# NARROW ADAPTIVE REGULARIZATION OF WEIGHTS FOR GRAPHEME-TO-PHONEME CONVERSION

*Keigo Kubo* *Sakriani Sakti* *Graham Neubig* *Tomoki Toda* *Satoshi Nakamura*

Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan

## ABSTRACT

As the speech recognition field proceeds to open domain and multilingual tasks, the need for robust g2p conversion has been increasing. Towards this objective, we propose a new g2p conversion training method based on the Narrow Adaptive Regularization of Weights (NAROW) online learning algorithm. NAROW improves over its predecessor AROW by automatically adjusting hyperparameters to reduce mistake bounds, and ensuring that the learning rate is not updated when features for the input data have already been updated enough. The contribution of this paper is first to extend NAROW to structured learning, and show the inequality to bound the maximum number of errors in structured NAROW. In experiments, our proposed approach significantly improved over MIRA with consistent phoneme error rate reductions of 1.3-3.8% on a variety of dictionaries.

***Index Terms*—** g2p conversion, out-of-vocabulary word, online discriminative training, structured learning, NAROW

## 1. INTRODUCTION

Out-of-vocabulary (OOV) words are the bottleneck in large-vocabulary open-domain speech recognition systems [1] and text-to-speech systems [2]. In order to solve the problem of OOV words, Grapheme-to-phoneme (g2p) conversion, which is structured learning problems for which there are an extremely large number of candidate answers, has been used for a long time. As the speech recognition field proceeds to the open-domain and the multilingual [3] the need for robust g2p conversion has been increasing.

Rule-based approaches [4] and statistical approaches based on methods such as neural networks [5], decision trees [6], maximum entropy [7], joint sequence model [8, 9] have all been proposed for the g2p task. Most recent attempts have applied online discriminative training employing rich features [10, 11, 12]. One of the representative methods is the Margin Infused Relaxed Algorithm (MIRA) [13], an online structured learning method extended to g2p conversion by Jiampojamarn et al. [10, 11]. We have also recently proposed a method for g2p conversion based on structured Adaptive Regularization of Weight Vectors (AROW) [12], an online learning method designed to resolve MIRA's overfitting problems by estimating the confidence of each weight represented by a second order information matrix. However, structured AROW is still not a complete solution. The inverse of the second order information matrix representing the confidence of the weights serves as a learning rate for each weight, and these values are reduced with each update. This can cause cases where a particular weight converges to an inappropriate value, and cannot be updated further because its learning rate approaches zero. Also, it is difficult to choose AROW's hyperparameter to adjust the generalization of learning [14].

In order to solve the above problems, an expansion of AROW called Narrow Adaptive Regularization of Weights (NAROW) [15] has been proposed. Specifically, NAROW chooses a better value of the hyperparameter on each update to minimize the mistake bound. In addition, the second moment is not updated when features for input data have high confidence, preventing early convergence to bad weights. In order to incorporate these advantages into g2p conversion, we propose an online structured learning algorithm based on NAROW, which we call structured NAROW.

The first contribution in this paper is to extend NAROW to structured learning problems, and show the inequality to bound the maximum number of errors in this context. The inequality is derived based on the online convex optimization framework [15]. We also evaluate structured NAROW on a g2p task comparing with the joint sequence model, structured learning based on MIRA, and structured AROW.

## 2. G2P CONVERSION

### 2.1. Formalization

We define g2p conversion as a process of converting a grapheme sequence $\boldsymbol{x}$ into a phoneme sequence $\boldsymbol{y}$. Given a correct phoneme sequence $\boldsymbol{y}$ for a grapheme sequence $\boldsymbol{x}$, we formalize g2p conversion as

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \boldsymbol{w}^{\mathrm{T}} \Phi(\boldsymbol{x}, \boldsymbol{y}), \quad (1)$$

where $\boldsymbol{w}$ indicates the classifier's weight vector and $\Phi(\boldsymbol{x}, \boldsymbol{y})$ indicates a feature vector which consists of arbitrary values such as frequencies of joint n-gram features [11] on $\boldsymbol{x}$ and $\boldsymbol{y}$. In Eq.(1), $\hat{\boldsymbol{y}}$ can be efficiently obtained using dynamic programming. Structured learning can be employed to obtain a $\boldsymbol{w}$ that allows for accurate prediction of the correct phoneme sequence in this framework.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

### 2.2. Existing G2P online structured learnings

The most widely used method for online structured learning in g2p is based on MIRA [13]. Given the $i$-th example $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ and the $N$-best hypotheses $\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N$, MIRA updates $\boldsymbol{w}_t$ by solving

$$\arg\min_{\boldsymbol{w}_t} \frac{1}{2}\|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|^2; \text{ s.t. } \ell(\boldsymbol{x}_i, \boldsymbol{y}_i, \hat{\boldsymbol{y}}_n, \boldsymbol{w}_t) = 0; \forall n, \quad (2)$$

where $\ell$ is a loss function that outputs zero when the correct $\boldsymbol{y}_t$ scores higher than the hypothesis $\hat{\boldsymbol{y}}_t$ with sufficient margin, and a positive value otherwise. MIRA moves weights aggressively to correctly classify the $N$-best according to Eq.(2), and thus is prone to overfitting outliers or noisy data.

To resolve the overfitting, we have proposed structured AROW, the binary classifier AROW [14] to online structured learning. It updates $\boldsymbol{w}_t$ by minimizing the following function over $\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N$ sequentially.

$$\begin{aligned} L(\boldsymbol{w}_t, \Sigma_t) = \;& \mathbf{D_{KL}}(\mathcal{N}(\boldsymbol{w}_t, \Sigma_t)||\mathcal{N}(\boldsymbol{w}_{t-1}, \Sigma_{t-1})) \\ & + \tfrac{1}{2r}\ell(\boldsymbol{x}_i, \boldsymbol{y}_i, \hat{\boldsymbol{y}}_n, \boldsymbol{w}_t) + \tfrac{1}{2r}\boldsymbol{o}_t^{\mathrm{T}}\Sigma_t\boldsymbol{o}_t \end{aligned} \quad (3)$$

$\mathbf{D_{KL}}(\mathcal{N}(\boldsymbol{w}_t, \Sigma_t)||\mathcal{N}(\boldsymbol{w}_{t-1}, \Sigma_{t-1}))$ is the Kullback-Leibler divergence between the Gaussian distributions for $\boldsymbol{w}_t$ and $\boldsymbol{w}_{t-1}$, $r > 0$ is a hyperparameter to adjust the generalization of learning and $\boldsymbol{o}_t$ is $\Phi(\boldsymbol{x}_i, \boldsymbol{y}_i) - \Phi(\boldsymbol{x}_i, \hat{\boldsymbol{y}}_n)$. The covariance matrix $\Sigma$, roughly speaking, is a learning rate for $\boldsymbol{w}$. Its inverse $\Sigma^{-1}$ is a second order information matrix representing the confidence of each feature. $\Sigma$ is also updated so that learning rates for observed features decrease, namely, confidences for observed features increase. The minimization of Eq.(3) updates the weights so that the correct hypothesis scores higher than other hypotheses with minimal change to the distribution, considering the variance of each weight. By introducing the $\Sigma$, AROW avoids excessively moving the weights of the important features that have frequently been observed and updated. This property reduces the overfitting problem. We showed that structured AROW improves phoneme error rate and word error rate over MIRA in a g2p task employing a dataset including artificial noisy data [12].

However, there are two main issues in structured AROW. One is a problem where a some weights converge to inappropriate values for some orderings of the training data and cannot be updated further because their learning rates approach zero. Another is that it is difficult to choose a good $r$ on each round $t$.

## 3. STRUCTURED NAROW

In this section, we describe our proposed extension of NAROW from binary classification to structured learning. We first overview NAROW and note its differences from AROW. The difference between AROW and NAROW is the setting of the $r$. AROW sets a fixed value to $r$ in every round $t$, and the setting increases confidences linearly. On the other hand, the setting in NAROW increases confidences logarithmically when features for input data have low confidence, and does not increase confidences otherwise. This implies that NAROW, unlike AROW, does not reduce learning rates close to zero, preventing early convergence to bad weights. The setting of the $r$ in NAROW derives from minimizing the mistake bound for NAROW, namely, the setting chooses a good $r$ on each round $t$, which we will describe in the structured learning context in Section 4. The rest of this section describes a concrete algorithm for our structured NAROW.

**Algorithm 1** The proposed online structured learning algorithm based on Follow the Regularized Leader

```
Input:Training dataset $\boldsymbol{D} = \{(\bar{\boldsymbol{x}}_1, \bar{\boldsymbol{y}}_1), ..., (\bar{\boldsymbol{x}}_{|\boldsymbol{D}|}, \bar{\boldsymbol{y}}_{|\boldsymbol{D}|})\}$
and a series of regularizers $f_0, \ldots, f_{T-1}$
Output:weight vector $\boldsymbol{w}_T$
$t = 1, \boldsymbol{\theta}_0 = \mathbf{0}$
repeat
    for $i = 1$ to $|\boldsymbol{D}|$ do
        $\boldsymbol{w}_t = \nabla f^*_{t-1}(\boldsymbol{\theta}_{t-1}) = \Sigma_{t-1}\boldsymbol{\theta}_{t-1}$
        Predict $N$-best hypotheses $\tilde{\boldsymbol{y}}_1, ..., \tilde{\boldsymbol{y}}_N$ by $\boldsymbol{w}_t^{\mathrm{T}}\Phi(\bar{\boldsymbol{x}}_i, \tilde{\boldsymbol{y}})$
        for $n = 1$ to $N$ do
            Consider $\boldsymbol{x}_t := \bar{\boldsymbol{x}}_i$, $\boldsymbol{y}_t := \bar{\boldsymbol{y}}_i$, $\hat{\boldsymbol{y}}_t := \tilde{\boldsymbol{y}}_n$ and
            $\ell_t(\boldsymbol{w}_t) := \max(0, v_t d_t - \boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{o}_t)$
            if $\ell_t(\boldsymbol{w}_t) > 0$ then
                $\boldsymbol{z}_t = -\boldsymbol{o}_t \in \partial\ell_t(\boldsymbol{w}_t)$
                $\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \boldsymbol{z}_t$
                $t = t + 1$
                $\boldsymbol{w}_t = \nabla f^*_{t-1}(\boldsymbol{\theta}_{t-1}) = \Sigma_{t-1}\boldsymbol{\theta}_{t-1}$
            end if
        end for
    end for
until Stop condition is met
```

The learning algorithm for structured NAROW builds on the Follow the Regularized Leader (FTRL) defined in [15] for binary classification. Our proposed method expands this to structured prediction employing $N$-best candidates for learning, as shown in **Algorithm 1**. The FTRL updates the $\boldsymbol{w}_t$ by solving

$$\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}_t} \sum_{i=1}^{t-1} \eta_i \boldsymbol{z}_i^{\mathrm{T}}\boldsymbol{w}_t + f_{t-1}(\boldsymbol{w}_t), \quad (4)$$

where $\eta_i$ and $\boldsymbol{z}_i$ denote a learning rate and a subgradient of a loss function in $\partial\ell_i(\boldsymbol{w}_i)$ on round $i$ respectively. $f_{t-1}(\boldsymbol{w}_t)$ is a regularizer (also known as potential function) which controls the amount of generalization of learning. In the proposed structured NAROW, the $\eta_t$ is 1 on every round $t$ and the loss function $\ell$ is defined as

$$\ell_t(\boldsymbol{w}_t) = \max(0, v_t d_t - \boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{o}_t), \quad (5)$$

where $v_t = \boldsymbol{o}_t^{\mathrm{T}}\Sigma_{t-1}\boldsymbol{o}_t > 0$ adjusts differences in the number of features considering the variance $\Sigma_{t-1}$. Also $d_t = d(\boldsymbol{y}_t, \hat{\boldsymbol{y}}_t)$ indicates the loss incurred by incorrectly classifying $\boldsymbol{y}_t$ as $\hat{\boldsymbol{y}}_t$. We define $d_t$ as the number of phoneme errors for the g2p task. As a subgradient $\boldsymbol{z}_t$ of the above loss, we choose

$-\boldsymbol{o}_t$ when $\ell_t(\boldsymbol{w}_t) > 0$ and $\mathbf{0}$ otherwise. Next, we define a regularizer $f_t(\boldsymbol{o})$ as $\frac{1}{2}\boldsymbol{o}^{\mathrm{T}}\Sigma_t^{-1}\boldsymbol{o}$, where $\Sigma_t^{-1} = \Sigma_{t-1}^{-1} + \frac{\boldsymbol{o}_t\boldsymbol{o}_t^{\mathrm{T}}}{r_t}$, $r_t > 0$ and $\Sigma_0 = I$. The update form of $\Sigma_t^{-1}$ is the same as that of AROW when $r_t$ is a fixed value in every $t$. The setting of the $r_t$ in our structured NAROW is $r_t = \frac{v_t}{bv_t - 1}$ when $bv_t > 1$ and $r_t = +\infty$ otherwise, where $b > 0$ is a new hyperparameter (for its derivation, see Section 4). Note that in NAROW of binary classification, $f_t(\boldsymbol{o})$ is used instead of $f_{t-1}(\boldsymbol{o})$ as a regularizer at $\boldsymbol{w}_t$ like second order perceptron [16]. However, we use $f_{t-1}(\boldsymbol{o})$ for it because $\boldsymbol{o}_t$ required for $f_t(\boldsymbol{o})$ is unknown before the classifying in structured problem. Finally the updated weight for Eq.(4) is obtained by

$$\boldsymbol{w}_t = \nabla f_{t-1}^{-1}(\boldsymbol{\theta}_{t-1}) = \nabla f_{t-1}^{*}(\boldsymbol{\theta}_{t-1}) = \Sigma_{t-1}\boldsymbol{\theta}_{t-1}, \qquad (6)$$

where $\boldsymbol{\theta}_{t-1} = -\sum_{i=1}^{t-1}\boldsymbol{z}_i$, $\boldsymbol{\theta}_0 = \mathbf{0}$ and $\nabla f_{t-1}^{*}(\boldsymbol{\theta}_{t-1})$ is a gradient of the Fenchel conjugate $f_{t-1}^{*}$ for $f_{t-1}$. $f_{t-1}^{*}$ is defined as $f_{t-1}^{*}(\boldsymbol{\theta}_{t-1}) := \sup_{\boldsymbol{v}}\left\{\boldsymbol{\theta}_{t-1}^{\mathrm{T}}\boldsymbol{v} - f_{t-1}(\boldsymbol{v})\right\} = \frac{1}{2}\boldsymbol{\theta}_{t-1}^{\mathrm{T}}\Sigma_{t-1}\boldsymbol{\theta}_{t-1}$.

## 4. MISTAKE BOUND FOR STRUCTURED NAROW

In the previous section, The setting of the $r_t$ in structured NAROW was $r_t = \frac{v_t}{bv_t - 1}$ when $bv_t > 1$ and $r_t = +\infty$ otherwise, For deriving the setting, we show the mistake bound of structured NAROW based on online convex optimization.

### 4.1. Online convex optimization

Online convex optimization is a method for designing online learning algorithms and analyzing them through a potential function (regularizer) [17], and plays an integral role in deriving a mistake bound for NAROW. Online convex optimization based on primal-dual progress has been proposed in [18, 19, 20]. Orabona et al. [15] has generalized it to time-varying potential functions. The generalized online convex optimization framework based on FTRL is employed to derive the mistake bound of NAROW.

We describe some definitions from convex analysis. A $\beta$-strongly convex w.r.t a norm $\|\cdot\|$ is a function satisfying $f(\boldsymbol{v}) \geq f(\boldsymbol{u}) + \nabla f(\boldsymbol{u})^{\mathrm{T}}(\boldsymbol{v} - \boldsymbol{u}) + \frac{1}{2}\beta\|\boldsymbol{v} - \boldsymbol{u}\|^2$, where $\boldsymbol{u}, \boldsymbol{v} \in ri(dom(f))$. The functions $f_t(\boldsymbol{o}) = \frac{1}{2}\boldsymbol{o}^{\mathrm{T}}\Sigma_t^{-1}\boldsymbol{o}$ defined in Section 3 are 1-strongly convex w.r.t. the norms $\|\boldsymbol{o}\|_{f_t}^2 = \boldsymbol{o}^{\mathrm{T}}\Sigma_t^{-1}\boldsymbol{o}$. The dual norm $\|\cdot\|_*$ for a norm $\|\cdot\|$ is the norm defined as $\|\boldsymbol{u}\|_* := \sup\left\{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v} : \|\boldsymbol{v}\| \leq 1\right\}$. A $\beta$-strongly smooth w.r.t $\|\cdot\|_*$ is a function satisfying $f_*(\boldsymbol{u}+\boldsymbol{v}) \leq f_*(\boldsymbol{u}) + \nabla f_*(\boldsymbol{u})^{\mathrm{T}}\boldsymbol{v} + \frac{1}{2}\beta\|\boldsymbol{v}\|_*^2$. The $f_t^*(\boldsymbol{o}) = \frac{1}{2}\boldsymbol{o}^{\mathrm{T}}\Sigma_t\boldsymbol{o}$ defined in Section 3 are 1-strongly smooth w.r.t $\|\boldsymbol{o}\|_{f_t^*}^2 = \boldsymbol{o}^{\mathrm{T}}\Sigma_t\boldsymbol{o}$. The $\beta$-strongly convex/smooth are important properties in order to derive a mistake bound.

### 4.2. Derivation of mistake bound

This section gives a brief sketch of the mistake bound derivation, and we refer readers to [15] for more detail although there is a difference between binary classification and structured learning. We introduce the following condition

$$dt - \ell_t(\boldsymbol{u}) \leq -\boldsymbol{u}^{\mathrm{T}}\boldsymbol{z}_t; \forall \boldsymbol{u} \in S, v_t \geq 1, \qquad (7)$$

where $\boldsymbol{z}_t$ denotes any subgradient satisfying $\ell_t(\boldsymbol{w}_t) > 0$. For now, we assume $v_t \geq 1$ in Eq.(7) is satisfied, and discuss exceptions in Section 4.3. From Lemma 1 in [15], the setting in section 3, Eq.(7), and $f_t(\lambda\boldsymbol{u}) \leq \lambda^2 f_t(\boldsymbol{u})$, we have

$$\begin{aligned}\sum_{t\in M\cup U}(d_t - \ell_t(\boldsymbol{u})) &= D + \sum_{t\in U} d_t - \sum_{t\in M\cup U}\ell_t(\boldsymbol{u}) \\ &\leq \frac{\lambda\|\boldsymbol{u}\|^2}{2} + \sum_{t\in M\cup U}(\frac{\lambda(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}_t)^2}{2r_t} \\ &\quad + \frac{v_t r_t}{2\lambda(r_t + v_t)} - \frac{m_t^2}{2\lambda(r_t + v_t)} + \frac{m_t}{\lambda}), \qquad (8)\end{aligned}$$

where $m_t = \boldsymbol{o}_t^{\mathrm{T}}\Sigma_{t-1}\boldsymbol{\theta}_{t-1}$, $\boldsymbol{u}$ is any weight vector, $\lambda \geq 0$ is any scale factor, D is the number of errors, M is the setting of round $t$ for prediction mistakes and U is the setting of round $t$ for correct prediction, with $\ell_t(\boldsymbol{w}_t) > 0$.

### 4.3. Selection of hyperparameters $r_t$

We would like to choose better hyperparameters $r_t$ minimizing the right side of the above mistake bound. Orabona et al. [15] focus on minimizing the terms $\frac{\lambda(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}_t)^2}{2r_t} + \frac{v_t r_t}{2\lambda(r_t+v_t)}$ in Eq.(8). The setting proposed by Orabona et al. was $r_t = \frac{v_t}{bv_t - 1}$ when $bv_t > 1$ and $r_t = +\infty$ otherwise. The setting implies we should increase confidences logarithmically when features for input data have low confidence, and not increase confidences otherwise. Then, we have

$$\begin{aligned}D \leq &\sum_{t\in M\cup U}\ell_t(\boldsymbol{u}) + \frac{\lambda\|\boldsymbol{u}\|^2}{2} + \sum_{t:bv_t>1}\frac{\lambda bv_t(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}_t)^2}{2(r_t + v_t)} \\ &+ \frac{1}{2\lambda}\sum_{t\in M\cup U}(\min(\frac{1}{b}, v_t) - \frac{m_t^2}{(r_t + v_t)} + 2m_t) - \sum_{t\in U} d_t. \qquad (9)\end{aligned}$$

By tuning $\lambda$, the mistake bound for structured NAROW is

$$\begin{aligned}D \leq &\sum_{t\in M\cup U}\ell_t(\boldsymbol{u}) + \sqrt{\|\boldsymbol{u}\|^2 + \sum_{t:bv_t>1}\frac{bv_t(\boldsymbol{u}^{\mathrm{T}}\boldsymbol{x}_t)^2}{(r_t + v_t)}} \\ &\times \sqrt{\sum_{t\in M\cup U}(\min(\frac{1}{b}, v_t) - \frac{m_t^2}{(r_t + v_t)} + 2m_t) - \sum_{t\in U} d_t}. \qquad (10)\end{aligned}$$

Note that because the number of features in g2p conversion is large, we employ a diagonal matrix as the second order information matrix using $\mathrm{diag}\left\{\Sigma_t^{-1}\right\}$ instead of $\Sigma_t^{-1}$.

For the inequality $v_t \geq 1$ in Eq.(7), the $v_t$ is not too small when $b$ is a small value. So we attempt to satisfy the inequality by setting $b$ to a small value. [1]

[1] Note that g2p conversion employs rich feature set and therefore the inequality is almost always satisfied.

**Table 1**. *Dataset used in the experiment on the g2p task; dataset name (Dataset), the number of grapheme and phoneme symbols (g/p), vocabulary sizes of training data (Train), development data (Dev), and test data (Test) and the number of trials of cross-validation (K-fold)*

| Dataset | g/p | Vocabulary size | | | |
|---|---|---|---|---|---|
| | | Train | Dev | Test | K-fold |
| NETtalk | 26/50 | 17595 | 1000 | 1000 | 10 |
| Brulex | 40/39 | 23353 | 1373 | 2747 | 5 |
| CELEX English | 26/53 | 39995 | 15000 | 5000 | 1 |
| CMUdict | 27/39 | 100886 | 5941 | 12000 | 2 |

**Table 2**. *Parameter settings for the experiment.*

| | Sequitur | DirecTL+ | SAROW | SNAROW |
|---|---|---|---|---|
| joint n-gram | 7 | 5 | 5 | 5 |
| context window | - | 6 | 6 | 6 |
| $N$-best hypotheses | - | 5 | 5 | 5 |
| hyperparameter $r$ | - | - | 500, 1000,1500 | - |
| hyperparameter $b$ | - | - | - | 0.0075, 0.01,0.0125 |
| beam width | - | 50 | 50 | 50 |

**Table 3**. *Evaluation result for phoneme error rate (PER) and word error rate (WER) in the g2p task. Values on NETtalk, Brulex and CMUdict in this table are obtained by averaging results on each cross-validation. The best performance and performances that have no significant difference according to Paired Bootstrap Resampling [22] at a level of 0.05 over the best performance are written in bold.*

| Dataset | Measure | Sequitur | DirecTL+ | SAROW | NAROW |
|---|---|---|---|---|---|
| NETtalk | PER(%) | 7.71 | 6.70 | 6.75 | **6.53** |
| | WER(%) | 31.6 | **28.18** | 28.66 | **27.97** |
| Brulex | PER(%) | 1.26 | 1.03 | 1.09 | **0.99** |
| | WER(%) | 6.57 | **5.24** | 5.59 | **5.14** |
| CELEX English | PER(%) | 2.62 | 2.39 | 2.51 | **2.30** |
| | WER(%) | 12.15 | **11.07** | 11.81 | **11.17** |
| CMUdict | PER(%) | 6.77 | 6.19 | **6.15** | **6.11** |
| | WER(%) | 28.55 | **26.35** | **26.48** | **26.46** |

## 5. EXPERIMENT AND RESULT

We evaluated our structured NAROW on the g2p task. Table 1 shows datasets employed in the experiment. The development data is employed to determine the optimal number of training iterations. For datasets in Table 1, NETtalk (English) and Brulex (French) were obtained from the Pascal Letter-to-Phoneme Conversion Challenge[2]. CMUdict (English) and CELEX (English) were also obtained from their corresponding Web pages[3] [4]. We attempted to faithfully follow the convention in [9] in terms of data exclusion and data split, except extracting development data from training data.

We employed Sequitur[5], DirecTL+[6] and structured AROW (SAROW) as baseline g2p conversion tools in this experiment. Sequitur is based on the generative model employing joint n-grams for graphemes and phonemes. DirecTL+ uses structured learning based on MIRA. The implementation of SAROW and structured NAROW (SNAROW) used in this experiment is implemented in slearp[7]. DirecTL+, SAROW and SNAROW, which is our proposed approach, employed context features, chain features, and joint n-gram features. For alignment used in DirecTL+, SAROW and SNAROW, we used the unconstrained many-to-many alignment method of [21] as implemented in mpaligner[8]. The context window size, joint n-gram size, hyperparameter $r$ and $N$-best hypotheses for training were determined based on our previous work [12], except beam width for beam-search pruning and hyperparameter $b$. Table 2 shows their details. The training iterations, the hyperparameter $r$ and $b$ are determined by phoneme error rate on the development data.

Table 3 shows the evaluation result on the g2p task. From Table 3, for the PER, SNAROW improved over all other approaches with a significant difference on all datasets except CMUdict according to Paired Bootstrap Resampling [22] at a level of 0.05. The error rate reduction (ERR) over MIRA was 2.5% on NETtalk, 3.9% on Brulex, 3.8% on CELEX and 1.3% on CMUdict. Thus, we can see that replacing structured AROW with structured NAROW is effective on real datasets, even the relatively clean ones used in our experiments.

## 6. CONCLUSION

We proposed structured NAROW, extending NAROW to structured learning, and evaluated it on the g2p task. Structured NAROW solves structured AROW's problems by choosing a better value of the hyperparameter $r$ on each update so that the mistake bound is low and avoiding updating the second-moment when features for input data have high confidence. In the experiment, our proposed approach significantly improved over MIRA, with consistent ERRs of 1.3-3.8% in PER on a variety of dictionaries.

## 7. ACKNOWLEDGMENTS

Part of this work was supported by JSPS KAKENHI Grant Number 24240032 and by the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

[2] http://pascallin.ecs.soton.ac.uk/Challenges/PRONALSYL/Datasets

[3] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[4] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC96L14

[5] http://sequitur.info/

[6] http://code.google.com/p/directl-p/

[7] http://sourceforge.jp/projects/slearp/

[8] http://sourceforge.jp/projects/mpaligner/

## 8. REFERENCES

[1] Bahl et al., "Automatic phonetic baseform determination," in *Proc. ICASSP*. 1991, pp. 173–176, IEEE.

[2] Juergen Schroeter, Alistair Conkie, Ann Syrdal, Mark Beutnagel, Matthias Jilka, Volker Strom, Yeon-Jun Kim, Hong-Goo Kang, and David Kapilow, "A perspective on the next challenges for tts research," in *Speech Synthesis, 2002. Proceedings of 2002 IEEE Workshop on*, 2002, pp. 211–214.

[3] Tim Schlippe, Sebastian Ochs, and Tanja Schultz, "Grapheme-to-phoneme model generation for Indo-European languages," in *Proc. ICASSP*, 2012, pp. 4801–4804.

[4] Ronald M Kaplan and Martin Kay, "Regular models of phonological rule systems," *Computational linguistics*, vol. 20, pp. 331–378, 1994.

[5] Terrence J Sejnowski and Charles R Rosenberg, "Parallel networks that learn to pronounce English text," *Complex Syst.*, vol. 1, pp. 145–168, 1987.

[6] Walter Daelemans and Antal Van Den Bosch, "Language-independent data-oriented grapheme-to-phoneme conversion," in *Progress in Speech Processing*. 1997, pp. 77–89, Springer-Verlag.

[7] Stanley F Chen et al., "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc. EUROSPEECH*, 2003, pp. 2033–2036.

[8] Sabine Deligne and Frédéric Bimbot, "Inference of variable-length linguistic and acoustic units by multigrams," *Speech Communication*, vol. 23, no. 3, pp. 223–241, 1997.

[9] Maximilian Bisani and Hermann Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[10] Sittichai Jiampojamarn and Grzegorz Kondrak, "Online discriminative training for grapheme-to-phoneme conversion," in *Proc. INTERSPEECH*, 2009, pp. 1303–1306.

[11] Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak, "Integrating joint n-gram features into a discriminative training framework," in *Proc. NAACL-HLT*, 2010, pp. 697–700.

[12] Keigo Kubo, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Grapheme-to-phoneme conversion based on adaptive regularization of weight vectors," in *Proc. INTERSPEECH*, 2013, pp. 1946–1950.

[13] Koby Crammer and Yoram Singer, "Ultraconservative online algorithms for multiclass problems," *Journal of Machine Learning Research*, vol. 3, pp. 951–991, 2003.

[14] Koby Crammer, Alex Kulesza, and Mark Dredze, "Adaptive regularization of weight vectors," in *Advances In Neural Information Processing Systems*, 2009, vol. 23, pp. 414–422.

[15] Francesco Orabona and Koby Crammer, "New adaptive algorithms for online classification," in *Proc. NIPS*, 2010, pp. 1840–1848.

[16] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile, "A second-order perceptron algorithm," *SIAM Journal on Computing*, vol. 34, no. 3, pp. 640–668, 2005.

[17] Nicolo Cesa-Bianchi and Gábor Lugosi, "Potential-based algorithms in on-line prediction and game theory," *Machine Learning*, vol. 51, no. 3, pp. 239–261, 2003.

[18] Shai Shalev-shwartz and Yoram Singer, "Convex repeated games and fenchel duality," in *Advances in Neural Information Processing Systems 19*, pp. 1265–1272. MIT Press, Cambridge, MA, 2006.

[19] Shai Shalev-Shwartz, *Online learning: Theory, algorithms, and applications*, Ph.D. thesis, The Hebrew University, 2007.

[20] S Kakade, Shai Shalev-Shwartz, and Ambuj Tewari, "On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization," *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, 2009.

[21] Keigo Kubo, Hiromichi Kawanami, Hiroshi Saruwatari, and Kiyohiro Shikano, "Unconstrained many-to-many alignment for automatic pronunciation annotation," in *Proc. APSIPA*, 2011, pp. 1–4.

[22] Philipp Koehn, "Statistical significance tests for machine translation evaluation.," in *EMNLP*, 2004, pp. 388–395.