

Word Extraction Associated with a Confidence Index for On-Line Handwritten Sentence Recognition

Solen QUINIOU, François BOUTERUCHE and Eric ANQUETIL

IRISA - INSA, Campus de Beaulieu, Rennes, FRANCE

Solen.Quiniou@irisa.fr, Francois.Bouteruche@irisa.fr, Eric.Anquetil@irisa.fr

Abstract. This paper presents an extension of our on-line sentence recognition system by integrating an automatic word extraction mechanism. Our word extraction task is based on the characterization of inter-stroke gaps, combined to a rejection strategy to evaluate the reliability of the gap classification results. A reconsideration mechanism then used this confidence index to create additional extracted word hypotheses by further controlling the complexity of the recognition task. Different metrics are used to evaluate the impact of this whole word extraction task on the recognition performance, on a set of 395 English sentences.

1. Introduction

With the rise of new devices like Tablet PC's, users are able to write larger pieces of text. Whereas the recognition of isolated characters and words already achieves high recognition rates, the handwritten text recognition task leads to new challenges. Word extraction is thus an important step since it allows the recognition of isolated words instead of whole text lines. As an on-line handwritten text is represented by a list of chronologically sorted strokes (a *stroke* is a list of chronologically sorted points, captured between a pen-down and a pen-up), the word extraction task consists in extracting each sublist of strokes corresponding to the words.

The main approach for the *a priori* word extraction is to identify the intra-word and inter-word gaps between consecutive strokes. The key of this problem is the computation of the distance between two consecutive strokes: it must be invariant to handwriting style variations such as slant, skew, or scale. In Oudot et al. (2004), the authors work on handwritten manuscript sentences and describe a *blank detection* task consisting in an inter-stroke gap classification task. They use the bounding box distance as inter-stroke horizontal distance which is heavily subject to handwriting style variations. In Liwicki et al. (2006), the authors use a bounding box distance too, but after preprocessing steps which normalize the signal with respect to slant and skew. Moreover, they determine the lower and upper baselines to perform height normalization. Then, they run an iterative segmentation algorithm adapted from off-line recognition (Varga & Bunke, 2005). The major drawback of this method is that it can not be used for an "on-the-fly" recognition since the whole text line is used to detect the inter-word gaps.

In this paper, we extend our on-line sentence recognition system (Quiniou & Anquetil, 2006) by adding an automatic word extraction step. Our word extraction task was conceived to be further extended to an "on-the-fly" recognition. To deal with the possible word extraction errors, we use a confidence index to evaluate the reliability of the extraction. Thanks to it, we control the complexity of the recognition task by limiting the number of extracted word hypotheses to reconsider. Different performance measures are used to evaluate the impact of this complete word extraction task with respect to a manual word extraction task (corresponding to the ground truth).

In Section 2, the recognition system is introduced. The word extraction task is then described in Section 3 while the sentence recognition is presented in Section 4. Finally, experimental results are discussed in Section 5.

2. Overview of the recognition system

The whole recognition system presented in Figure 1 extends our on-line sentence recognition system previously presented in Quiniou and Anquetil (2006). The word extraction module (see Section 3) aims at gathering the parts of the input handwritten sentence that correspond to the same handwritten word, to initialize the word graph. The nodes of this graph represent the segmentation frontiers between two consecutive words and the edges stand for the hypothetical handwritten words (initial edges are in bold in Figure 1). The task of the following module (see Section 4) is to create additional edges in the word graph (represented by dotted edges in Figure 1), based on the segmentation scores given by the previous module. Finally, the sentence recognition is performed with the Viterbi algorithm to find the likeliest paths corresponding to the N -best sentence list. For this task, our word recognition system RESIFMot (Carbannel & Anquetil, 2003) gives a list of 10 candidate words for each handwritten word supported by an edge and a language model is also used to provide the probabilities of word sequences.

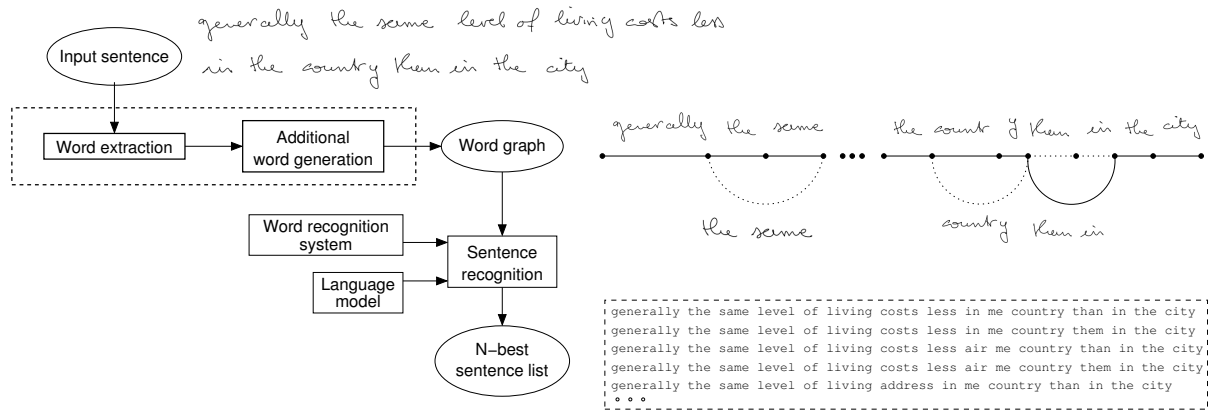


Figure 1. Sentence recognition system.

3. Word extraction

Our word extraction scheme consists in an inter-stroke gap classification task. Since our data are English handwritten sentences without any punctuation mark, we consider three kinds of gaps: the *intra-word* gap (between two strokes from the same word), the *inter-word* gap (between two strokes from two consecutive words) and the *inter-line* gap (between two strokes from two consecutive words written on two consecutive lines). As we work on Latin script, the first stroke of a new word is most likely on the right of the last stroke of the previous word. Let's consider a couple of strokes (S_{ref}, S_{new}), where S_{ref} denotes the last stroke that has been written and classified (also called *reference stroke*) and S_{new} the next written stroke, in the chronological order, that has to be classified. We propose to use the distance in x-coordinate Δx_{ref}^{new} between the most-on-the-right point P_{ref}^{mtr} of S_{ref} and the most-on-the-left point P_{new}^{mtl} of S_{new} to characterize the inter-stroke gap between S_{ref} and S_{new} (see Figure 2).

To deal with the problem of slant without applying costly preprocessing steps, we limit the evaluation of the distance to the strokes points contained between the lower and upper baselines. To detect these baselines, we run our detection algorithm on the strokes written before S_{new} . To avoid the skew problem, we only consider the last few strokes written before S_{new} . To determine the number of previous strokes to keep, we use the pertinent downstrokes (Anquetil & Lorette, 1997). Our detection algorithm thus needs a group of strokes that represents at least 3 or 4 characters to detect these baselines properly. As a character is composed of 1 to 3 pertinent downstrokes, we keep as many previous strokes to have at least 10 pertinent downstrokes in the group. Thereafter, this group of strokes associated to a S_{new} will be denoted *BRG* (Baseline Reference Group). In an "on-the-fly" recognition system, the *BRG* corresponds to a buffer containing the last written strokes. Each newly written stroke is analyzed and added to the buffer. The oldest stroke of the buffer is then removed if it remains at least 10 pertinent downstrokes. The word extraction process can start when at least 10 pertinent downstrokes have been written.

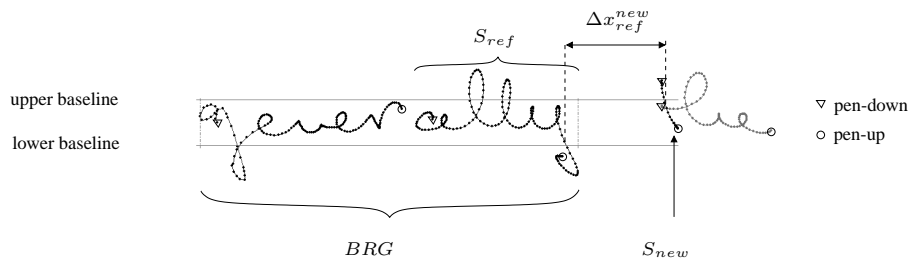


Figure 2. Example of Δx_{ref}^{new} and Baseline Reference Groupe (*BRG*).

Once Δx_{ref}^{new} has been computed, we use a Radial Basis Function Network (RBFN) to classify the inter-stroke gap. In addition to the Δx_{ref}^{new} input, we give to the RBFN three other inputs: the distance between the top of the S_{new} bounding box and the lower baseline (mainly used for the detection of the *inter-line* gaps), the maximum and the median Δx_{ref}^{new} in the current *BRG*. The RBFN outputs are the score obtained by the *intra-word*, *inter-word* and *inter-line* classes. The class with the highest score is then associated to the stroke.

Finally, we associate a confidence index to the result of the RBFN: it evaluates the reliability of the first answer. If this confidence index is too low, the second answer is then also considered. To design our confidence index, we use an ambiguity reject (Mouchère & Anquetil, 2006). We learn a threshold on the relative difference $diff_{2best}$ between the two best classes. If $diff_{2best}$ is below this threshold, the first answer of the RBFN must be reconsidered and additional extracted word hypotheses must be generated.

4. Word graph construction and sentence recognition

The initial word graph is built by considering only the first answer of the gap classification task. Additional extracted word hypotheses are then created using the second answer. To control the size of the final word graph (in terms of number of edges), some limitations are set. Firstly, inter-line gaps are supposed to be correctly identified. The other restrictions aims at dealing with either under-segmentations or over-segmentations.

To deal with potential under-segmentations, the number of potential under-segmented words represented by an edge is set to three *i.e.* an edge of the initial word graph cannot be separated into more than three parts. Thus, for every edge, the intra-word gaps with a positive Δx_{ref}^{new} and a $diff_{2best}$ below the rejection threshold are sorted according to their ascending $diff_{2best}$. Then, at most the two best intra-word gaps are considered as potential inter-word gaps and additional edges and nodes are created from the current edge, as shown in Figure 3(a) (illustrated by dotted edges and their corresponding nodes).

In the same way, initial edges are limited to be at most an over-segmentation of a word into three parts. Each group of three consecutive initial edges is then considered. When one or the two inter-word gaps have their $diff_{2best}$ below the rejection threshold, additional edges are created, as can be seen in Figure 3(b). Moreover, these edges can only be created if their total number of downstrokes is less than 25.



Figure 3. Creation of additional edges and nodes, for under-segmentation (a) and over-segmentation (b).

The Viterbi algorithm is finally performed on the whole graph to retrieve the N -best sentences $\{\hat{W}\}$ among all the sentences $W_k = w_{k,1} \dots w_{k,n_k}$, by combining graphic and linguistic information as given by equation 1:

$$\{\hat{W}\} = \left\{ \arg \max_{W_k} score(W_k) \right\} = \left\{ \arg \max_{W_k} \sum_{i=1}^{n_k} score(s_{k,i} | w_{k,i}) + \gamma \log [p(w_{k,i} | w_{k,i-n+1}^{i-1})] + \delta \right\} \quad (1)$$

where $score(s_{k,i} | w_{k,i})$ is the score of the part of the signal s_i corresponding to word $w_{k,i}$ and is estimated by the word recognition system; $p(w_{k,i} | w_{k,i-n+1}^{i-1})$ is the a priori probability of the word $w_{k,i}$ given its predecessor words $w_{k,i-n+1} \dots w_{k,i-1}$ and is given by a statistical language model. The *Grammar Scale Factor* γ weights the impact of the statistical language model whereas the *Word Insertion Penalty* δ controls the deletion and insertion of words.

5. Experiments and results

The language model is a bigram model built with the SRILM toolkit (Stolcke, 2002), from the Brown corpus (Francis & Kucera, 1979). This corpus contains 52,954 English sentences (1,002,675 words) where 46,836 sentences (900,108 words) were used to learn the language model. We use the associated lexicon including 13,748 words.

The handwritten material consists of handwritten sentences among the 2,598 remaining ones from the Brown corpus. The training set includes 488 sentences (7,650 words) written by 23 writers (this set is used to learn the RBFN for the inter-word gap classification and the rejection thresholds as well as to tune the values of the grammar scale factor and of the word insertion penalty) whereas the test set includes 395 sentences (6,038 words) written by 15 writers. The writers of the test set are different from those of the training set.

Different measures were used to evaluate the impact of the word extraction task on the performance of the recognition system. The graph density (GD) represents the average number of edges per word and shows the complexity of the word graph. The edge presence rate (EPR) is the percentage of edges in the word graph corresponding to edges in the ground truth word graph. The word presence rate (WPR) stands for the percentage of true words (*i.e.* words to recognize) which are in the word candidate lists given by the word recognition system. Finally, the word recognition rate (WRR) defines the percentage of correctly recognized words.

Word extraction strategy	GD	EPR	WPR	WRR
Ground truth	1.00	100.00 %	92.46 %	87.86 %
Word extraction with 0 % reconsideration	0.99	90.13 %	83.42 %	79.13 %
Word extraction with 8 % reconsideration	1.41	97.05 %	89.78 %	84.86 %
Word extraction with 20 % reconsideration	1.80	97.78 %	90.44 %	85.38 %
Word extraction with 40 % reconsideration	2.18	97.98 %	90.63 %	85.06 %
Word extraction with 60 % reconsideration	2.38	97.99 %	90.64 %	84.95 %

Table 1. Performance of the recognition system on the test set.

Table 1 summarizes the comparison between the word extraction associated with different reconsideration strategies as well as relatively to the ground truth. The reconsideration strategies only differ by the percentage of reconsidered gap classification results. The EPR is about 90 % for the word extraction with no reconsideration (and corresponds to the word extraction rate presented in Liwicki et al. (2006)). It exceeds 97 % for the word extraction associated with the other reconsideration strategies. The WPR and the WRR achieve also an absolute rise between 6 % and 7 %. This edge addition obviously leads to an augmentation of the GD but the graph densities remain reasonable. By comparing the different reconsideration strategies, we can see that beyond a certain percentage of reconsidered gap classification results, no significant improvements are obtained. Thus, the reconsideration strategy which allows the better tradeoff between the augmentation of the word recognition rate and the limitation of the graph density is the one where 20 % of the inter-stroke gap classification results are reconsidered. Indeed, the word recognition rate rises from 79.13 % (for the word extraction with no reconsideration) to 85.38 % (for the best reconsideration strategy) which corresponds to a 29.95 % word error rate reduction: relatively to the ground truth, it corresponds to a 71.59 % diminution.

6. Conclusion and perspectives

In this paper, we have presented an extension of our previous on-line sentence recognition system to a complete recognition system by integrating an automatic word extraction mechanism. Our word extraction strategy leans on the characterization of inter-stroke gaps. A reconsideration step is also added to reconsider not enough reliable gap classification results. This allows the generation of additional extracted words into the word graph on which the sentence recognition is performed. Different reconsideration strategies were thus compared between each other as well as toward the ground truth. This generation of extra hypotheses has led to an improvement of the recognition system performances, in terms of different indicators. Thus, with the best reconsideration strategy (which reconsiders 20 % of the inter-stroke gap classification results), the word error rate was reduced by 30 % relatively to the ground truth. With our presented word extraction strategy, the recognition system will easily be extended to perform an “on-the-fly” recognition.

7. References

- Anquetil, E., & Lorette, G. (1997). Perceptual Model of Handwriting Drawing, Application to the Handwriting Segmentation Problem. In *Proceedings of the International Conference on Document Analysis and Recognition* (pp. 112–117). Ulm.
- Carbonnel, S., & Anquetil, E. (2003). Lexical Post-Processing Optimization for Handwritten Word Recognition. In *Proceedings of the International Conference on Document Analysis and Recognition* (pp. 477–481). Edinburgh.
- Francis, W., & Kucera, H. (1979). *Brown Corpus Manual*.
- Liwicki, M., Scherz, M., & Bunke, H. (2006). Word Extraction from On-Line Handwritten Text Lines. In *Proceedings of the International Conference on Pattern Recognition* (pp. 929–9330). Hong-Kong.
- Mouchère, H., & Anquetil, E. (2006). A unified strategy to deal with different natures of reject. In *Proceedings of the International Conference on Pattern Recognition* (pp. 792–795). Hong-Kong.
- Oudot, L., Prevost, L., & Milgram, M. (2004). An Activation-Verification Model for On-Line Texts Recognition. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition* (pp. 485–490). Tokyo.
- Quiniou, S., & Anquetil, E. (2006). A Priori and A Posteriori Integration and Combination of Language Models in an On-line Handwritten Sentence Recognition System. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition* (pp. 403–408). La Baule.
- Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing* (pp. 901–904). Denver.
- Varga, T., & Bunke, H. (2005). Tree Structure for Word Extraction. In *Proceedings of the International Conference on Document Analysis and Recognition* (pp. 352–356). Seoul.