
Xiang-Dong Zhou¹, Feng Tian¹, Cheng-Lin Liu², Hong-An Wang¹

¹Beijing Key Lab of Human-Computer Interaction, Institute of Software, Chinese Academy of Sciences, Beijing 100190, P.R. China
²National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China
¹{xiaodong, tianfeng, hongan}@iscas.ac.cn, ²liucl@nlpr.ia.ac.cn

Abstract—Semi-Markov conditional random fields (semi-CRFs) are usually trained with maximum a posteriori (MAP) criterion which adopts the 0/1 cost for measuring the loss of misclassification. In this paper, based on our previous work on handwritten Chinese/Japanese text recognition (HCTR) using semi-CRFs, we propose an alternative parameter learning method by minimizing the risk, in which the misclassification costs are not equal, but different depending on the hypothesis and the ground-truth. The proposed method is lattice-based, i.e., the hypothesis space is the entire lattice on which the semi-CRF is defined. Experimental results on two online handwriting databases: CASIA-OLHWDB and TUAT Kondate demonstrate that minimum-risk training can yield superior string recognition rates compared to MAP training.

I. INTRODUCTION

Due to the large character set and the ambiguity of character segmentation, handwritten Chinese/Japanese text recognition (HCTR) is generally accomplished by an integrated segmentation and recognition approach based on character over-segmentation [1]. The input string (text line image for offline data or pen-tip trajectory for online data) is over-segmented into a sequence of components according to the overlapping between strokes (Fig. 1(a)), with each component being a character or part of a character. Subject to constraints of character width, consecutive components are combined to generate candidate characters, which constitute the segmentation candidate lattice (Fig. 1(b) and Fig. 1(c)). On assigning each candidate character a number of candidate classes using a character classifier, we construct the segmentation-recognition candidate lattice (referred to as lattice for brevity). Each path in the lattice corresponds to a segmentation-recognition hypothesis, which is evaluated by a parameterized function combining the character recognition score, geometric and linguistic contexts, and the string recognition result is obtained by searching for the optimal path with maximum score.

In our previous work [2], a semi-Markov conditional random field (semi-CRF) [3] based approach has been proposed for HCTR. The semi-CRF model is defined on the lattice to directly estimate the a posteriori probability of a segmentation-recognition hypothesis, in which the information of character recognition, geometric and linguistic contexts are defined as feature functions. This model provides principled tools for both parameter learning and decoding under the maximum a posteriori (MAP) criterion and enables the fusing of high-order features (long range context, such as the trigram language model).

![Fig. 1. Generation of segmentation-recognition candidate lattice. (a) Component sequence; (b) Candidate characters; (c) Segmentation candidate lattice, where each node denotes a candidate segmentation point, each edge corresponds to a candidate character, and the bold lines indicate the desired segmentation; (d) Candidate classes of the desired segmentation path.](image)

According to the Bayesian decision theory [4], the optimal decision should be made by minimizing the overall risk associated with a cost function measuring the loss of misclassification. When the 0/1 cost is adopted in the Bayes decision rule, we get the popular MAP criterion. The 0/1 cost simply assigns no loss to a correct prediction and a unit loss to an error. Consequently, for HCTR, it aims to minimize the string error rate rather than the character error rate. For example, a hypothesized path (cf. Fig. 1) containing one or more character-level errors, or a totally different path, as compared to the correct, will incur the same amount of loss. However, the performance of HCTR is usually measured in terms of character errors (insertions, deletions and substitutions) [2], [5], instead of the errors of whole sentence or text line. So, for HCTR, the use of 0/1 cost will lead to a mismatch between classifier training and performance evaluation.

In this paper, based on our previous work on HCTR [2], we apply the minimum-risk (MR) estimation framework to parameter learning of semi-CRFs. By incorporating the non-
uniform (non-0/1) misclassification cost, this criterion is more directly related to the character error rate in contrast to the MAP rule which aims at minimizing the string error rate.

II. RELATED WORK

In the community of speech recognition, risk minimization has been used for parameter learning of hidden Markov models (HMMs). The overall risk criterion [6], whose calculation is based on the Levenshtein distance between the correct transcription and the N-best recognized transcriptions, can consistently decrease the recognition errors when compared to the standard maximum likelihood training. Minimum phone error (MPE) training [7] can be interpreted as an instance of minimum-risk training where the set of all possible phone sequences forms the hypothesis space. Use of this criterion has been shown to outperform the maximum mutual information (MMI) criterion on several speech recognition tasks [7]. In [8], several objective functions based on the MPE criterion are compared on the task of broadcast news recognition, and the results show that the most promising technique is the minimum phone frame error rule, which is a frame-level version of MPE. Gibson et al. [9] evaluated different error approximation strategies based on the frame error metric and demonstrated that significant improvements can be observed on a large vocabulary speech recognition task when the symmetrically normalized frame error (SNFE) is adopted. Xiong et al. [10] propose a minimum tag error criterion for discriminative training of linear-chain CRFs, which is an average of the raw tag accuracy over all possible label sequences weighted by their likelihood. However, to the best of our knowledge, such risk minimization technique has not been applied to semi-CRFs. For HCTR, the only work on minimum-risk learning is the maximum character accuracy method [5], which takes the N-best list as the hypothesis space when calculating the risk. In contrast, use of a lattice to represent the hypothesis space is favored because it is a more compact representation of usually many hypotheses. Moreover, lattice-based training can directly take advantage of the inference algorithms for semi-CRFs.

III. SEMI-CRFs FOR STRING RECOGNITION

In our previous work [2], a semi-CRF based approach for HCTR is proposed, in which the semi-CRF model is defined on the lattice (cf. Fig. 1) to directly estimate the a posteriori probability \( P(S,Y|X;\Lambda) \) of a hypothesized segmentation-recognition path \((S,Y)\) given the string \(X\):

\[
P(S,Y|X;\Lambda) = \frac{1}{Z(X;\Lambda)} \prod_{c \in S} \psi_c(X,Y_c;\Lambda),
\]

where \(S\) denotes a segmentation of \(X\) (character sequence) and \(Y\) denotes a label sequence of \(S\). \(\psi_c(X,Y_c;\Lambda)\) is the potential function on maximal clique \(c\) (consecutive characters in the lattice):

\[
\psi_c(X,Y_c;\Lambda) = \exp \left\{ \sum_{k=1}^{K} \lambda_k f_k(X_c,Y_c) \right\}.
\]

\(f_k(X_c,Y_c)\) is the \(k\)-th feature function defined on clique \(c\), which models character recognition, geometric or linguistic context. We also refer to \(Y_c\) as a labeling of clique \(c\). \(\Lambda = \{\lambda_k | k=1,\ldots,K\}\) are the weighting parameters to be learned. \(Z(X;\Lambda)\) is the partition function defined as the summation over all the paths in the lattice:

\[
Z(X;\Lambda) = \sum_{(S',Y') \in S'} \prod_{c \in S'} \psi_c(X,Y'_c;\Lambda).
\]

Given \(N\) training samples: \(\{(X^i,S^i,Y^i) | i=1,\ldots,N\}\) (strings with segmentation points and character classes labeled), following the standard MAP estimation, the conventional training criterion is to minimize the conditional log-likelihood (CLL) loss with \(L_2\)-norm regularization:

\[
L_{CLL}(\Lambda) = - \sum_{i=1}^{N} \log P(S^i,Y^i|X^i;\Lambda) + \frac{C}{2} \|\Lambda\|^2
\]

where \(C\) is a positive constant balancing the loss term against the regularization term.

The Viterbi-like decoding tries to find the optimal path with maximum a posteriori probability:

\[
(S^*,Y^*) = \arg \max_{(S,Y)} P(S,Y|X;\Lambda).
\]

In practice, to accelerate the recognition process, approximate decoding, such as beam search is usually adopted [2].

IV. MINIMUM-RISK TRAINING

In contrast to the CLL loss (cf. Eq. (4)) which aims to maximize the posterior probability on the training set, the minimum-risk (MR) objective is to minimize the expected cost:

\[
L_{MR}(\Lambda) = \sum_{i=1}^{N} \sum_{(q,Y_q) \in H} P(q,Y_q|X^i;\Lambda) \tilde{\ell}((q,Y_q),(S^i,Y^i)),
\]

where the summation space \(H\) is the entire lattice. \(\ell((S,Y),(S',Y'))\) signifies the cost when recognizing string \(X^i\) as \((S,Y)\) instead of the ground-truth \((S',Y')\). In Eq. (6), to avoid enumerating numerous paths in the lattice, we consider the cost functions that can be decomposed onto each character along the hypothesized path:

\[
\tilde{\ell}((q,Y_q),(S^i,Y^i)) = \sum_{q \in S} \tilde{\ell}(q,Y_q), (S^i,Y^i))
\]

where \(\tilde{\ell}(q,Y_q),(S^i,Y^i))\) is the cost for an edge (character-label pair) \((q,Y_q)\) on path \((S,Y)\). Note here we do not force \(\ell((S^i,Y^i),(S',Y'))\) to be zero (for the MPE cost, cf. Section IV-B1).

With the above defined cost function, the minimum-risk loss can be rewritten as

\[
L_{MR}(\Lambda) = \sum_{i=1}^{N} \sum_{(q,Y_q) \in H} P(q,Y_q|X^i;\Lambda) \tilde{\ell}((q,Y_q),(S^i,Y^i)),
\]

where \(P(q,Y_q|X^i;\Lambda)\) denotes the marginal probability on \((q,Y_q)\):

\[
P(q,Y_q|X^i;\Lambda) = \sum_{(S,Y) \in H(q,Y_q)} P(S,Y|X^i;\Lambda).
\]

It can be seen from Eq. (8) that, to optimize the objective with gradient descent, we should first calculate the derivatives of the marginal probabilities. Note that Eq. (6) and Eq. (8) are actually regularized as in Eq. (4). We drop the regularization terms for succinct representation.
A. Derivatives of Marginal Probabilities

The partial derivatives of \( P(q, Y_q | X; \Lambda) \) with respect to the weighting parameters can be computed by

\[
\frac{\partial P(q, Y_q | X; \Lambda)}{\partial \lambda_k} = \sum_{(c, Y_c) \in \mathcal{C}} f_k(X_c, Y_c) \times (P(c, Y_c, q, Y_q | X; \Lambda) - P(c, Y_c | X; \Lambda) P(q, Y_q | X; \Lambda)),
\]

where \( c \) denotes the maximal clique (cf. Section III). Here \( (q, Y_q) \) is relaxed to be an arbitrary sub-segmentation path (character sequence) and its labeling. In Eq. (10), to avoid summing over all clique-labeling pairs in the lattice, we adopt the approximation that if \( c \) and \( q \) are disjoint (Fig. 2(a)), \( (c, Y_c) \) and \( (q, Y_q) \) are conditionally independent:

\[
P(c, Y_c, q, Y_q | X; \Lambda) \approx P(c, Y_c | X; \Lambda) P(q, Y_q | X; \Lambda).
\]

With the above approximation, the summation in Eq. (10) is reduced to those cliques that intersect with \( q \) (Fig. 2(b)(c)(d)). Note that \( P(c, Y_c, q, Y_q | X; \Lambda) \) is the marginal probability that both \( (c, Y_c) \) and \( (q, Y_q) \) are on a hypothesized path, so if \( c \) and \( q \) overlap but there is a common component whose label is different in \( Y_c \) and \( Y_q \), \( P(c, Y_c, q, Y_q | X; \Lambda) \) will be zero. Another case that makes \( P(c, Y_c, q, Y_q | X; \Lambda) \) zero is the overlapping part of \( c \) and \( q \) does not form common character(s) of them (Fig. 2(d)).

Fig. 2. Illustration of the relationships between a maximal clique (red) and a candidate character (black). The clique size is 2.

B. Cost Functions

To facilitate lattice-based training, we select the cost functions which can decompose the errors along the hypothesized path (cf. Eq. (7)). In our experiments, three cost functions initially proposed in speech recognition are investigated, including the MPE cost [7], the Hamming distance (HD) cost (also called frame error in [12]) and the SNFE cost [9]. Unlike the 0/1 cost which characterizes whether a transcription is identical to the ground-truth or not, the non-uniform cost functions measure the character-level errors according to how many characters or components are incorrect. In the following, we will detail the three cost functions and Fig. 3 illustrates the calculation process.

1) MPE Cost: The MPE cost is derived from the accuracy function proposed in [7] for measuring the gains of classifying the genuine path \((S^i, Y^i)\) as a hypothesized path \((S, Y)\):

\[
A((S, Y), (S^i, Y^i)) = \sum_{q \in S} \tilde{A}(q, Y_q), (S^i, Y^i)),
\]

where \((q, Y_q)\) is an edge (character-label pair) on path \((S, Y)\). In [7], to avoid the need for a dynamic programming alignment when calculating the Levenshtein distance, \(A((S, Y), (S^i, Y^i))\) is used to approximate the true accuracy (the number of characters in the correct transcript minus the Levenshtein distance to the reference). The edge accuracy \(\tilde{A}(q, Y_q), (S^i, Y^i))\) is defined as

\[
\tilde{A}(q, Y_q), (S^i, Y^i)) = \max_{q' \in S^i} \left\{ \begin{array}{ll} 1-2e(q, q') & \text{if } Y_q=Y_{q'}' \\ 1-e(q, q') & \text{otherwise} \end{array} \right.
\]

Here \((q', Y_{q'}')\) is an edge on genuine path \((S^i, Y^i)\), and \(e(q, q')\) is defined as the common component number between \(q\) and \(q'\), divided by the component number of \(q\). With the above accuracy function, the MPE cost is defined as

\[
\ell_{MPE}((S, Y), (S^i, Y^i)) = -A((S, Y), (S^i, Y^i)),
\]

and we can also derive the per-edge cost (cf. Eq. (7)):

\[
\ell_{MPE}(q, Y_q), (S^i, Y^i)) = -\tilde{A}((q, Y_q), (S^i, Y^i)).
\]

2) Hamming Distance Cost: The Hamming distance between two paths: \((S, Y)\) and \((S^i, Y^i)\) is defined as the number of components at which the labels differ:

\[
\ell_{HD}((S, Y), (S^i, Y^i)) = \sum_u (1 - \delta(Y_u, Y_u')), \]

where \(Y_u\) and \(Y_u'\) are the labels of component \(u\) on the hypothesized path \((S, Y)\) and on the genuine path \((S^i, Y^i)\), respectively. Inspired by the work on minimum time frame error rate decoding [13], by substituting this cost function into Eq. (6), the minimum-risk loss can be rewritten as

\[
L_{LR}(\Lambda) = \sum_{i=1}^{N} \sum_{u} (1 - P(Y_u'|X^i; \Lambda)),
\]

where \(P(Y_u'|X_i; \Lambda)\) is the marginal probability to observe \(Y_u'\) at component \(u\) and can be calculated by summing over the probabilities of all edges (character-label pairs) intersecting with component \(u\) and having identical label as \(Y_u'\):

\[
P(Y_u'|X_i; \Lambda) = \sum_{(S, Y) \in \mathcal{N}} \delta(Y_u, Y_u^{'})P(S, Y | X^i; \Lambda) = \sum_{(q, Y_q) \in \mathcal{N}} \sum_{(q', Y_{q'})} P(q, Y_q | X^i; \Lambda).
\]

Note that Eq. (16) can also be transformed to the form formulated in Eq. (7) by decomposing the errors on each edge of \((S, Y)\). However, instead of summing over all the lattice edges as in Eq. (8), Eq. (18) considers only those having identical labels as the overlapped reference characters.

3) SNFE Cost: In [9], some limitations of the MPE cost are discussed, including the overestimation and the asymmetry in error approximation, which causes an undesirable insertion to deletion bias. In contrast to the MPE cost, the SNFE cost [9] is symmetric as the Levenshtein distance and yields more accurate approximations for the deletion and insertion errors. The SNFE cost between a hypothesized path \((S, Y)\) and the reference path \((S^i, Y^i)\) is defined as

\[
\ell_{SNFE}((S, Y), (S^i, Y^i)) = \sum_{q \in S} \sum_{q' \in S^i} l((q, Y_q), (q', Y_{q'})),
\]

where \((q, Y_q)\) and \((q', Y_{q'})\) are edges (character-label pairs) on \((S, Y)\) and \((S^i, Y^i)\), respectively. \(l((q, Y_q), (q', Y_{q'})\) is defined as the number of overlapping components between \(q\) and \(q'\) at which \(Y_q\) and \(Y_{q'}\) differ, divided by the smaller component number of \(q\) and \(q'\). If no overlap exists between \(q\) and \(q'\), \(l((q, Y_q), (q', Y_{q'})\) is defined as zero. From Eq. (19),
we can derive the per-edge cost (cf. Eq. (7)):
\[ \ell_{SNFE}(q, Y_q, (S', Y')) = \sum_{q' \in S'} \ell((q, Y_q), (q', Y'_q)). \] (20)

<table>
<thead>
<tr>
<th>Symbol Digit Letter</th>
<th>Correct Rate (%)</th>
<th>String Error Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td>94.34</td>
<td>1.83</td>
</tr>
<tr>
<td>Symbol</td>
<td>92.04</td>
<td>95.06</td>
</tr>
<tr>
<td>Digit</td>
<td>93.58</td>
<td>94.62</td>
</tr>
<tr>
<td>Letter</td>
<td>94.54</td>
<td>94.53</td>
</tr>
</tbody>
</table>

Fig. 3. Illustration of the calculation of cost functions.

V. EXPERIMENTS

We evaluated the proposed method on unconstrained online handwritten text lines of a Chinese handwriting database CASIA-OLHWDB [14] and a Japanese handwriting database TUAT Kondate [15]. The test sets contain 10,510 text lines (269,674 characters of 2,631 classes) and 3,511 text lines (35,766 characters of 791 classes), respectively. The text lines of both databases have been annotated with segmentation points and character labels.

A. Experimental Setting

The features functions employed in our experiments include character recognition score, geometric and linguistic contexts [2]. The character recognition scores are given by the modified quadratic discriminant function (MQDF) [16]. The scores for class-dependent and class-independent geometries are given by the quadratic discriminant functions (QDFs) and the linear SVMs, respectively. All the classifier outputs are transformed to confidence [2].

4/5 of the merged training samples (isolated characters and those segmented from the texts) were used for training the character classifier (MQDF), and the remaining 1/5 were used for confidence parameter estimation. For both the two databases, the QDFs and the SVMs for geometric feature functions were trained on the features extracted from the respective training text lines. The Chinese language models were trained on a text corpus from the CLDC (Chinese Linguistic Data Consortium) [5]. The Japanese language models were estimated from the text corpus of the Japanese Mainichi Newspaper [2]. For each database, the semi-CRF parameters were learned on the training text lines. The details of experimental setting can be found in [2].

B. Performance Metrics

Following [5] and [2], the string recognition performance is evaluated by character-level Correct Rate (CR) and Accurate Rate (AR) derived by aligning the result string with the transcript using dynamic programming. We also use the term Character Error Rate (CER), which equals 1 − AR. And we denote the error rates of three error types (substitution, deletion and insertion) by SUB, DEL and INS, respectively. The string-level performance is measured by String Error Rate (SER), which is as the percentage of mis-recognized strings.

C. Experimental Results

We implemented the methods in MS Visual C++ 2008 and tested on a PC with Intel Quad Core 2.83GHz CPU and 4 GB-RAM. In training, the string samples were processed iteratively for five cycles in stochastic gradient descent. Trigram language models were used in our experiments. Following our previous work [2], to alleviate the computational burden, in training the lattice was first pruned with a pre-trained first-order semi-CRF with bigram language model, and in decoding we adopted the ratio threshold based beam search with threshold 10.

Table I compares the string recognition results on test string sets for minimum-risk (MR) training and MAP training (0/1 cost), as well as the training time for each cost function. In contrast to the CLL loss, the minimum-risk objective is non-convex, therefore the optimization procedure is prone to getting stuck in local optima. Inspired by the work of [11], in stochastic gradient descent, we took the learned parameters from CLL as initialization for risk minimization. This is because the convexity of the CLL loss can generally guarantee the parameters are in the right region.

<table>
<thead>
<tr>
<th>Costs</th>
<th>AR</th>
<th>CR</th>
<th>Chinese</th>
<th>Symbol</th>
<th>Digit</th>
<th>Letter</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/1</td>
<td>93.66</td>
<td>94.26</td>
<td>95.51</td>
<td>85.64</td>
<td>91.72</td>
<td>87.23</td>
<td>2.35</td>
</tr>
<tr>
<td>MPE</td>
<td>93.88</td>
<td>94.46</td>
<td>95.70</td>
<td>85.84</td>
<td>92.24</td>
<td>86.97</td>
<td>3.36</td>
</tr>
<tr>
<td>HD</td>
<td>93.95</td>
<td>94.53</td>
<td>95.76</td>
<td>85.98</td>
<td>92.47</td>
<td>86.97</td>
<td>3.15</td>
</tr>
<tr>
<td>SNFE</td>
<td>93.96</td>
<td>94.54</td>
<td>95.77</td>
<td>85.92</td>
<td>92.50</td>
<td>87.23</td>
<td>3.42</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Costs</th>
<th>AR</th>
<th>CR</th>
<th>Chinese</th>
<th>Symbol</th>
<th>Digit</th>
<th>Letter</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0/1</td>
<td>93.58</td>
<td>94.62</td>
<td>97.95</td>
<td>92.04</td>
<td>95.06</td>
<td>93.12</td>
<td>1.09</td>
</tr>
<tr>
<td>MPE</td>
<td>94.23</td>
<td>95.40</td>
<td>98.06</td>
<td>93.03</td>
<td>96.60</td>
<td>94.38</td>
<td>1.78</td>
</tr>
<tr>
<td>HD</td>
<td>94.34</td>
<td>95.43</td>
<td>98.15</td>
<td>93.10</td>
<td>96.42</td>
<td>94.20</td>
<td>1.35</td>
</tr>
<tr>
<td>SNFE</td>
<td>94.28</td>
<td>95.42</td>
<td>98.09</td>
<td>93.05</td>
<td>96.66</td>
<td>94.26</td>
<td>1.83</td>
</tr>
</tbody>
</table>

From Table I we can see that in contrast to the 0/1 cost (MAP training), the use of non-uniform costs can generally improve the string recognition rates (AR and CR) and the correct rates of almost all character types. The HD cost and the SNFE cost outperform the MPE cost on both the two test string sets. The performance of SNFE and HD are comparable, however, the training time with HD is much lower than that with SNFE due to less computational complexity (cf. Section IV-B2).

D. Error Analysis

Table II lists the error rates on the test string sets for the four cost functions (0/1, MPE, HD and SNFE), from which we can...
see that MR training outperforms MAP training on both character error rates and string error rates. Although the 0/1 cost aims at minimizing the string-level errors, the non-uniform costs still achieve lower SERs by reducing the character-level errors. Compared to MAP training, MR training can reduce all the three types of errors (substitutions, deletions and insertions) on CASIA-OLHWDB, while slightly increasing the insertion errors on TUAT Kondate. With comparable CERs, the SERs on CASIA-OLHWDB are much higher than those on TUAT Kondate. This is because the text lines in the former database are usually much longer than those in the latter (25.66 characters per string vs. 10.19 characters per string, on average). Taking the HD cost as an example, Fig. 4 illustrates some recognition results of the two types of training criteria. By MR training, the model can correctly recognize some characters which are mis-recognized by MAP training. In most of the cases, when MR training fails, MAP training also fails.

VI. CONCLUSION

This paper presents a minimum-risk training method for handwritten Chinese/Japanese text recognition using semi-CRFs, which aims at minimizing the character error rate rather than the string error rate by taking advantage of the non-uniform (non-0/1) cost functions. Three cost functions, including the MPE cost [7], the Hamming distance (HD) cost [12] and the symmetrically normalized frame error (SNFE) [9] are compared with the conventional 0/1 cost. Experimental results on CASIA-OLHWDB database and TUAT Kondate database show that minimum-risk training yields better string recognition rates than the conventional MAP training. HD and SNFE outperform MPE on test sets of both the two databases. The performance of SNFE and HD are comparable, while the training time with HD is much lower than that with SNFE due to less computational complexity.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no. 61232013, no. 61273269, no. 60933010, no. 61101015 and no. 61170182. The authors would like to thank TUAT Nakagawa laboratory for providing the Japanese databases.

REFERENCES