Handling out-of-vocabulary words and recognition errors based on word linguistic context for handwritten sentence recognition

Solen Quiniou Mohamed Cheriet
Synchromdia Laboratory - École de technologie supérieure
1100, rue Notre-Dame Ouest
Montréal (Québec) H3C 1K3, Canada
Solen.Quiniou@synchromedia.ca, Mohamed.Cheriet@etsmtl.ca

Eric Anquetil
IRISA - INSA
Campus de Beaulieu
35042 Rennes Cedex, France
Eric.Anquetil@irisa.fr

Abstract

In this paper we investigate the use of linguistic information given by language models to deal with word recognition errors on handwritten sentences. We focus especially on errors due to out-of-vocabulary (OOV) words. First, word posterior probabilities are computed and used to detect error hypotheses on output sentences. An SVM classifier allows these errors to be categorized according to defined types. Then, a post-processing step is performed using a language model based on Part-of-Speech (POS) tags which is combined to the n-gram model previously used. Thus, error hypotheses can be further recognized and POS tags can be assigned to the OOV words. Experiments on on-line handwritten sentences show that the proposed approach allows a significant reduction of the word error rate.

1. Introduction

Most of the handwritten text recognition systems are using a closed vocabulary. While this is suitable for tasks like recognizing literary handwritten texts, it is not well-adapted for new applications like recognizing freeform notes written on a TabletPC [14] or on a whiteboard [7]. Indeed, for this task, the vocabulary is generally infinite due to the use of personal abbreviations. Thus, when using a recognition system with a closed vocabulary, it is interesting to add a mechanism to detect out-of-vocabulary words. On one hand, these words might be further reconsidered using a recognition based on sub-word units like characters or strokes, for example. On the other hand, as the OOV words are not correctly recognized, they are substituted by words from the vocabulary, which has been shown to cause recognition errors on neighboring words due to the use of language models [3]. Thus, detecting and processing the OOV words could also allow the correction of other recognition errors and increase the performance of the whole system.

In some handwritten text recognition systems like [16], OOV words may occur but, to our knowledge, no strategy has been proposed to deal with them. Some recognition systems associate confidence scores to the output sentences to allow the rejection of some of the words. In [9], anti-letter models are used and, in [11], different confidence measures both at the letter and at the word levels are compared. Nevertheless, no linguistic information is used at the sentence level. A rejection strategy based on varying the weight of a language model (LM) is presented in [2] and relies on the assumption that non-recognized words are more sensitive to this variation. Nonetheless, the handwritten texts do not contain OOV words.

In the field of speech recognition, most recognition systems deal with OOV words. Among approaches to detect and recognize these words are phone-based models [1] (with specific models that may depend on the category of the OOV words) or recognition systems based on sub-word units [3]. In [6], an approach based on word posterior probabilities computed on a confusion network is proposed to detect OOV words. Finally, works on tagging texts containing OOV words rely on POS categories which are used in conjunction with n-gram LMs to achieve better results [10].

In this paper, we focus on the detection and on the post-processing of OOV words in an on-line handwritten sentence recognition system. Since these words may also cause other recognition errors, our proposed approach allows the detection of different kinds of recognition errors (including OOV words) and is performed on output sentences given by our baseline sentence recognition system [13]. We thus extend our previous works [12] using posterior probabilities as confidence scores on words of output sentences. We then use a classifier to identify the type of each error hypothesis thus detected. To allow the correction of errors that may be due to unrecognized neighboring words, we add a post-processing step using a *word-to-POS backoff LM* whose

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.78

aim is two-fold: improving the recognition of in-vocabulary words and associating OOV words with their POS category.

The remaining parts of this paper are as follows. In section 2, an overview of the whole recognition system is given. The proposed approach for detecting and characterizing error hypotheses is then presented in section 3. Section 4 describes the construction of the POS-based LM and its use to correct error hypotheses and to retrieve the categories of OOV words. Finally, experimental results are discussed in section 5 while section 6 draws some conclusions.

2. Recognition system overview

In this section, we present the different steps of our whole sentence recognition system (see figure 1).

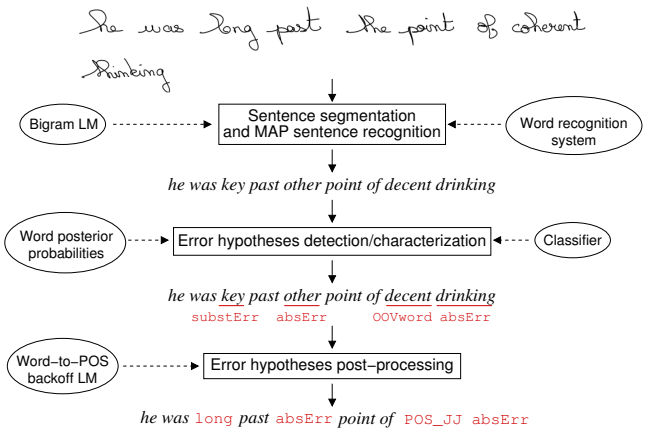

Figure 1. Sentence recognition system.

Given an input handwritten sentence, the sentence recognition system first builds a graph containing handwritten word segmentation hypotheses. To identify these hypotheses, a Radial Basis Function Network (RBFN) is used to classify inter-stroke gaps. A confidence index associated to each of these classification results is also used to create additional segmentation hypotheses (see [13] for further details). A Maximum A Posteriori (MAP) decoding is then performed on this graph to find the likeliest sentence $\hat{W}$ (called MAP sentence), using graphic and linguistic information on words as given by equation 1:

$$\hat{W} = \arg\max_{W_k} \; score(S|W_k) + \gamma \; \log\left[p(W_k)\right] + \delta n_k \quad (1)$$

where $score(S|W_k)$ is the score of the handwritten signal S for the given sentence W_k, estimated by the recognition system: it combines graphic and lexicon scores given by our word recognition system [4]. The *graphic score* includes adequation measures between each character and its corresponding model as well as spatial and statistical information between characters and the *lexicon score* depends on edit operations performed during the lexical post-processing step. $p(W_k)$ is the *a priori* probability of the sequence W_k, given by a bigram LM and n_k is the number of words in W_k. The weight γ is used to balance the influence of the LM against the score from the recognition system whereas δ controls the deletion and insertion of words.

Error hypotheses are then detected using the posterior probabilities of the MAP sentence words. A classifier is then used to characterize each so-detected error hypothesis into four types (*OOVword*, *segErr*, *absErr* and *substErr*). This whole approach is presented in section 3.

Finally, a post-processing step using a word-to-POS backoff LM is performed on the MAP sentence given the error hypotheses types. It allows the correction of some of the errors as well as the identification of the POS category of the OOV words (in figure 1, the OOV word is identified as an adjective). This is described in section 4.

3. Identification of OOV words and other recognition errors

In this section, we describe how error hypotheses are detected on MAP sentences and how they are further characterized into four different error types.

3.1. Detection of error hypotheses using word posterior probabilities

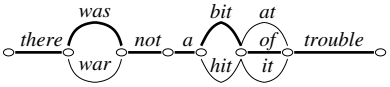

Figure 2. Example of a confusion network.

Word posterior probabilities are computed on a confusion network representation of the word graph [8]; figure 2 shows an example of a confusion network (edges in bold correspond to the MAP sentence). In this representation, nodes represent confusions between word hypotheses, for a given position in the input sentence, and adjacent nodes are linked by as many edges as word hypotheses and define *confusion sets*. Each word is associated with a *posterior probability* corresponding to the sum of the probabilities of all the graph paths that contain this word. The word posterior probabilities are computed as described in [12].

The word posterior probabilities thus integrate graphic and linguistic information and can be used as confidence score on the words. To detect error hypotheses on a MAP sentence, we compare the posterior probability of each of its words to a learnt threshold σ_{err}: words whose probability is below σ_{err} are considered as error hypotheses. In fact, in our approach, we use the difference between the posterior probabilities of the considered MAP word and of the second likeliest word in the corresponding confusion set (in these sets, words are ranked according to these probabilities).

3.2. Characterization of error hypotheses

Now that error hypotheses have been detected, it is interesting to identify why the corresponding MAP word has not been correctly recognized and especially to know if it corresponds to an OOV word. Moreover, this characterization can further allow different strategies to try to correct these errors. Here, we considered four error types: (i) *OOV words*, (ii) *segmentation errors i.e.* the MAP word does not correspond to a correctly segmented part of the sentence, (iii) *absent words i.e.* the correct word does not appear in the confusion set where the word MAP belongs, and (iv) *substitution errors i.e.* the correct word appears in the confusion set but does not correspond to the MAP word (respectively named *OOVword*, *segErr*, *absErr* and *substErr*).

To categorize error hypotheses, we use a Support Vector Machine (SVM) with a gaussian kernel. SVMs have been chosen because of their efficiency and their capacity to deal with unbalanced classes (in terms of training examples). Here, we consider two sets of features. The *baseline set* corresponds to the one used in our previous work [12] and includes 6 features: 3 for the considered MAP word and the same 3 for the second likeliest word of its corresponding confusion set. These 3 features are the posterior probability of the word, its normalized graphic score and its normalized lexicon score (see section 2 for a description of these two latter scores). In the *extended set*, 4 additional features are considered: the length of the MAP word, a boolean feature indicating if the word segmentation was initially created (or if it was additionally generated, as presented in section 2) and 2 boolean features indicating whether or not the neighboring words (on the left and on the right) are detected as error hypotheses. The two latter features were inspired by [6].

4. Identification of OOV word categories and correction of substitution errors

Error hypotheses are now reconsidered using a word-to-POS backoff LM; this post-processing only deals with substitution errors and OOV words. In this section, we describe how this LM is created and used to correct error hypotheses.

4.1. Adapting a POS-based LM to handle OOV words

Category-based LMs provides the probability of a word w_i, given its previous words on the sentence and according to the classes of each word. When POS categories are considered, a word may belong to several ones because they represent the grammatical nature of the words which depends on the context of the word. Two approaches can be used to take into account the classes of words by either considering all the possible class sequences of a given word sequence or by only considering the likeliest class sequence. Since we want to retrieve the classes corresponding to OOV words, we choose the latter approach. Equation 2 gives the probability of a word w_i with its context h_i:

$$p(w_i|h_i) = \max_{c_{i-n+1}^{i} \in C_i \times \ldots \times C_{i-n+1}} p(w_i|c_i)\, p(c_i|c_{i-n+1}^{i-1}) \quad (2)$$

where n is the order of the LM, $h_i = w_{i-n+1}^{i-1}$ is the history of word w_i and C_j is the class set of each word w_j.

To handle OOV words in the POS-based LM, POS categories for these words have first to be determined. One straightforward solution would be to allow these words to belong to any POS category but, since there are more than one hundred POS classes (see section 5), the computation of the whole sentence probability would be hardly manageable. To define the class set of OOV words, we allow them to only belong to *open classes* such as nouns, adjectives, verbs or adverbs. Indeed, since OOV words are in perpetual creation in a given language, they are most likely to correspond to one of these open classes words (as opposed to *closed classes* including determiners, pronouns, prepositions, conjunctions and auxiliary verbs, which are in a limited number in a given language).

Now, for each OOV word, the probability in each of its classes has to be determined. As no *a priori* knowledge on belonging to one particular class is given, these probabilities are equal, as given by equation 3:

$$(\forall w_{OOS})\,(\forall c_i \in C_{OOS})\;\; p_c(w_{OOS}|c_i) = K_{OOV} \quad (3)$$

where w_{OOV} is an OOV word, C_{OOS} is the class set for the OOV words and K_{OOV} is a constant.

Whereas the initial probabilities $p(c_i|c_{i-n+1}^{i-1})$ remain the same in the extended POS-based LM, the initial probabilities $p(w_i|c_i)$ have to be modified for the classes $c_i \in C_{OOV}$. Indeed, to ensure $\sum_{w_j} p_c(w_j|c_i) = 1$ $(\forall c_i \in C_{OOS})$, the probability of each in-vocabulary word belonging to one of the C_{OOS} classes is reduced according to the K_{OOV} constant. The modified probability is given by equation 4:

$$p_c(w_i|c_i) = p(w_i|c_i) - \frac{K_{OOV}}{n_i}. \quad (4)$$

where n_i is the number of in-vocabulary words in class c_i.

4.2. Post-processing using a word-to-POS backoff LM

The aim of this post-processing step is both to correct substitution error hypotheses and to assign POS categories to OOV words. To do so, we use the POS-based LM previously presented and we combine it to an n-gram language model, based on [10] (the whole LM is called *word-to-POS backoff LM*). Thus, we use the POS-based LM instead of the n-gram LM when the history of the current word w_i

contains at least one detected OOV word. The probability of a word w_i is then given by equation 5:

$$p_{wc}(w_i|h_i) = \begin{cases} p_w(w_i|\Phi(h_i)) & \text{if } w_{i-n+1}^{i-1} \in V^{n-1} \\ \\ p_c(w_i|\Phi(h_i)) & \text{else} \end{cases} \quad (5)$$

where V is the vocabulary, $p_w(.)$ is the probability given by the n-gram LM and $p_c(.)$ is the probability given by the POS-based LM. $\Phi(h_i)$ is the history reduced to the n' last words ($n' \leq n-1$) so it does not contain any error hypotheses identifying as absent words or segmentation errors.

To use the probability defined by equation 5, we generate a simplified word graph from the MAP sentence, using the error types previously detected. Since segmentation errors are not taken into account, the segmentation of the MAP sentence is not reconsidered. In this simplified graph, words identified as segmentation errors or OOV or absent words are replaced by their error type whereas, for substitution errors, all the words of the corresponding confusion set are added to the graph. For other words of the MAP sentence, *i.e.* not detected as error hypotheses, only the MAP word is added to the graph. Finally, the path corresponding to the likeliest sentence is retrieved on this graph, using equation 1 where the word-to-POS backoff LM gives the probability $p(W_k)$ for each sentence. Segmentation errors and absent words remain in this final sentence whereas OOV words are replaced by their POS categories (see figure 1).

5. Experiments and results

In this section, we first describe the experimental setup and then we present the results of the experiments on the detection and characterization of error hypotheses as well as on the use of the designed word-to-POS backoff LM.

5.1. Experimental setup

The language models are built on the Brown corpus [5] using the SRILM toolkit [15]. This corpus contains 52,954 sentences (1,002,675 words) where 46,836 sentences (900,108 words) were used to learn the LMs. For the POS-based LM, we use the tagged version of the Brown corpus, containing 145 POS tags. 25 of these POS tags were considered as possible classes for the OOV words.

The handwritten material consists of sentences written from 2,598 sentences of the Brown corpus (corresponding to the sentences not considered for the LMs learning). The training set includes 557 sentences (8,769 words) written by 25 writers (it is used to learn the SVMs, to tune the parameters σ_{err}, γ, δ and to compute the word posterior probabilities) whereas the test set contains 460 sentences (7,080 words) written by 17 writers. The writers of the test set are different from those of the training set.

To consider OOV words, we use a lexicon reduced to the 5,000 most frequent words of the vocabulary closed on the Brown corpus (containing 44,101 words). The words of the handwritten sentences that do not belong to this lexicon are then considered as OOV words. Thus, the OOV word rate is 5.5% on the training set and 5.6% on the test set.

5.2. Detection of error hypotheses

The parameters used to compute word posterior probabilities are optimized toward the normalized cross-entropy (NCE), commonly used to measure the quality of confidence scores. With confidence values clipped at 0.05 and 0.95 (to avoid negative NCE values, as suggested by [6]), the NCE is 0.25. The difference between the posterior probabilities of the MAP word and of the second likeliest word is then used as a confidence score to detect error hypotheses.

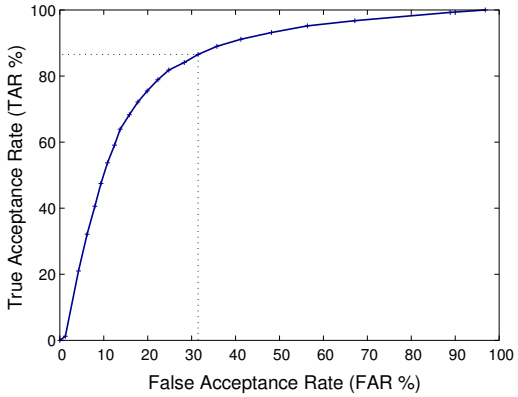

Figure 3. ROC curve for the detection of word error hypotheses.

Figure 3 plots the ROC curve for different thresholds σ_{err} on this confidence score: it shows the compromise between correctly recognized words whose confidence score is above σ_{err} (TAR) and error hypotheses whose confidence score is above σ_{err} (FAR). The chosen threshold $\sigma_{err} = 0.3$ corresponds to a 88.2% TAR and a 33.0% FAR and allows the detection of 71.4% of the OOV words.

5.3. Characterization into error types

Table 1 gives the rates of correct characterization of the detected error hypotheses into the four error types considered, using either the baseline or the extended feature set.

Table 1. Characterization rates by error type.

Feature set	OOV words	Subst. errors	Absent words	Segment. errors
Baseline	18.8 %	62.5 %	63.9 %	41.8 %
Extended	56.1 %	59.2 %	73.7 %	65.5 %

These rates are given on the test set where only error hypotheses are considered (thus corresponding to the ground truth of a perfect detection step). Using the extended feature set highly improves the characterization of OOV words which are twice as less mistaken for absent words or segmentation errors than when using the baseline set. Furthermore, absent words are less mistaken for segmentation errors. Nonetheless, the characterization rate of substitution errors is slightly reduces (with more confusion with absent words or substitution errors). Finally, the global characterization rate, among all the error hypotheses, is 65.2% with the extended feature set and 49.2% with the baseline set.

5.4. Evaluation of the overall post-processing strategy

Table 2 gives the word recognition rate as well as the word error rates on error hypotheses identified using the proposed approach and on residual errors (corresponding to error hypotheses not detected by the approach).

Table 2. Recognition and error rates for the overall approach.

Strategy	Recognition rate	Identified error rate	Residual error rate
MAP	77.7 %	0.0 %	22.3 %
Error ident.	68.4 %	23.1 %	8.5 %
Error ident. + correction	73.8 %	14.5 %	11.7 %

Relatively to the baseline system (using only the MAP-based recognition approach), the recognition rate achieved with the error detection approach is decreased but the remaining word error rate is greatly reduced, corresponding to a 61.3% relative reduction. The detected error hypotheses are distributed as follows: 17.0% of segmentation errors, 30.9% of absent words, 40.2% of substitution errors and 11.9% of OOV words. When the post-processing step is added (using a 4-class LM based on POS categories, combined to the bigram LM used in the baseline system), the word recognition rate is increased by 5% thanks to the recognition of 42.5% of the substitution errors previously identified. Furthermore, the POS categories of 33.8% of the OOV words are correctly retrieved.

6. Conclusion

In this paper, we have presented an approach to identify word error hypotheses on sentences (given by a MAP recognition approach) and to further correct or associate them with their POS categories. Word posterior probabilities are used as confidence scores to detect error hypotheses and to characterize them into four types with an SVM (using also other features). A post-processing step using a word-to-POS backoff LM is then performed to correct substitution errors and to associate POS categories to the OOV words detected. This approach allows the reduction of the word error rate and the correct identification of the POS categories of some of the OOV words.

Future works will investigate using additional features to better characterize the error types. Furthermore, segmentation errors will be considered (using alternate segmentation hypotheses in the simplified word graph) as well as errors due to absent words. Moreover, POS categories of OOV words will be used to try to recognize these words (to select an appropriate lexicon, for example).

References

[1] I. Bazzi. *Modelling Out-Of-Vocabulary Words for Robust Speech Recognition*. PhD thesis, MIT, 2002.

[2] R. Bertolami, M. Zimmermann, and H. Bunke. Rejection strategies for offline handwritten text recognition. *Pattern Recognition Letters*, 27:2005–2012, 2006.

[3] M. Bisani and H. Ney. Open vocabulary speech recognition with flat hybrid models. In *Interspeech'05*, pages 725–728.

[4] S. Carbonnel and E. Anquetil. Lexicon Organization and String Edit Distance Learning for Lexical Post-Processing in Handwriting Recognition. In *IWFHR'04*, pages 462–467.

[5] W. Francis and H. Kucera. *Brown Corpus Manual*. Brown University, 1979.

[6] D. Hillard and M. Ostendorf. Compensating for Word Posterior Estimation Bias in Confusion Networks. In *ICASSP'06*, pages 1153–1156.

[7] M. Liwicki and H. Bunke. Handwriting Recognition of Whiteboard Notes – Studying the Influence of Training Set Size and Type. *IJPRAI*, 21(1):83–98, 2007.

[8] L. Mangu. *Finding Consensus in Speech Recognition*. PhD thesis, Johns Hopkins University, 2000.

[9] S. Marukatat, T. Artières, and P. Gallinari. Rejection Measures for Handwriting Sentence Recognition. In *IWFHR'02*, pages 24–29.

[10] T. Niesler. *Category-Based Statistical Language Models*. PhD thesis, University of Cambridge, 1997.

[11] J. Pitrelli, J. Subrahmonia, and M. Perrone. Confidence Modeling for Handwriting Recognition: Algorithms and Applications. *IJDAR*, 8(1):35–46, 2006.

[12] S. Quiniou and E. Anquetil. Use of a Confusion Network to Detect and Correct Errors in an On-line Handwritten Sentence Recognition System. In *ICDAR'07*, pages 382–386.

[13] S. Quiniou, F. Bouteruche, and E. Anquetil. Word Extraction Associated with a Confidence Index for On-Line Handwritten Sentence Recognition. *IJPRAI*, 2009. to appear.

[14] M. Shilman, Z. Wei, S. Raghupathy, P. Simard, and D. Jones. Discerning Structure from Freeform Handwritten Notes. In *ICDAR'03*, pages 60–65.

[15] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *ICSLP'02*, pages 901–904.

[16] A. Vinciarelli, S. Bengio, and H. Bunke. Offline Recognition of Unconstrained Handwritten Texts using HMMs and Statistical Language Models. *PAMI*, 26(6):709–720, 2004.