Proceedings of the International Conference on Computer and Information Science and Technology
Ottawa, Ontario, Canada, May 11 – 12, 2015
Paper No. 100

Model Prioritization Voting Schemes for Phoneme Transition Network-based Grapheme-to-Phoneme Conversion

Seng Kheang, Kouichi Katsurada
Toyohashi University of Technology
Toyohashi city, Aichi prefecture, Japan
kheang@vox.cs.tut.ac.jp; katsurada@cs.tut.ac.jp

Yurie Iribe
Aichi Prefectural University,
Nagakute city, Aichi prefecture, Japan
iribe@ist.aichi-pu.ac.jp

Tsuneo Nitta
Waseda University,
Shinjuku, Tokyo, Japan
tsuneo_nitta@aoni.waseda.jp

Abstract– Performance of the automatic transcription of out-of-vocabulary (OOV) words into their corresponding phoneme sequences has been difficult to get improved because using a single approach does not suffice to cover most of the problems existing in grapheme-to-phoneme (G2P) conversion. Therefore, we employ a novel phoneme transition network (PTN)-based architecture for G2P conversion that allows various approaches to be combined to treat different kinds of related problems simultaneously. This proposed approach first uses different approaches to convert an input word into various phoneme sequences. Second, it generates a confusion network from these obtained sequences and then applies our proposed model prioritization voting algorithm for selecting the best scoring phoneme sequence from the generated PTN sequence. Evaluation results using the CMUDict corpus show that the proposed approach achieves higher word accuracy than previous baseline approaches ($p < 0.005$).

Keywords**:** grapheme-to-phoneme conversion, multiple approaches combination, phoneme transition network (PTN), model prioritization voting schemes.

1. Introduction

The automatic phoneme prediction of arbitrary text, usually known as grapheme-to-phoneme (G2P) conversion, plays an important role in speech synthesis system because the knowledge relating to the process of word reading instead of the orthographic representing of the word is required.

Over the last few years, many well-known data-driven approaches such as the G2P conversion based on Hidden Markov Model (Ogbureke et al., 2010), joint-sequence models (Bisani et al., 2008), Weighted Finite-State Transducer (WFST) (Novak et al., 2012), have been proposed with good accuracy. However, in terms of performance improvement, it seems very difficult and limited to use a single approach to deal with a variety of problems existing in G2P conversion because each approach was designed using different techniques to address different challenges (Kheang et al., 2014b).

Therefore, inspired by Furuya et al. (2012) and Kanda et al. (2013), in this paper, we present a novel phoneme transition network (PTN)-based G2P conversion that allows many different approaches to be applied together to possibly solve different kinds of related problems. First, it converts a target word into many phoneme strings using various data-driven approaches: a multi-layer artificial neural network (ANN) using both grapheme and phoneme contexts (Kheang et al., 2014a), joint-sequence models (Bisani et al., 2008), and a WFST-based approach (Novak et al., 2012). Second, it generates a PTN using the obtained phoneme sequences and then selects the best phoneme from each block between two nodes in the PTN—a PTN bin—to represent the final output. For the best output phoneme selection, in this study, we also propose a model prioritization voting algorithm that is more accurate than the voting algorithm implemented in the NIST Recognizer Output Voting Error Reduction (ROVER) system (Ficus, 1997).

2. PTN-based G2P Conversion

2. 1. Six Data-Driven Models for G2P Conversion

Many data-driven approaches for G2P conversion have been proposed, but the joint-sequence models implemented in Sequitur-g2p (Web-1) and the WFST-based G2P conversion available in Phonetisaurus toolkit (Web-2) have proven to be the most powerful statistical approaches for dealing with OOV words. In addition, the use of the context information of each output phoneme in our two-stage ANN-based G2P conversion has also proven to be important for increasing the accuracy of OOV words (Kheang et al., 2014a). In order to build our new approach, we therefore used three kinds of existing approaches, the G2P conversions based on joint-sequence models, WFSTs, and ANNs, to implement six different models.

The first model is a statistical joint-sequence model-based G2P conversion built in the Sequitur-g2p toolkit (Bisani et al., 2008). The second model refers to the original WFST-based approach proposed by (Novak et al., 2012), which was implemented to develop a rapid and high quality joint sequences-based G2P conversion model. For the third model, we integrated a specific grapheme generation rule (GGR) listed in Table 1, into the previous WFST-based model to allow the addition of extra detail to the vowel graphemes appearing in a given word (Kheang et al., 2014b); the rule in Table 1 can distinguish the separated vowel V in the CVC pattern and the last vowel V_n in the $V_1V_2...V_n$ pattern from the connecting vowels V_1, V_2, ..., V_{n-1} in the $V_1V_2...V_n$ pattern. According to the first-stage of our previous two-stage ANN-based G2P conversion (Kheang et al., 2014a), three other remaining models were implemented based on the ANNs using a context window of plus/minus x graphemes (i.e., a window of $2x+1$ graphemes) as input and a window of plus/minus y phonemes as output of the network; in this study, we used 17 graphemes (i.e., $x = 8$) and three different values of y (i.e., $y = \{0, 1, 2\}$) for three different models (i.e., ANN1, ANN3 and ANN5 depicted in Fig.1). By displaying all the output windows one after another, Fig.1 demonstrates that there are 2y+1 columns of phonemes, and hence 2y+1 different phoneme sequences can be extracted vertically by using the information of the surrounding columns if necessary.

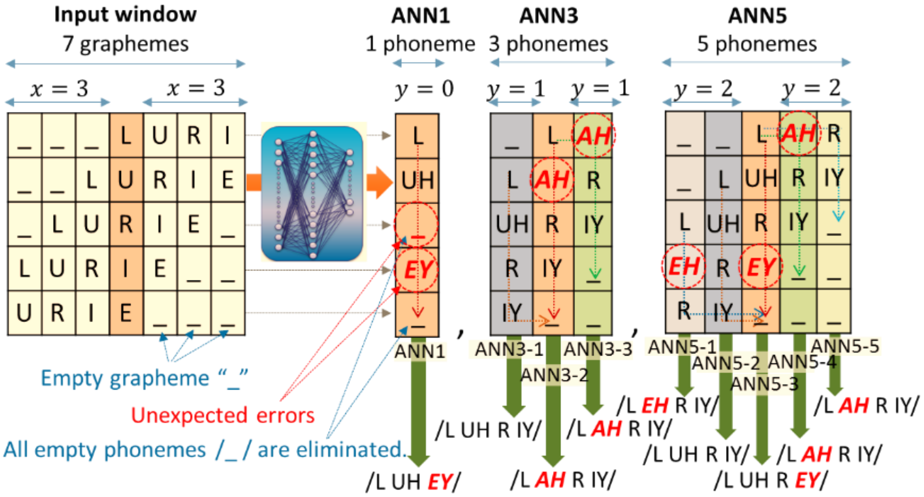

Fig. 1. Schema of the three proposed ANN-based G2P conversion models. This figure also demonstrates the method for generating multiple phoneme sequences from the output of each model.

Table 1. The selected grapheme generation rule (GGR)

Rule (Word → Grapheme Sequence)		Example
If (n >1):	$v_1 \dots v_n c_{n+1} \rightarrow v_1 v_2 \;\; v_2 v_3 \; \dots \; v_{n-1} v_n \;\; v_n c_{n+1} \;\; c_{n+1}$	“OKEECHOBEE” ↓ “O K **EE** **EC** C H O B **EE** **E**”
	$v_1 \dots v_n$ ↵ $\rightarrow v_1 v_2 \;\; v_2 v_3 \; \dots \; v_{n-1} v_n \;\; v_n$	
If (n = 1):	$g_i \rightarrow g_i$	
Where $g_i = \{c_i\, , v_i\}$; $g_i, c_i\, , v_i$ = *grapheme, consonant and vowel at index i; n= number of connecting vowels in a given word;* ‘↵’ = *End of word*		

2. 2. PTN Generation Using Multiple Phoneme Sequences

As shown in Fig.2, our proposed approach for the automatic conversion of an input word into various phoneme sequences uses six G2P conversion models described in Section 2.1. Second, the use of the ROVER system (Ficus et al., 1997) allows us to align those obtained phoneme sequences using the dynamic time warping (DTW) algorithm, and then merge all of them to a single confusion network (CN) or PTN, as depicted in Fig.2. In this context, when there is any insertion or deletion problem during the alignment process, a NULL phoneme /@/ is used in the PTN to represent a NULL transition.

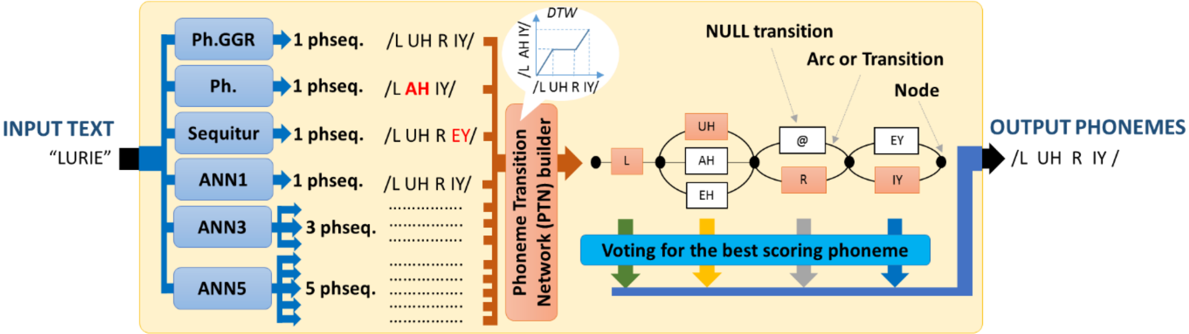

Fig. 2. Fundamental architecture of a PTN-based G2P conversion

By default, the ROVER system sets the costs of insertions (*Ins*), deletions (*Del*), and substitutions (*Sub*) for the alignment process to 3, 3, and 4, respectively. Hence, every two unmatched phonemes are treated equally, which means that the cost of phoneme substitution is equal to 0 if the comparing phonemes are the same and 4 otherwise. As a consequence, the method sometimes provokes incorrect alignments between vowel and consonant phonemes (e.g., /EH/ and /HH/), or between phonemes with close features (e.g., /AA/ and /AH/). In order to create a PTN with better alignment in this study, instead of a static value, we use the Hamming distance of articulatory features—an AF sequence (Yurie et al., 2010) represents a phoneme using 28 dimensions (place of articulation and manner of articulation)—(AFdist) and type similarity coefficient (*Tcoef*) to calculate the cost of substitution used in the DTW-based alignment process, as shown in the following equations:

$$D(i,j) = \min \begin{cases} D(i,j-1) & + Del, \quad where\ Del = 6 \\ D(i-1,j) & + Ins, \quad where\ Ins = 6 \\ D(i-1,j-1) & + AFdist(i,j) + Tcoef(i,j) \end{cases} \tag{1}$$

$$\mathrm{Tcoef(i,j)} = \begin{cases} 0, & If\ \big(Type(a_i) == Type(b_i)\big) \\ 10, & Otherwise \end{cases} \tag{2}$$

where a_i and b_j are the phonemes at index i and j of the two aligning phoneme sequences $phseq_1 = a_1a_2...a_n$ and $phseq_2 = b_1b_2...b_m$, respectively, $D(i,j)$ is the distance between $a_1a_2...a_i$ and $b_1b_2...b_j$, $AFdist(i,j)$ is the Hamming distance calculated from the AF of a_i and b_j, and $Tconf(i,j)$ is the coefficient indicating if a_i and b_j are in the same group of consonant or vowel phonemes. Both *Ins* and *Del* are set to the smallest value of *AFdist* between vowel and consonant phonemes. To avoid the mis-alignment between consonant and vowel phonemes, *Tcoef* must be bigger than the other parameters when a_i and b_j are in different groups.

2. 3. Best Phoneme Determination Using Model Prioritization Voting Schemes

When the PTN sequence has been established, we select the best scoring output phoneme from each PTN bin using our newly proposed voting schemes (known as the model prioritization voting schemes). As seen in Algorithm 1, these voting methods are the modified versions of three voting schemes (i.e., voting by frequency, average confidence score and maximum confidence score) in the ROVER system, which were proposed for maintaining the high accuracy of accurate source models when combined with other, poorer models. The scoring function is calculated based on the following formula:

$$score(ph) = \alpha\left(\frac{N(ph,i)}{n}\right) + (1-\alpha)C(ph,i) \qquad (3)$$

$$\mathrm{C(ph,i)} = \begin{cases} AVG\big(conf_1(ph,i), conf_2(ph,i), \dots, conf_n(ph,i)\big) & \blacktriangleright \text{ the voting by Avg. conf. score} \\ MAX\big(conf_1(ph,i), conf_2(ph,i), \dots, conf_n(ph,i)\big) & \blacktriangleright \text{ the voting by Max. conf. score} \end{cases} \qquad (4)$$

Where *N(ph,i)* is the number of occurrences of phoneme *ph* in the i^{th} PTN bin, while *n* here indicates the number of phoneme sequences to be combined. *C(ph,i)* represents the calculated confidence score for phoneme *ph* in the i^{th} PTN bin, where $conf_1(ph,i), \dots, conf_n(ph,i)$ indicate the different confidence scores for phoneme *ph* in the i^{th} PTN bin given by different models. The real value of α = [0... 1] refers to the tradeoff between using phoneme frequency and confidence score. In contrast, the value of the NULL confidence score *ncfs* in this paper was not a static value as in the original ROVER system, but a value equal to the confidence score assigned to the model where it belongs (e.g., $conf_2(NULL,i)$).

Algorithm. 1. Best phoneme selection using model prioritization voting schemes.

```
PROCEDURE Model_prioritization_voting($PTNbin_i$, $\alpha$, $Conf_1(ph,i)$, ..., $Conf_n(ph,i)$)
    Assign the N-best models                          ▶ e.g., the models with high accuracy
    if (N best models produce the same phoneme ph) and (N>1) then
        $bestPh \leftarrow ph$                                       ▶ Rapid selection
    else
        $bestPh \leftarrow argmax_{ph} score(ph)$                      ▶ using Eq. (3) and (4)
    end if
    return bestPh                          ▶ e.g., the best phoneme of the $i^{th}$ PTN bin
END PROCEDURE
```

3. Evaluation

3. 1. Datasets

In this study, we conducted experiments using the American English word-based pronunciation dictionary (CMUDict corpus available in Web-3) used in our previous studies (Kheang et al., 2014a), except that the newly prepared training and testing datasets selected only the words after the alignment process using the m2m-aligner software (available in Web-4), the aligned grapheme-phoneme pairs of which appeared at least four times in both datasets. Therefore, the training and testing datasets contained a total of 100,564 and 11,125 words, respectively.

3. 2. Performance Metrics

We evaluated the model performance in terms of phoneme accuracy (PAcc) and word accuracy (WAcc) using the NIST Sclite scoring toolkit (ref. Web-5). However, in this paper, we mostly report the results of the accuracy evaluated on OOV words. We also conducted statistical significance testing (measuring p-values) using McNemar's test.

3. 3. Experimental Results and Discussion

In our experiments, all six separate models, the Phonetisaurus using GGR (Ph.GGR), Phonetisaurus (Ph.), Sequitur-g2p (Sequitur), ANN1, ANN3, and ANN5, presented in Section 2.1 were treated as the baselines. The accuracy for ANN1, ANN3, and ANN5 was evaluated at their best epochs, 25, 31, and 47, respectively, while the accuracy of Sequitur was evaluated after the seventh training process (i.e., Model-7). As a result, in terms of the PAcc and WAcc of the OOV dataset, Fig.3 shows that the Ph.GGR, Ph., and Sequitur models outperform the ANN1, ANN3, and ANN5 models. Moreover, Ph.GGR provides the highest accuracy (PAcc = 93.63% and WAcc = 73.89%).

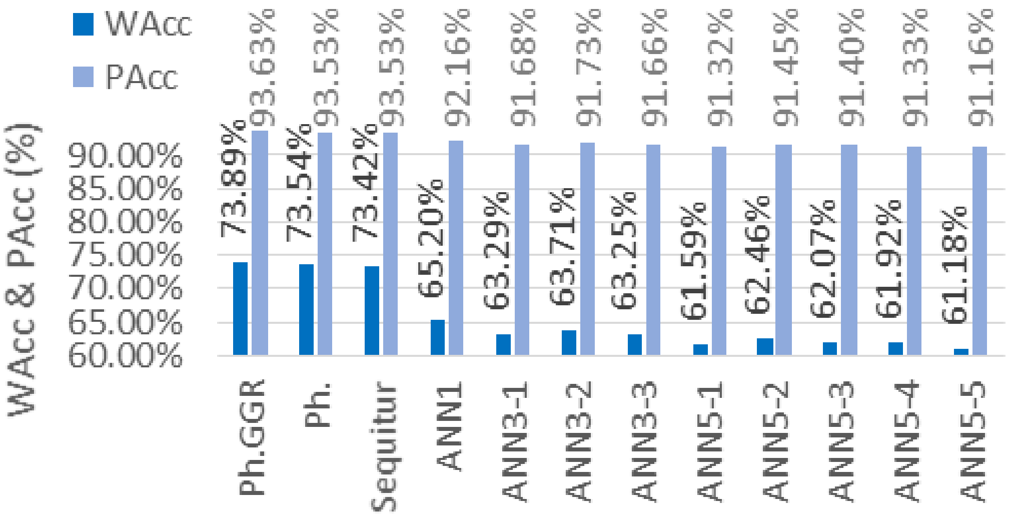

Fig. 3. WAcc and PAcc for the baseline approaches

As listed in Table 2, in order to compare our approach with the baselines as well as understand the impact of different model combinations, we proposed various PTN-based G2P conversion models (denoted as $\text{PTN}_{\text{n-m}}$) using different combinations of *n* phoneme sequences obtained from *m* models. For example, $\text{PTN}_{3\text{-}1}$ uses three (e.g., ANN3-1, ANN3-2, and ANN3-3) phoneme sequences and $\text{PTN}_{5\text{-}1}$ uses five (e.g., ANN5-1,…, ANN5-5) sequences, both obtained from the same ANN3 and ANN5, respectively. In the model prioritization voting schemes, we assign the models with highest accuracy to represent the N-best models, hence the symbol “P” in each row of Table 2 represents one of the N-best models involved in the PTN generation. Each symbol “x” in the table represents a model to be combined with the chosen N-best models. Moreover, in Eqs. (3) and (4), the confidence scores of the models involved in the PTN generation were manually assigned based on their performances; the model with the highest accuracy was assigned the highest score, while the one with the lowest accuracy was assigned the lowest score. Therefore, Table 2 reports the WAcc of all the proposed PTN-based models obtained when the confidence scores of Ph.GGR, Ph., Sequitur, ANN1, ANN3-x, and ANN5-x were assigned to 0.6, 0.5, 0.4, 0.3, 0.2, and 0.1, respectively.

Table 2. WAcc of the eleven proposed test sets using the model prioritization voting schemes.

	Ph.GGR	Ph.	Sequitur	ANN1	ANN3			ANN5					WAcc (%)		
					ANN3-1	ANN3-2	ANN3-3	ANN5-1	ANN5-2	ANN5-3	ANN5-4	ANN5-5	**M.P. Voting by frequency (α = 1)**	**M.P. Voting by Avg. conf. score (α = 0.5)**	**M.P. Voting by Max. conf. score (α = 0.5)**
PTN3-1					x	x	x						67.43%	67.43%	67.43%
PTN5-1								x	x	x	x	x	67.53%	67.53%	67.53%
PTN4-2	P				x	P	x						70.93%	73.35%	73.35%
PTN6-2	P							x	x	P	x	x	71.00%	72.80%	**74.30%**
PTN9-3				P	x	P	x	x	x	P	x	x	70.04%	70.13%	70.10%
PTN3-3.1	P	x	P										73.91%	73.90%	73.90%
PTN3-3.2	P		P	x									**74.65%**	**74.56%**	**74.56%**
PTN4-4	P	P	P	x									73.99%	**74.01%**	**74.27%**
PTN6-4	P	P	P		x	x	x						**74.77%**	**74.69%**	**74.60%**
PTN8-4	P	P	P					x	x	x	x	x	**74.86%**	**74.69%**	**74.79%**
PTN12-6	P	P	P	x	x	x	x	x	x	x	x	x	**74.92%**	**74.86%**	**74.97%**

The evaluation results show that the PTN-based models using multiple phoneme sequences extracted from a single model such as ANN3 or ANN5 (i.e., PTN$_{3-1}$ or PTN$_{5-1}$ can achieve a 4-5% higher WAcc than the original ANN-based approaches (i.e., ANN3-x or ANN5-x). Moreover, when we combined the results obtained from the three ANN-based models (i.e., ANN1, ANN3, and ANN5), the results of PTN$_{9-3}$ demonstrate that the WAcc is further increased.

In addition, the result of PTN$_{3-3.1}$ (where WAcc = ~73.91%) reveals that the combination of many accurate models with a similar design is not always helpful for improving the WAcc of the OOV words. In contrast, when the PTN model combines more accurate models with inaccurate models (e.g., in the case of PTN$_{3-3.2}$, PTN$_{4-4}$, PTN$_{6-4}$, PTN$_{8-4}$, and PTN$_{12-6}$), its performance level improves ($p<0.05$).

On the other hand, according to our experimental results using different values of α (not reported in this paper due to the space constraint), the three voting schemes in ROVER system are highly correlated with the threshold α and NULL confidence score compared to our proposed model prioritization voting schemes. In contrast to the models that use the original voting schemes, when α is increased, the model prioritization voting schemes that use the average and maximum confidence scores attempt to increase the performance of the PTN-based G2P conversion model by choosing the most accurate models for the N-best models and then maintain that performance by assigning the model confidence scores based on their individual performances. Furthermore, in this study, among the three model prioritization voting schemes, the evaluation results demonstrate that voting by frequency is the most stable and reliable voting scheme.

4. Conclusion

In this paper, we showed that the proposed PTN-based G2P conversion is a new effective method to improve the quality of phoneme prediction for OOV words because it allows different approaches for dealing with different problems to be combined. The evaluation results revealed that our model prioritization voting schemes could maintain and provide a reliably better model performance compared to the baseline approaches. To further improve our proposed approach, we plan to consider the use of the real phoneme confidence scores obtained from each combination approach into the model prioritization voting schemes and the use of other accurate models with different designs in place of ANN1 and ANN3.

Acknowledgements

This work is supported by a Grant-in-Aid for Young Sciences (B) 24700167 2012 from MEXT, Japan.

References

Bisani M., Ney H.. (2008). “Joint-Sequence Models for Grapheme-to-Phoneme Conversion,” *Speech Communication*, vol. 50, pp. 434-451.

Ficus G.J. (1997). “A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER),” *Proc. of ASRU*, Santa Barbara, CA, pp. 347-354.

Furuya, Y., Natori, S., Nishizaki, H. and Sekiguchi, Y. (2012). “Introduction of False Detection Control Parameters in Spoken Term Detection,” *Proc. of APSIPA ASC*, Hollywood, CA.

Kanda N., Itoyama K., Okuno G.H. (2013). “Multiple Index Combination for Japanese Spoken Term Detection with Optimum Index Selection based on OOV-Region Classifier,” *Proc. of ICASSP*, Canada, pp. 8540-8544.

Kheang S., Katsurada K., Iribe Y., Nitta T. (2014a). >Solving the phoneme conflict in Grapheme-To-Phoneme Conversion using a Two-Stage Neural Network-based approach “The Journal of the Institute of Electronics, Information and Communication Engineers,” E97-D(4), pp. 901-910.

Kheang S., Katsurada K., Iribe Y., Nitta T. (2014b). Novel Two-Stage Model for Grapheme-to-Phoneme Conversion using New Grapheme Generation Rules “Proc. of ICAICTA,” Indonesia.

Novak, J.R., Dixon, P.R. and Minematsu N. (2012). Improving WFST-based G2P Conversion with Alignment Constraints and RNNLM N-best Rescoring “Proc. of Interspeech,” Portland, Oregon.

Ogbureke, K.U., Peter. C., Julie. B.C. (2010). Hidden Markov Models with Context-Sensitive Observations for Grapheme-to-Phoneme Conversion “Proc. of Interspeech,” Japan.

Yurie, I., Mori. T., Katsurada. K., Nitta. T. (2010). Pronunciation Instruction using CG Animation based on Articulatory Feature “Proc. of ICCE2010,” Japan, pp. 501-508.

Web sites:

Web-1: http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html consulted Jun. 2014.

Web-2: http://code.google.com/p/phonetisaurus/, consulted Jun. 2014.

Web-3: http://pascallin.ecs.soton.ac.uk/Challenges/PRONALSYL/Datasets/, consulted May. 2014.

Web-4: https://code.google.com/p/m2m-aligner/, consulted Jul. 2014.

Web-5: http://www.itl.nist.gov/iad/mig/tools/, consulted Jul. 2014.