Lexicon Organization and String Edit Distance Learning for Lexical Post-Processing in Handwriting Recognition

Sabine Carbonnel and Éric Anquetil
IRISA, INSA
Campus de Beaulieu
F-35042 Rennes Cedex, France
sabine.carbonnel, eric.anquetil@irisa.fr

Abstract

This paper presents an optimized lexical post-processing designed for handwritten word recognition. The aim of this work is to correct recognition and segmentation errors using lexical information from a lexicon. The presented lexical post-processing is based on two phases: in the first phase a lexicon organization is made to reduce the search space into sub-lexicons during the recognition process. The second phase develops a specific edit distance to identify the handwritten word using a selection of the sub-lexicons. The paper exposes two original strategies of lexicon reduction and a new approach to automatically learn an edit distance specifically adapted to the properties of the on-line handwritten word recognition. Experimental results are reported to compare the two lexicon reduction strategies and first results emphasize the impact of the learning process of the new edit distance.

1 Introduction

The context of this study is handwriting recognition. Today many applications need an handwriting recognition module: PDA, tabletPC or smart-phones. Our isolated character recognition system RESIFCar [1] was integrated in a smart-phone device. We work now to optimize RESIFMot [2], an handwritten isolated words recognizer, which is based on an analytic approach that proceeds by segmenting the words according to different hypothesis of letter allographs. This analytical approach involves two known combined problems: segmentation ambiguity and letter confusion.

A lexical post-processing is necessary: it corresponds to a disambiguation step where contextual knowledge is introduced for the correction and the validation of the recognition results that we name "hypotheses". Two kinds of lexical knowledge are often used [5]: a statistical representation of letter n-grams [7] which are often used to order the word recognition hypotheses (but can not be used to validate or correct them); or a lexicon (this approach often implies the development of string comparisons based on edit distance algorithms). That is why the post-processing we propose focuses on an approach based on lexicon. However the approaches with lexicons are computationally inefficient in a large vocabulary context (above 10,000 words). The use of a lexicon reduction is a way to decrease the number of string comparisons [11, 10].

In this paper we focus on the lexical knowledge integration for the RESIFMot system. We study particularly two aspects: the lexicon organization and the edit distance. The aim is to deal with large lexicons according to material constraints (limited memory and resources) and to adapt an edit distance to the specific problem of handwriting recognition. In [4] we presented a first approach to organize and reduce the lexicon which is quite static and offer few improvement possibilities. That is why we explore a new and more flexible lexicon organization approach. A classic edit distance does not allow to solve specific problems of handwriting recognition. Some adaptations have been carried out to take this problem into account [4]. Even if good results are obtained, an empiric and manual estimation of the different edit operations and their respective costs is needed. We present here an automatic method to define edit operations and costs according to the recognition system properties: so it is possible to follow the evolution and improvement of the recognition module. Moreover, this new edit distance improves lightly the recognition rate. Few works relate string edit distance learning methods: in [12] we can find an other method based on a probabilistic distance learning and an optimization with an EM algorithm.

In section 2 we present the principle of the lexical post-processing. Section 3 contains a brief presentation of the first lexicon organization approach and a more detailed presentation of the second organization approach. In section

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

4 we expose the edit distance principle, the first realized adaptations and their limits. Section 5 is dedicated to the new edit distance based on our automatic learning method. Experiments and results are reported in section 6: we compare the two lexicon organization approaches and we evaluate the impact of the new edit distance on the recognition rate.

2 Basic principle of lexical post-processing

We focus our study on two problems: lexicon organization and adaptation of the edit distance to the handwriting recognition problem. The lexical post-processing approaches we present are made up of two phases: lexicon organization (and indexation of sub-lexicons from the complete lexicon) and lexical post-processing (exploitation of the knowledge, the organized lexicon, using a specific edit distance). The lexicon organization phase corresponds to an *a priori* clustering of the words having the same global characteristics into sub-lexicons. During the post-processing phase only some pertinent sub-lexicons are activated: it is the reduction step (step 1 in figure 1). This step permits to limit the search space when recognizing. The second step of the post-processing phase is based on a matching (edit distance) between the recognition hypotheses and the words inside the pertinent selected sub-lexicons (step 2 in figure 1). Choices concerning lexicon organization, indexation,

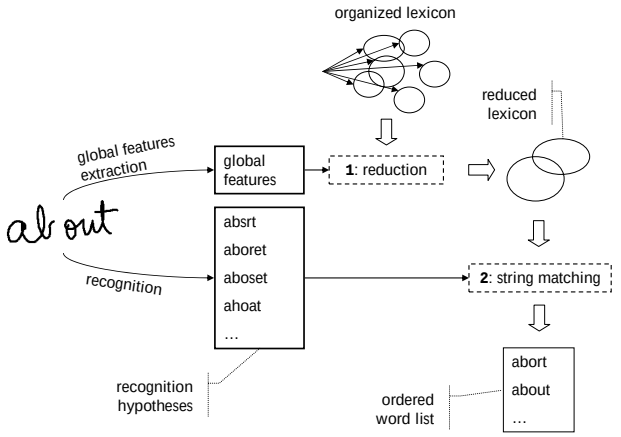

Figure 1. Lexical knowledge exploitation.

reduction and edit distance are important because they have a direct influence on the post-processing quality in terms of time, recognition rate and memory requirements.

3 Lexical knowledge modeling

The aim of the lexicon organization is to structure the lexicon for the reduction process. The reduction consists in the selection of one or several pertinent sub-lexicons with different criteria. The reduction can be either dynamic, sub-lexicons are formed during the post-processing phases, or static, sub-lexicon are *a priori* determined and indexed. We choose a static reduction method to lower the computation time. The aim is to produce the smallest sub-lexicons with a robust access criteria: the post-processing time decreases with the sub-lexicon size, but the risk to select a "bad" sub-lexicon increases.

Following this idea, we use global word characteristics to organize the lexicon into sub-lexicons according to their shapes. These characteristics can be considered as orthogonal information from those obtained by the analytic recognition process [9]. Word shapes are based on the most pertinent downstrokes (downstrokes shapes in figure 2), which are robust in handwriting [2] and composed of four sorts of pertinent downstrokes: ascender, descender, long and median.

We worked on two lexicon organization approaches. In the first approach [4] words are grouped according to their visual similarities and their size. Then the reduction step consists in selecting the sub-lexicons having exactly the same generic shape and similar size than the recognition hypotheses. In the second approach characteristic vectors are extracted from words and grouped in fuzzy clusters using an unsupervised clustering algorithm. A distance computation between vector hypotheses and sub-lexicon indexes is needed to select the nearest sub-lexicons.

First approach of organization: words classification from their generic shapes

The principle of this organization is to group the words with near shapes and sizes in the same sub-lexicon. The shapes are compressed. This is done by replacing a sequence of successive median downstrokes by only one. This coding increases the importance given to the prominent downstrokes because they are more robust than the median downstrokes in handwriting. We name such compressed shape: generic shape (see figure 2). To compensate the informa-

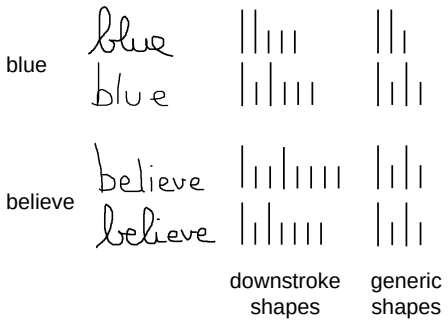

Figure 2. Generic shapes.

tion loss due to the compression step, a characteristic of the word length is added. It is based on an approximation of the downstroke number. These two global characteristics di-

rectly induce the static lexicon organization in sub-lexicons (they index the sub-lexicons).

Second approach of organization: unsupervised clustering from words characteristic vector

We present here a new approach based on an automatic clustering of visually similar words using a fuzzy unsupervised clustering algorithm. Each word is represented by a n-dimensional characteristic vector depending on robust information of handwriting. These characteristics are defined from word shape elements: median downstroke number, prominent downstroke number, and informations about the relative position of prominent downstrokes in the word. We use the unsupervised clustering algorithm FCM (fuzzy c-mean [6, 3]), to generate the sub-lexicons. These clusters are indexed by their centers or prototypes. The words are distributed in one or several clusters according to their membership degree. Redundancy is introduced specifically for words whose global form is near from several centers. Several parameters have an influence on this organization performances: first, the choice of the cluster numbers influences the sub-lexicons size and the post-processing time and quality and second the choice of the characteristics is an important factor for the organization quality. These two parameters bring flexibility and improvement possibilities to the organization. After this *a priori* organization step, we obtain a set of indexed sub-lexicons.

4 Lexical knowledge exploitation

The lexical post-processing, based on the hypothesis list, coming from the recognition process is constituted of two steps: first, the lexicon reduction to select one or several pertinent sub-lexicons (step 1 on figure 1); second, the string matching between recognition hypotheses and reduced lexicon words using a specific edit distance, in order to determine the nearest lexicon word from the handwritten word (step 2 on figure 1). We present in the following subsection strategies for the lexicon reduction depending of the lexicon organization. In subsection 4, we introduce the notion of edit distance and its adaptations to handwriting recognition.

Lexicon reduction strategies

During the reduction step, the strategy depends on the chosen lexicon organization. In the case of the organization with generic shapes, the reduction corresponds to the exact comparison between the pair (generic shape, size) and the sub-lexicon index. In the case of the organization with FCM unsupervised clustering, the distance between word vectors and each index is computed to estimate the nearest sub-lexicons. The number of selected sub-lexicons depends on a threshold ε defined as follow: $d_{min}(i)/d(i,j) \leq \varepsilon$ where $d_{min}(i)$ is the distance between the word vector i and its nearest center and $d(i,j)$ is the distance between the word vector i and the center j. The threshold ε permits to adapt the number of selected sub-lexicons: if a word vector is near only one center then only one sub-lexicon is selected. An evaluation of the impact of ε is presented in section 6.

Principle of an edit distance dedicated to handwriting recognition

Several distances can be used to correct and validate the recognition hypotheses. First, we present an extension of the Levenshtein distance [8] (substitution, insertion and deletion) to handwriting and then an optimized version more adapted to the recognition system RESIFMot.

Extended distance

Seni, Kripasundar and Srihari [13] extended the Levenshtein distance to compensate the specific errors induced by an handwriting recognition process. They added three edit operations with static costs: fusion, division and pair substitution. These costs are static and can belong to three categories for each operation: very likely, likely and unlikely. Edit operation tables and costs associated to each category for each operation are empirically determined (see table 1).

Table 1. Examples of edit operations for fusion (fus) and substitution (sub).

very likely fus cost: 0.3	likely fus cost: 0.35	very likely sub cost: 0.25
cl → d	ls → h	a → u
cr → a	oj → g	b → h
... → ...	... → ...	... → ...

Modified distance: $dist_{modif}$

We have optimized this extended distance [4] by introducing a penalization factor depending on structural differences between the compared characters (first modeling level of RESIFCar system [1]). An other penalization factor is based on the absence or on the presence of diacritic symbols (dots, t-bar, ...) which is a global information detected during the recognition phase.

This modified edit distance offers interesting results in comparison with the extended distance. However an empirical determination of the edit operations and their associated costs is needed (see table 1). Taking into account all the possible operations is very long and difficult. That is why we introduce here an automatic learning mechanism, presented in the following section, in order to define the set of possible edit operations and their associated costs.

5 Automatic learning of an edit distance for handwriting

Manually determining the tables set and the costs is difficult. Moreover these operations depend on the recognizer

properties, that is why improving the recognizer require to make a new empirical adaptation of the edit distance. In order to automatically build all the edit tables and their associated costs, we design an automatic learning process based on two steps (see figure 3): the first step uses the recognition system in a supervised mode to detect the set of all possible edit operations with a basic matching rule ($dist_{learn}$); the second step corresponds to the edit cost computation for all the possible edit operations in order to determine the edit distance parameters. The two steps are presented in

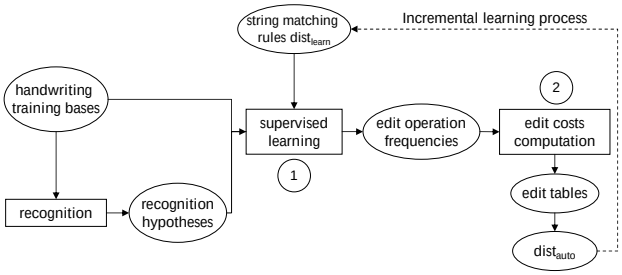

Figure 3. Principle of the edit distance learning.

the following and permit to learn the parameters of the new edit distance: $dist_{auto}$. Then we present an improvement method for $dist_{auto}$ inspired from boosting approaches.

Basic matching rule definitions: $dist_{learn}$

Our aim in this section is to initialize the learning by a first estimation using basic matching rules. The basic matching rules are defined according to the word shapes. We consider the word shape as a canonic downstroke sequence. We denote by $shape(m)$ the shape of the word m composed with the downstroke sequence. The $word_{ref}$ corresponds to a symbolic transcription of the handwritten word: this is the ground truth. The hyp is a character string produced by the recognition process.

The learning distance $dist_{learn}$ is composed of six character edit operations: substitution, insertion, deletion, fusion, division and pair substitution. An initial cost is allowed to each operation according to the character shapes. For op_1, regrouping the operations of substitution, fusion, division and pair substitution, we have:

$$cost_{op_1}(A,B) = \begin{cases} 0 & \text{if } A = B \\ cost_{fixed_{op1}} & \text{if } shape(A) = shape(B) \\ cost_{infinite} & \text{otherwise} \end{cases}$$

For op_2, the costs for the insertion and deletion operations are defined as follow:

$$cost_{op_2}(A) = \begin{cases} cost_{fixed_{op2}} & \text{if } A \text{ contains only median downstrokes} \\ cost_{infinite} & \text{otherwise} \end{cases}$$

The fixed costs associated to op_1 and op_2 operations are empirically defined according to the modified distance (see section 4). This basic distance $dist_{learn}$ is based on fundamental word structures and allows a matching between the learning pair: a word label $word_{ref}$ and an hypothesis hyp. The edit operations selected and counted in the learning tables correspond to the operations that obtained the minimal distance.

Step 1: computation of the edit operation frequencies

We use $dist_{learn}$ to match the learning pairs $word_{ref}$-hyp. Insertion and deletion are allowed only if the hypothesis downstroke number is not correctly detected *i.e.* if the hypothesis downstroke number is different from the expected $word_{ref}$ downstroke number. At the end of each matching, confused character sequences and their occurrences are counted in edit tables. For instance, if we consider the matching between "vmascimum" (hyp) and "maximum" ($word_{ref}$). The minimal distance is obtained with the v deletion and the fusion of s and c in x. Then v is added in the deletion table and (sc, x) in the fusion table. This learning is realized with the matching between labeled words $word_{ref}$ and the n first recognition hypotheses. It is possible to increase the importance of the edit operations learned from the nearest hypothesis by introducing a weight in the count of these operations in the edit tables.

Step 2: edit costs computation

This step consists in assigning a cost to each edit operation according to the frequencies which have been learned in the edit tables. The main idea is to assign a low cost to very frequent operations and a high cost to non-frequent operations. For each edit operation op the cost is proportional to $1/\#_{op}$ ($\#_{op}$ is the frequency computed in 5). After this step we obtain the new edit distance $dist_{auto}$.

Improvement of the learning process

We recently experimented an improvement process for the edit distance learning. The main idea is inspired by the boosting concepts [14]. We developed an incremental learning process based on three iterative steps: first, learning a new edit distance using the automatic learning distance presented before, second compute the errors for this distance on the learning base (pairs $word_{ref}$-hyp), third modify the weights of the learning base elements according to errors. At each iteration a new distance is computed according to the precedent distance. The incremental process permits to modify the impact of the counted edit operations according to the error rate: the importance of an edit operation is decreased if the current edit distance can not correct the hypothesis, but is increased otherwise. That means the learning process focuses on the errors that the current distance does not correct. After N iterations of this improvement algorithm we obtain N edit distances and their associated error rates. For the post-processing, a new edit distance ($dist_{autoInc}$) is computed by a combination of these N distances in order to give them an importance inverse to their error rate.

6 Experimentations and results

First, we report experimental results to compare the two different post-processing approaches according to the recognition rate, the post-processing time and the memory size requirement. Then we illustrate the impact of the edit distance learning method. The word recognizer RESIFMot we use is in an optimization phase, that means that we focus for the experiments on relative recognition rates.

Impact of the lexicon organization

The tests have been carried out on a base of 2000 handwritten words (from two writers). These words are the 1000 more likely words in English. We present the post-processing impact using a 25000 word lexicon including these 1000 words. The handwritten words are in lower case but this is the only constraint for the writers. The test set is independent of the learning set recognition system. The conditions of this experiments are the same than those exposed in [4]. We present results for the second organization approach, *i.e.* the word vectors clustering, before comparing them with the results of the first method, *i.e.* the generic shapes (section 3).

Figure 4-a concerns the influence of the sub-lexicon number on post-processing time and recognition rate. The post-processing time, which is directly dependent on the reduced lexicon size, decreases when the sub-lexicon number for the organization phase increases: the sub-lexicons are smaller. The recognition rate is around 78% (using a 25000 word lexicon) and is maximal for this experiment for a value of 250 clusters. Figure 4-b presents the impact of the threshold ε (section 3). This parameter has an influence on the number of selected sub-lexicons during the reduction. The number of selected sub-lexicons is inversely proportional to ε. According to this study, the optimal value for our lexicon is ε = 0.95. We use several criteria to compare the two lexicon organization approaches (see table 2): recognition rate, correct selection rate (*i.e.* how much times the correct word in is the reduced lexicon), lexicon words redundancy rate (*i.e.* how much time a word is present in the lexicon), and relative post-processing time. The "list" column corresponds to a lexicon without any specific organization. It is a reference to compare the two others. We observe that the first approach has a slightly better influence on the recognition rate and on the post-processing time than the second, but the redundancy rate is higher: in average a word is present 1.9 times in the lexicon. This introduces an important memory requirement. For the second approach, the redundancy rate is interesting, but the correct selection rate is lower (and the recognition rate decreases in the same way).

Table 2. Comparison of the organizations for a lexicon of 25000 words.

	1. generic shapes	2. word vectors clustering	list
redundancy rate	1.9	1.3	1
reduced lex. size	1000 w.	3000 w.	25000 w.
	20 x faster	7 x faster	
correct selection	98%	96%	100%
recognition rate	80%	78%	80 %

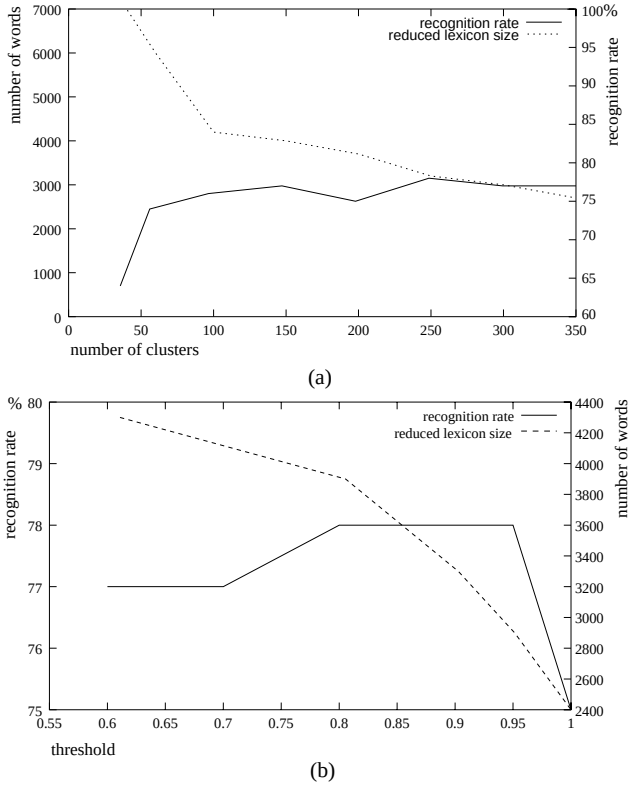

Figure 4. Influence of the sub-lexicon number (a) and of the threshold ε (b) on the recognition rate and on the reduced lexicon average size.

Impact of edit distance on recognition rate

Now we aim to compare the recognition rate obtained with the modified edit distance $dist_{modif}$ (with an empirical elaboration), the automatically learned edit distance $dist_{auto}$ and its improved version $dist_{autoInc}$. For these experiments we use a data base composed of 6000 words written by 7 writers. The learning set is a 3000 handwritten words base (4 different writers). The distance evaluation is carried out with an other handwritten base: 3000 words written by 3 other writers. This approach is writer-independent because learning set and test set are independent. The results reported in table 3 show that the automatically learned distance $dist_{auto}$ permits to obtain slightly better results (er-

Table 3. Recognition rate; omni-writer context, 25000 words lexicon.

rank	$dist_{modif}$	$dist_{auto}$	$dist_{autoInc}$
1	72.69 %	74.17 %	74.49 %
2	78.81 %	80.56 %	80.88 %
3	81.27 %	83.34 %	83.72 %
4	83.37 %	84.76 %	86.12 %
5	84.23 %	85.79 %	87.01 %

ror reduction rate: 5.7%) than with the modified distance $dist_{modif}$ empirically computed. These results are very interesting because they introduce the possibility to follow the recognition module evolutions by an automatic adaptation of the edit distance. The first impact of the edit distance learning improvement $dist_{autoInc}$ is low (error rate reduction: 6.9% after 15 iterations) but this method allows to reach the edit distance limits: the study of the remaining error cases show us that they can not be solved by an edit distance. Our perspectives are to introduce statistical informations (letters n-grams) directly in the segmentation graph of the recognition process.

7 Conclusion

In this article we presented two lexicon organization approaches and a method to automatically learn an edit distance dedicated to handwriting recognition. Experimental results highlight the main interests of the lexicon organizations: for the first approach using generic shapes, a time reduction and for the second approach using word vectors clustering, a reduced memory size. The second approach offers future optimization possibilities, that we are going to explore in our future works.

The automatically edit distance learning method avoids the empiric and hard elaboration of the precedent distances for the handwriting recognition. The results obtained show that it is possible to automatically adapt the distance to the recognition module and to its evolutions. This is a fundamental point to converge on an accurate handwriting recognition system. The first results for the incremental improvement of the edit distance learning permits to reach the limits of the edit distance correction possibilities.

8 acknowledgment

The authors would like to thank France Télécom R&D for collaborating on this work and Guy Lorette from the Université de Rennes 1 for providing advise.

References

[1] E. Anquetil and H. Bouchereau. Integration of an on-line handwriting recognition system in a smart phone device. In *International Conference on Pattern Recognition*, volume 3, pages 192–196, Quebec, Canada, August 2002.

[2] E. Anquetil and G. Lorette. Perceptual model of handwriting drawing application to the handwriting segmentation problem. In *International Conference on Document Analysis and Recognition*, volume 1, pages 112–117, Ulm, Germany, August 1997.

[3] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1981.

[4] S. Carbonnel and E. Anquetil. Lexical post-processing optimization for handwritten word recognition. In *International Conference on Document Analysis and Recognition*, pages 477–481, Edinburgh, August 2003.

[5] A. Dengel, R. Hoch, F. Hnes, T. Jger, M. Malburg, and A. Weigel. Techniques for improving OCR results. In *Handbook on Character Recognition and Document Image Analysis*, chapter 8. World Scientific Publishing Company, 1997.

[6] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.

[7] I. Guyon and F. Pereira. Design of a linguistic postprocessor using variable memory length markov models. In *International Conference on Document Analysis and Recognition*, pages 454–457, Montreal, Canada, August 1995.

[8] Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[9] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(2):149–164, February 2001.

[10] S. Madhvanath and V. Krpasundar. Pruning large lexicons using generalized word shape descriptors. In *International Conference on Document Analysis and Recognition*, pages 552–555, Ulm, Germany, 1997.

[11] S. Madhvanath and S.N. Srihari. Effective reduction of large lexicons for recognition offline cursive script. In *Fifth International Workshop on Frontiers in Handwriting Recognition*, pages 445–452, Essex, U.K., 1996.

[12] E. S. Ristad and P. N. Yianilos. Learning string edit distance. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 20(5):522–532, May 1998.

[13] G. Seni, V. Kripasundar, and R. K. Srihari. Generalizing edit distance for handwritten text recognition. In *Proceedings of SPIE/IS&T Conference on Document Recognition*, pages 54–65, San Jose, CA, February 1995.

[14] R. E. Shapire. A brief introduction to boosting. In *International Joint Conference on Artificial Intelligence*, pages 1401–1406, 1999.