \right) \left(\begin{array}{c} |\alpha_i| + p \\ p \end{array} \right)$$

An alternative is to use a table to enumerate all the possible alignments and then count the transitions on the table (i.e. the probabilities; Figure 1). The number of updates N_u in this case is quadratic:

$$N_u = 3 \times |\alpha_i| \times |\beta_i| + 4 \times (|\alpha_i| + |\beta_i|) + 5$$

Still another technique is to use dynamic programming to calculate the number of times a particular position $I(x,y)$ is visited according to the following recursive equation:

$$N(x,y) = N(x-1,y) + N(x,y-1) + N(x-1,y-1)$$

where $N(x,y)$ is the number of alignments reached $I(x,y)$ from $I(0,0)$. However, we used the previous method for the initial estimate of the probabilities which incur less processing time and space than the dynamic programming technique.

A better estimate of the probabilities is obtained by re-estimation [3]. However, Luk and Damper [3] showed that the increase in performance in ML translation after two

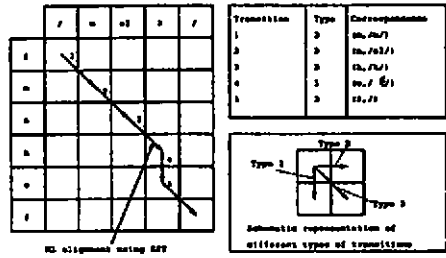

Figure 2: The ML alignment of the word *make* using GST. N
that when deriving correspondences, transition 3 and transi
4 are combined to produce the correspondence (*ke*, /k/).
bottom right diagram shows the schematic representation of
ferent types of transitions. Type 1 and type 2 defines transit
that are connected as defined by the 4-neighbourhood con
tivity. Type 3 transitions represent a break from the follow
transition in the ML alignment.

the increase in performance in ML translation after two
three re-estimation is small and therefore, we limited
number of re-estimation to 2 since each re-estimation ta
a long time.

2.4. Deriving correspondences

The GST can be used directly to translate word spelli
but it is used to infer correspondences in order to eli
nate null-letter correspondences (ϵ,μ_k) and null-phone
correspondences (δ_k,ϵ) where δ_k is a letter string and
is a phoneme string. The former increases the comple
of the Viterbi algorithm and the latter can cause durat
modelling problem [9] when the hidden Markov statis
are used (equation 3).

For each word i in the training data, a table $I(.,.)$ is u
to enumerate all possible alignments of the word spelling
and the associated phonemic form β_i (Figure 2). The
alignment is found using the Viterbi algorithm which be
at $I(0,0)$ and ends at $I(|\alpha_i|+1,|\beta_i|+1)$. Each position I
represents the state reached from $I(0,0)$ which is alway
since there is only one state in Q. There are three type
state transitions in $I(.,.)$ according to δ:

$$\begin{array}{ll} \text{Type 1:} & \{(a,b)|a \in \Sigma_o, b = \epsilon\} \\ \text{Type 2:} & \{(a,b)|a = \epsilon, b \in \Sigma_p\} \\ \text{Type 3:} & \{(a,b)|a \in \Sigma_o, b \in \Sigma_p\} \end{array}$$

These transitions (Figure 2) can be used to define whe
letters and phonemes in the word i are linked togethe
form correspondences. If the letters and phonemes are c
nected along the ML alignment by type 1 or type 2 tra
tions, then they can be grouped to form corresponden
Letters and phonemes are grouped starting from $I(0,0)$
lowing the ML alignment until at $I(x,y)$ where a typ
transition is encountered. At this point, a corresponde
is formed and the next correspondence is derived start
from $I(x+1,y+1)$.

For example, the word (*make*,/meIk/) is aligned in F
ure 2. Based on the ML alignment, the first corresp
dence found is (*m*,/m/). The next correspondence fo
is (*a*,/eI/) and the last correspondence is (*ke*,/k/). N
that the word boundary character at the end of the
alignment is not included because a special corresponde
(#,//) is reserved for it [5].

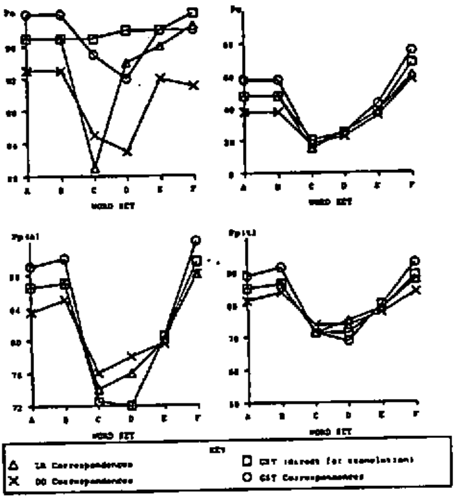

Figure 3: Performance of the Lawrence and Kaye, DD inferred and GST inferred correspondences including the performance of the GST used directly for translation. Note that vertical axis scaling are all different.

Comp	*S*	Median	$H_o(\alpha > 0.1)$
G-L	8.57	0.742	accept
G-D	5.94	3.825	accept
H-G	16.7	1.359	reject
H-L	12.7	1.458	reject
H-D	6.00	3.830	accept

Table 1: Results of Friedman analysis of differences of aggregate performance for each word set. If $S < 9.24$ at $\alpha > 0.1$, H_o is accepted; otherwise H_o is rejected. Key: *Comp* is the different comparison of two sets of correspondences. G-L is the difference between the GST and the Lawrence and Kaye set. G-D is the difference between GST and the DD-inferred set. H-G is differences between the GST-inferred set and the GST used directly for translation. H-L is difference between the GST-inferred set and the Lawrence and Kaye set. H-D is the difference between the GST-inferred set and the DD inferred set.

Comp	*S*	Median	$H_o(\alpha > 0.1)$
G-L	6.20	0.836	marginal
G-D	16.2	4.687	reject
H-G	10.2	2.309	reject
H-L	8.80	3.964	reject
H-D	13.4	7.168	reject

Table 2: Results of Friedman analysis of differences for each performance measure across different word sets. If $S < 6.25$ at $\alpha > 0.1$, we accept H_o; otherwise H_o is rejected. For the G-L case, since $S \approx 6.25$, we accept H_o with reservation.

3. EVALUATION

We evaluate the GST-inferred correspondences by comparing their performance with the Lawrence and Kaye set [10] which are manually derived, the inferred correspondences using the DD techniques [1] and the GST (section 4.1) used directly for translation. The performance is measured from the training data, unseen words, country names, bibliographical names, forenames and novel words as in [1] (denoted as A, B, C, D, E and F respectively in the Tables). Non-parametric statistical tests are applied to determine if there are significant differences in performance between the different sets of correspondences.

3.1. GST for translation

We make a minor modification of the GST when it is used to find the ML translation: i.e. eliminating null-letter correspondences to simplify the task to find the ML translation as in [4] because null-letter correspondences increases the complexity of the Viterbi algorithm and the number of words that cannot be translated because of deleting the null-letter correspondences is small.

Effectively, the GST uses a set of 1172 correspondences (section 2.1). Using 26 orthographic characters and 45 phonemic symbols (i.e. 44 phonemes and an ϵ character), 1170 correspondences are formed. In addition, two correspondences are added for word boundaries and the total number of correspondences becomes 1172.

3.2. Test

Figure 3 shows the alignment and translation performance of the Lawrence and Kaye (L+K) set, the DD-inferred set, the GST-inferred set and the GST used directly for translation (section 3.1). The performances were obtained using the stochastic phonographic transduction scheme.

The GST used directly for translation shows good alignment performance across all the word sets. The alignment performance of the GST-inferred set is excellent for training, unseen and novel words and P_a is consistently better than the DD-inferred set ($\approx$ 8%). The alignment performance of the L+K set is lower than the GST-inferred set except for bibliographical names.

In general, the GST-inferred correspondences yield good translation performance for training, unseen and novel words. The mean perceptual error is around 10% and for novel words, it is as low as 8%. The corresponding word translation accuracy is around 57% for training and unseen words. For novel words, P_c is as high as 75%. The GST used directly for translation yield similar performance with the L+K set for training, unseen and novel words.

For country and bibliographical names, the GST-inferred correspondences and the GST used directly for translation yield lower performance than the other two sets of correspondences. The difference in $P_{p(a)}$ is particularly pronounced for these two word sets (i.e. C and D).

3.2.1. *Statistical Tests*

Friedman analysis is carried out if there are any statistically significant differences in all the performances for each word set. Table 1 shows the results of aggregate differences in performance for each word set. Only, the difference in performance between GST-inferred set and GST, and between GST-inferred set and L+K set are significant. Friedman analysis was also carried out differences for each performance measure across all the word sets. All of the comparisons are significant except the GST versus L+K set which is only marginal.

The Wilcoxon signed ranks test was used to determine individual differences in all the performances of two sets of correspondences (i.e. H-G and H-L in this case) for each

Word Set	*Comp*	W^+	Median	$H_o(\alpha > 0.1)$
A	H-G	10	3.90	marginal
B	H-G	10	3.51	marginal
C	H-G	0	-1.08	accept
D	H-G	3	-1.66	accept
E	H-G	9	0.46	accept
F	H-G	6.5	2.53	accept
A	H-L	10	4.01	marginal
B	H-L	10	3.69	marginal
C	H-L	8	1.68	accept
D	H-L	1	-2.83	accept
E	H-L	10	1.88	marginal
F	H-L	9	4.18	accept

Table 3: Results of using Wilcoxon signed ranks test to determine whether there are significant differences in all the performance measures of two sets of correspondences for each word set. The Wilcoxon statistics are found only for H-G and H-L because H_o is accepted for the other comparisons (table 1). If $W^+ < 11$ at $\alpha > 0.1$, we accept H_o; otherwise H_o is rejected.

word set (table 3). In general, the GST-inferred correspondences were better ($\approx$ 3.5% to 4%) than the GST used for translation and the L+K set in P_a and P_c, only marginally significantly. In addition, the GST-inferred correspondences yield marginally better $P_{p(a)}$ performance ($\approx$ 1.88%) than the L+K set.

The Wilcoxon test was also applied to determine if there are significant differences in each performance measure across all the word set between two sets of correspondences (table 4). In general, there are no significant differences in $P_{p(a)}$ and $P_{p(t)}$ between any comparisons *Comp* although the correspondences inferred using GST is marginally better than the GST used directly for translation with a higher median $P_{p(a)}$ of 1.46%. The GST-inferred correspondences and the GST used for translation have better alignment performance than L+K and the DD-inferred correspondences. In terms of word translation accuracy, the GST-inferred correspondences yield better performance than the L+K set (i.e. 7%) and the GST used for translation (i.e. 5.4%) which in turn is better than the DD-inferred set.

Perf	*Comp*	W^+	Median	$H_o(\alpha > 0.1)$
P_a	G-L	21	2.33	reject
P_a	G-D	21	8.08	reject
P_a	H-G	9	-0.68	accept
P_a	H-L	18	2.33	marginal
P_a	H-D	21	8.06	reject
P_c	G-L	17	1.02	accept
P_c	G-D	21	6.22	reject
P_c	H-G	20	5.37	reject
P_c	H-L	21	7.01	reject
P_c	H-D	15	11.6	accept
$P_{p(a)}$	G-L	11	0.025	accept
$P_{p(a)}$	G-D	10	-0.43	accept
$P_{p(a)}$	H-G	18	1.46	marginal
$P_{p(a)}$	H-L	14	1.1	accept
$P_{p(a)}$	H-D	13	0.99	accept
$P_{p(t)}$	G-L	15	0.45	accept
$P_{p(t)}$	G-D	18	1.62	accept
$P_{p(t)}$	H-G	15	1.35	accept
$P_{p(t)}$	H-L	15	2.35	accept
$P_{p(t)}$	H-D	16	2.28	accept

Table 4: Results of using Wilcoxon signed ranks test to determine whether there are significant differences in performance across all the word sets for two sets of correspondences. If $W^+ < 19$ at $\alpha > 0.1$, we accept H_o; otherwise H_o is rejected.

4. CONCLUSION

An algorithm that infers correspondences using a generalised stochastic transducer is described. The inference process is consistent with stochastic phonographic transduction and correspondences are obtained on the basis of reducing translation complexity and the reduction of duration-modelling problem. The inferred correspondences have high alignment and translation performance for training, unseen and novel words. For training and unseen word, the mean perceptual error is moderately lower (i.e. $\approx$ 2%) than the connectionist model for British RP [11,12]. For novel words, the mean perceptual error is as low as 8% and the corresponding word translation accuracy is 75%. However, for proper names, the GST-inferred correspondence has slightly lower alignment and translation performance than others. The inferred correspondences yield better word translation accuracy than the GST used directly for translation.

REFERENCES

[1] Luk, R.W.P. & Damper, R.I. (1992) "Inference of letter-phoneme correspondences using delimiting and dynamic time warping techniques", *IEEE ICASSP '92, vol 2,* San Francisco, USA, pg. II.61-II.64.

[2] Luk, R.W.P. & Damper, R.I. (1991) "A novel approach to inferring letter-phoneme correspondences", *IEEE ICASSP '91, vol 2,* Toronto, Canada, 741-744.

[3] Luk, R.W.P. & Damper, R.I. (1991) "Stochastic transduction for English text-to-phoneme conversion", *Eurospeech '91, vol 2,* Genoa, Italy, 779-787.

[4] Parfitt, S.& Sharman, R.A. (1991) "A bidirection model of English pronunciation", *Eurospeech '91, vol 2,* Genoa, Italy, 801-804.

[5] Luk, R.W.P. & Damper, R.I. (1992) "A modification of the Viterbi algorithm for stochastic phonographic transduction", to appear in *ICSLP '92,* Alberta, Canada.

[6] Gonzalez, R.C. & Wintz, P. (1987) *Digital Image Processing,* Reading, MA: Addison Wesley.

[7] Mealy, G.H. (1955) "A method for synthesizing sequential circuits", *Bell System Technical Journal,* **34**, **5**, 1045-1079.

[8] Van Coile, B. (1991) "Inductive learning of pronunciation rules with the DEPES system", *IEEE ICASSP '92, vol 2,* Toronto, Canada, 745-748.

[9] Holmes, J.N. (1988) "Speech synthesis and recognition", Wokingham, UK: Van Nostrand Reinhold.

[10] Lawrence, S.G.C. & Kaye, G. (1986) "Alignment of phonemes with their corresponding orthography", *Computer Speech and Language,* 1, 153-165.

[11] McCulloch, N., Bedworth, M. & Bridle, J. (1987) "Netspeak: a re-implementation of NETtalk", *Computer Speech and Language,* 2, 289-301.

[12] Ainsworth, W.A. & Pell, B. (1989) "Connectionist architecture for text-to-speech", *Eurospeech '89, vol 1,* Paris, 125-128.