Statistical Hypothesis Testing for Handwritten Word Segmentation Algorithms

Mehdi Haji
CENPARMI, Concordia University
Montréal, Canada
m_haji@encs.concordia.ca

Kalyan Asis Sahoo
Department of Mathematics, IIT Kharagpur
West Bengal, India
kalyaniitkgp07@gmail.com

Tien. D. Bui
CENPARMI, Concordia University
Montréal, Canada
bui@cs.concordia.ca

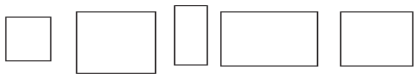
Ching Y. Suen
CENPARMI, Concordia University
Montréal, Canada
suen@cs.concordia.ca

Dominique Ponson
IMDS Software Inc.
Montréal, Canada
dponson@imds-world.com

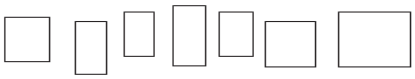
***Abstract*—We present a statistical hypothesis testing method for handwritten word segmentation algorithms. Our proposed method can be used along with any word segmentation algorithm in order to detect over-segmented or under-segmented errors or to adapt the word segmentation algorithm to new data in an unsupervised manner. The main idea behind the proposed approach is to learn the geometrical distribution of words within a sentence using a Markov chain or a Hidden Markov Model (HMM). In the former, we assume all the necessary information is observable, where in the latter, we assume the minimum observable variables are the bounding boxes of the words, and the hidden variables are the part of speech information. Our experimental results on a benchmark database show that not only we can achieve a lower over-segmentation and under-segmentation error rate, but also a higher correct segmentation rate as a result of the proposed hypothesis testing.**

I. INTRODUCTION

Words are the building blocks of text. In document understanding applications, we often need to divide a document into its constituent lines, and further to divide each line into its constituent words. Due to its practical importance, word segmentation has been an ongoing topic of research in the document analysis community and there are numerous methods that have been proposed over the last decades to address this problem [1]. Nevertheless, the segmentation of words in unconstrained handwritten documents remains a challenging task mainly because the boundaries between words are not well-defined without knowing their meanings. Inter-word-spacing is sometimes wider than the intra-word-spacing and thus it is not always possible to perfectly segment the document at the word level using geometrical information only.



(a) Hypothesis #1



(b) Hypothesis #2

(c) Hypothesis #3

Figure 1. Three word segmentation hypotheses for a text line represented as lists of bounding boxes.

Considering the fact that both the segmentation and the recognition information are unknown for a given document image, there are two major approaches to address the word segmentation problem: implicit and explicit. In the former, the segmentation and recognition are done simultaneously; in other words, the word boundaries are found as a by-product of a sentence recognition algorithm [2, 3]. In the latter, the segmentation is done as an independent step before recognition [4-6].

No matter what type of word segmentation algorithm is used, the output of the algorithm can be thought of as a list of rectangles corresponding to the bounding boxes of the words in the input text line. The main motivation behind this work is to find a statistical testing method in order to detect unlikely

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.272

segmentation hypotheses given that the minimum information available to the testing method is the coordinates of the bounding boxes. An example is shown in Fig. 1. We see three word segmentation hypotheses for an input text line. Given that the rectangles correspond to words, we can tell that Hypothesis #1 is more likely to be the correct segmentation compared to Hypotheses #2 and #3. Hypothesis #2 is most likely over-segmented because it is rare that a long sentence is composed of many consecutive short words. Hypothesis #3 is most likely under-segmented because it is rare that a long sentence is composed of only two long words. The idea is to learn a statistical model from a set of correctly segmented lines so that it assigns higher probabilities to more likely hypotheses, and lower probabilities to highly over-segmented and highly under-segmented hypotheses. To the best of our knowledge, this work is the first to propose a trainable word segmentation hypothesis testing method for handwritten documents.

In the rest of this paper, firstly, we will talk about the choice of the model for the distribution of words, and present the training of the proposed models based on a standard database of English sentences. Secondly, we will show how to detect over-segmentation and under-segmentation errors and how to adapt the free parameters of a word segmentation algorithm to new data using the trained models. Finally, we will present our experimental results.

II. MODELING OF WORDS DISTRIBUTION

We consider the word segmentation process as a discrete-time stochastic phenomenon that satisfies the Markov property. The Markov property obviously holds because the unidirectional property of text lines implies that the conditional probability distribution of future words only depends on the current word, and not the preceding or future words. Therefore, depending on whether we assume all variables in the problem are observable or not, there are two general strategies for the modeling of the distribution of words in a text line: observed-variable approach and latent-variable approach. In the former, we model the words distribution based on a Markov chain; while in the latter, we model the words distribution using a Hidden Markov Model (HMM).

Assuming that the minimum information available to the testing method is the bounding boxes of the words, there are a number of other variables in this problem that are considered as hidden (i.e. they are not directly observable). The HMM framework allows us to somehow infer these hidden variables from the observable variables. We may consider as hidden variables the meanings or shapes of the words, the context of the writing, the author's writing style, number of letters in the words, part of speech information etc. Out of these hidden variables, it is quite meaningful to associate the number of letters in the words and the part of speech information with the observable variables (i.e. the bounding boxes in the simplest case). The reason we are interested in the part of speech information is mainly prepositions (on, in, to, by, for, with, at, of, from, as, …) and pronouns (I, me, we, us, you, him, it, …) that are typically short length words. Therefore, by inferring the part of speech information from the observable variables, we want to enhance the distribution model so that it can, to some extent, distinguish sequences of short length prepositions or pronouns from sequences of over-segmented words.

In the following, first we will present the Markov chain and then the hidden Markov model for the distribution of words.

A. *Markov Chain*

A Markov chain is the simplest Markov model where the system states are fully observable. We represent a Markov chain by a pair $\mu = (S, A)$ where S denotes the state space, and A denotes the transition matrix that defines the probability of going from any state to any other state of the system. The transition matrix defines the random process that is governed by the model, or equivalently, the distribution of state sequences that are generated by the model.

For the modeling of the distribution of words using Markov chains, first we have to identify the state space. Depending on the type of the word segmentation algorithm (implicit or explicit), there are two ways to define the state space, either based on: 1) the bounding box information; or 2) the transcription information. These two state spaces lead to two Markov chain models.

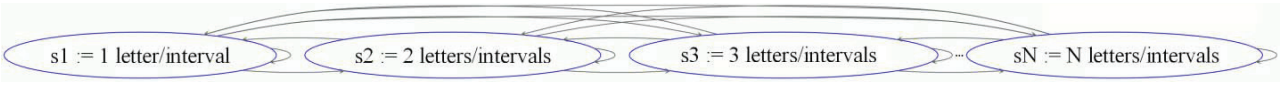


Figure 2. Markov chain model for distribution of words.

In the Markov chain model based on the bounding box information, we discretize the words bounding boxes into a number of non-overlapping equi-length frames (i.e. windows), and represent each frame with a separate state. Let $H_L = [\ b_1, b_2, \ldots, b_n\ ]$ denote a word segmentation hypothesis for a text line L, where b_i's represent the bounding boxes of the words hypotheses. We define the average height of the line avg(height(H_L)) as the average of the heights of the bounding boxes in H_L. Then, we define the number of frames for a word bounding box b_i as the closest integer to the ratio of the length of b_i to the average height of the line.

In the Markov chain model based on the transcription information, we represent each letter in a word with a separate state; therefore, we assume that the number of letters in a word image is an observable variable for the model. If the transcription information for the test data is not available, we have to estimate the number of letters from the word image. For this purpose, we use a regression neural network that is trained based on the Average Number of Transition (ANT) features [7].

We denote the set of states by $S = \{\ s_1, s_2, \ldots s_N\ \}$, where s_i is the state corresponding to words with i frames or i letters. In both models, we limit the number of states to a predefined maximum N_{max}. Therefore, any word with more than N_{max} frames/letters is represented by s_{Nmax}. The Markov chain model for distribution of words is shown as a directed weighted graph in Fig. 2, where the nodes represent the states and the edges represent the transition probabilities between the states.

For the training of the Markov chain, that is to find the transition probabilities, we use the standard IAM database [8] for both the bounding box-based and the transcription-based models. The ground-truth data of the IAM database is available at both the bounding box level and the transcription level for words and lines. In our experiments, we set N_{max} to 15.

B. Hidden Markov Model

A Hidden Markov Model (HMM) can be thought of as a Markov chain with unobserved (i.e. hidden) states, where in each state the model generates an output token based on a stochastic process. In HMM modeling, we assume that it is only the sequence of output tokens that we observe, but not the underlying sequence of states. In other words, the most likely sequence of states has to be inferred from the sequence of output tokens.

We represent a HMM by a 5-tuple $\lambda = (S, A, V, B, \prod)$, where S and A denote the state space and the state transition matrix that belong to the underlying Markov chain model. The three other elements are defined as follows: $V = \{\ v_1, v_2, \ldots v_M\ \}$ is the set of the observation symbols; B is the emission matrix that defines the probability of observing any observation symbol at any given state; and $\prod$ is the set of initial state probabilities, that defines the chance of any state as being the first in the sequence of states that corresponds to the sequence of output symbols.

1) Specification of HMM for Modeling of Words Distribution

In the HMM-based approach to hypothesis testing for word segmentation algorithms, the hidden states correspond to the part of speech information (i.e. linguistic categories). We use all of the nine traditional linguistic categories that have been defined for English words, which are: article, noun, pronoun, adjective, verb, adverb, preposition, conjunction and interjection.

As for the observation symbols, we may use either the number of frames or the number of letters in each word (which is available from the transcription information or estimated from the image). Similar to the Markov chain models that we discussed before, these two different observation spaces lead to two different HMM models, which we will refer to as the bounding box-based HMM and the transcription-based HMM.

For the training of the HMM models, we used the standard IAM database; as the ground-truth data contain the part of speech information, beside the bounding box and transcription information. We used the standard Baum-Welch algorithm [9] in order to estimate the initial, transition and emission probabilities. However, as the Baum-Welch algorithm is based on a local optimization strategy, it is important to start the optimization process with good initial guesses in order to avoid local minima. For this purpose, we estimated the initial guesses for the transition and emission matrices based on a few documents of the IAM database. We used Laplace (a.k.a. additive) smoothing [10] in order to avoid zero probabilities for unknown events (i.e. events that do not appear in a limited set of training data).

III. HYPOTHESES TESTING

In the following we describe how to detect over-segmented or under-segmented text lines using the words distribution models that we presented in the previous section.

Let $O = [\ o_1, o_2, \ldots o_T\ ]$ be an observation sequence corresponding to a word segmentation hypothesis $H_L = [\ b_1, b_2, \ldots, b_T\ ]$ for a text line L, where each o_i is the number of frames or letters corresponding to a word b_i. Obviously, if only the bounding box information is available, we have to use the bounding box-based

models. However, if the transcription information is available as well, or if the number of letters can be estimated from word images, we can use the transcription-based models.

In the Markov chain model, we already know the system state corresponding to each observation symbol. Therefore, the probability of the observation sequence can simply be computed as follows:

$$P(O \mid \mu) = \prod_{i=1}^{T-1} A(s_{o_i}, s_{o_{i+1}}) \quad (1)$$

In the HMM model, $P(O \mid \lambda)$ can be computed using the forward or backward algorithms [9].

Having obtained the probability of observation sequence for a word segmentation hypothesis H, we have to determine whether H must be accepted as good segmentation or not. For this purpose, we need three statistical populations: a population of perfectly segmented lines D_p; a population of over-segmented lines D_o; and a population of under-segmented lines D_u. In the following, first we describe the automatic generation of these three populations based on the IAM database, and then, the process of threshold selection for hypothesis testing.

A. Automatic Generation of Training Data

Having a collection of training documents that contain the transcription/bounding box information at line/word level, D_p is readily available from the ground-truth data. For the generation of D_o and D_u, first, we estimate the average intra-word distance w_{intra} and the average inter-word distance w_{inter} by 2-means clustering of the distances between all neighboring connected components in a document. Then, in order to generate over-segmented data, we merge neighboring connected components that are closer than a percentage of w_{intra}; and in order to generate under-segmented data, we merge neighboring connected components that are closer than a percentage of w_{inter}

.

B. Threshold Selection for Hypothesis Testing

Having obtained D_p, D_o, and D_u, given a words distribution model ψ, we simply define three populations of words distribution probability corresponding to perfectly segmented lines, over-segmented lines and under-segmented lines as follows, respectively referred to as P_{Dp}, P_{Do}, and P_{Du}:

$$P_{Dp} = \{ P(d_i \mid \psi) : \forall\, d_i \in D_p \} \quad (2)$$
$$P_{Do} = \{ P(d_i \mid \psi) : \forall\, d_i \in D_o \} \quad (3)$$
$$P_{Du} = \{ P(d_i \mid \psi) : \forall\, d_i \in D_u \} \quad (4)$$

Now, for the detection of a correctly segmented line from an incorrectly segmented line, we simply set the threshold to a value that minimizes the empirical classification error between P_{Dp} and $P_{Do} \cup P_{Du}$.

IV. UNSUPERVISED ADAPTATION OF WORD SEGMENTATION ALGORITHM USING WORDS DISTRIBUTION MODEL

A word segmentation algorithm typically has one or more parameters that are manually or automatically adjusted over a limited set of training documents. Using the words distribution models that we descried in Section 2, it is possible to automatically adapt a word segmentation algorithm to new data sets without the need for the ground-truth information.

Given that we have a range of valid values for each parameter and a model of words distribution, we can adapt the algorithm to a new data set (e.g. a new document) by finding the combination of parameters that gives the best word segmentation probability over the new data set. In general, there is no guarantee that the search space is linear or convex; therefore, it is better to conduct an exhaustive search over the space of all possible combinations of values in order to find the global optimum. The exhaustive search approach is possible only if the search space is small, which is normally the case here, as a typical word segmentation algorithm has only few parameters that need to be adjusted.

V. EXPERIMENTAL RESULTS

We evaluated the proposed hypothesis testing method over a set of unseen documents from the IAM database. Fig. 3 shows numerical examples of the log probabilities of words distribution computed using the bounding box-based Markov chain and HMM models for three segmentations of a text line. As can be seen, both models assign much higher probabilities to the correctly segmented hypothesis, compared to over-segmented and under-segmented hypotheses. It is interesting to observe that the margin between the correct and incorrect hypotheses is larger when the probabilities are computed by the HMM model.

In order to estimate the performance of the proposed models in the word segmentation context, we evaluated the proposed over/under-segmented detection method on a test set containing 2500 correctly segmented text lines and 2500 incorrectly segmented text lines. The over/under-segmented text lines were generated by applying the method described in Section 3.1 to the test database. The results are summarized in Table I in terms of the average detection rate of correctly segmented lines R_c, average detection rate of incorrectly segmented lines R_i, and the

harmonic mean (F-measure) of these two rates $F_{RcRi} = 2R_c.R_i / (R_c + R_i)$.

The detection rates achieved using the HMM models are higher than the Markov chain models as we expect. The transcription-based HMM achieves a very high performance for the detection of correctly segmented lines. However, the detection rate for over/under-segmented lines is lower. This is because the geometrical distribution of some incorrectly segmented lines overlaps that of correctly segmented lines, particularly if the line is over-segmented and under-segmented at the same time as shown in Fig. 4.

This example suggests that in some applications, such as word spotting, we may have to generate more than one segmentation hypothesis in order to make sure that the union of all hypotheses contain the right boundaries for all words. In the word spotting system proposed in [11], the authors use 10 different distance thresholds to generate words segmentation hypotheses for each text line. This fixed number of thresholds may result in a number of unlikely hypotheses and thereby unnecessary computations on the word spotting side. However, using the words distribution models that we proposed, one can easily reduce unlikely hypotheses prior to word spotting, which besides the reduction of the computation time, may result in the reduction of the false positive rate as well.

We used the proposed hypothesis testing as a post-processing step for a typical explicit gap-based word segmentation algorithm. The main idea behind gap-based algorithms is to connect (i.e. consider as part of the same word) all connected components that are closer than a certain threshold T_w. The threshold is dynamically adjusted based on the properties of the text. One common way of adjusting the threshold is based on the estimates of intra-word and inter-word distances [11]. Our experimental results show that a threshold value that is set to the weighted mean of w_{intra} and w_{inter} with much higher weight for w_{intra} gives better segmentation results than a threshold value that is set to the arithmetic mean of w_{intra} and $w_{inter.}$ In our experiments, we set $T_w = 0.9\ w_{intra} + 0.1\ w_{inter}$. The correct segmentation rate achieved by the baseline algorithm over the test database was 90.3%, which was increased to 91.5% using the proposed unsupervised adaption technique. Also, the over-segmentation and under-segmentation error rates were reduced by a factor of 2, i.e. the proposed hypothesis testing method was able to automatically detect around 50% of the segmentation errors in the output of the word segmentation algorithm.

VI. CONCLUSION

We presented statistical models for the distributions of words in text lines. We studied both the observed-variable approach and latent-variable approach. We presented the training of the proposed models based on a standard database of handwritten forms, and then used the trained models for the detection of over-segmentation and under-segmentation errors in the output of a word segmentation algorithm, and also for the unsupervised adaptation of the free parameters of the word segmentation algorithm to new data.

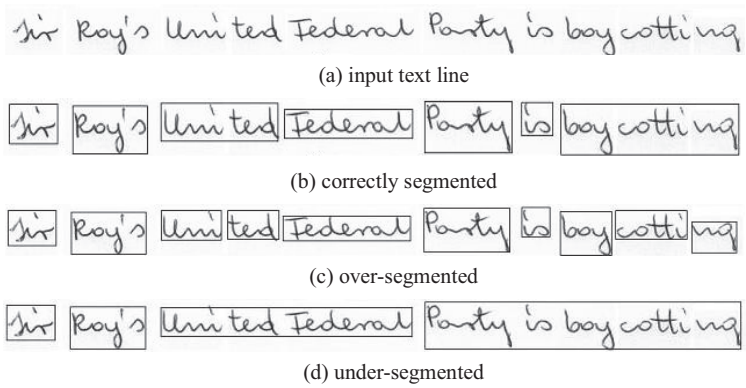
The main advantage of our proposed method is to provide a framework for adding prior knowledge about the grammar of the language, using the HMM model, to any word segmentation algorithm. Our experimental results showed that, although perfect segmentation is not always possible without using a larger context, but using the proposed method as a post-processing step for a word segmentation algorithm, we are able to increase the correct segmentation rate and the reliability of the algorithm by the automatic detection of unlikely segmentation hypotheses.

ACKNOWLEDGMENT

The authors would like to thank the MITACS and NSERC of Canada for financial support of this research through the MITACS Accelerate Award and the CRD grant.

TABLE I. Performance of proposed models for detection of correctly and incorrectly segmented text lines.

Performance	**Markov Chain**		**HMM**	
	bounding box-based	***transcription-based***	***bounding box-based***	***transcription-based***
Detection rate for correct segmentation	93.7%	93.2%	97.4%	99.1%
Detection rate for incorrect sgmntation.	71.1%	72.3%	80.9%	82.6%
Harmonic mean	80.9%	81.4%	88.4%	90.1%



(a) input text line

(b) correctly segmented

(c) over-segmented

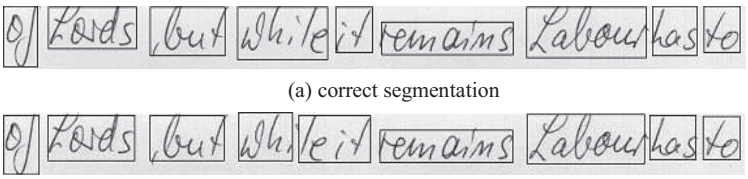
(d) under-segmented

Log probability:
Bounding box-based HMM: -19.60
Bounding box-based Markov Model: -18.53

Bounding box-based HMM: -22.41
Bounding box-based Markov Model: -20.74

Bounding box-based HMM: -25.37
Bounding box-based Markov Model: -24.51

Figure 3. Words distribution probability corresponding to three different segmentation of a text line.



(a) correct segmentation

(b) wrong segmentation

Log probability:
Bounding box-based HMM: -20.40
Transcription-based HMM: -22.71

Bounding box-based HMM: -20.40
Transcription -based HMM: -22.66

Figure 4. Example of a correct and a wrong segmentation for a text line that have been assigned almost the same distribution probability.

REFERENCES

[1] Gatos, B.; Stamatopoulos, N. & Louloudis, G., ICFHR 2010 Handwriting Segmentation Contest, Proceedings of the 2010 12th International Conference on Frontiers in Handwriting Recognition, IEEE Computer Society, 2010, 737-742.

[2] Zimmermann, M.; Chappelier, J.-C. & Bunke, H. Offline Grammar-Based Recognition of Handwritten Sentences, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28, 818-821.

[3] Bertolami, R. & Bunke, H. Hidden Markov model-based ensemble methods for offline handwritten text line recognition, Pat. Recog., 2008, 41, 3452- 3460.

[4] Manmatha, R. & Rothfeder, J. L. A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents, IEEE Trans. Pattern Anal. Mach. Intell., 2005, 27, 1212-1225.

[5] Louloudis, G.; Gatos, B.; Pratikakis, I. & Halatsis, C. Text line and word segmentation of handwritten documents, Pattern Recog., 2009, 42, 3169- 3183.

[6] Papavassiliou, V.; Stafylakis, T.; Katsouros, V. & Carayannis, G., Handwritten document image segmentation into text lines and words, Pattern Recognition, 2010, 43, 369-377.

[7] Haji, M., Arbitrary Word Spotting in Handwritten Documents, Doctoral Thesis, Concordia University, April, 2012.

[8] Marti, U.-V. & Bunke, H., The IAM-database: an English sentence database for offline handwriting recognition, International Journal on Document Analysis and Recognition, 2002, 5(1), 39-46.

[9] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE, 1989, 77, 257-286.

[10] Christopher D. Manning, P. R. & Schütze, H., Introduction to Information Retrieval, Cambridge University Press, 2008.

[11] Rodriguez-Serrano, J. A. & Perronnin, F., Handwritten word-spotting using hidden Markov models and universal vocabularies, Pattern Recognition, 2009, 42, 2106-2116.