Sentence Boundary Detection for Handwritten Text Recognition

Matthias Zimmermann
International Computer Science Institute
Berkeley, CA 94704, USA
zimmerma@icsi.berkeley.edu

Abstract

In the larger context of handwritten text recognition systems many natural language processing techniques can potentially be applied to the output of such systems. However, these techniques often assume that the input is segmented into meaningful units, such as sentences. This paper investigates the use of hidden-event language models and a maximum entropy based method for sentence boundary detection. While hidden-event language models are simple to train, the maximum entropy framework allows for an easy integration of various knowledge sources. The segmentation performance of these two approaches are evaluated on the IAM Database for handwritten English text and results on true words as well as recognized words are provided. Finally, a combination of the two techniques is shown to achieve superior performance over both individual methods.

1 Introduction

Unconstrained handwritten text recognition has reached word recognition rates between 50% and 80% for lexicons of 10,000 or more words [23, 26]. Therefore, handwriting recognition starts to become attractive for applications beyond mail sorting [5] or check reading [9]. The retrieval of handwritten documents has already been shown to be feasible at the document level even for relatively low word recognition rates [23]. However, natural language processing techniques typically require a segmentation of the (recognized) text into sentences. Examples are tagging, parsing, summarization or machine translation. The goal of this paper is to overcome the assumption of segmented input text that we made in our previous work on the integration of parsing [27] and more generally, to close the gap between the output of todays handwritten text recognizers and the required input for the above mentioned natural language processing techniques.

Sentence boundary detection consists in inserting sentence boundary tokens <s> into a stream of words. This task is not trivial even when the stream of words does not contain any recognition errors. Although the end of a sentences can almost always be found at a sentence final word (., ..., *!*,*?*,*:*, or ") ambiguities result around abbreviations, quotation marks, etc. (see Fig. 1 for an example). In the presence of recognition errors sentence boundary detection becomes significantly harder. We no longer can rely on sentence final words, as the recognition process can easily miss such words or hypothesize them in wrong locations. In the context of speech recognition systems the situation is even worse as sentence final words are completely missing. To segment the output of automatic speech recognition systems hidden-event language models are often used and it is that technique that is first investigated in this paper. Then, a different approach based on maximum entropy is considered that has been reported to perform well on printed text. Finally, the maximum entropy based approach is integrated into the hidden-event language model framework.

1) The summonses say they are " likely to persevere in such unlawful conduct . " <s> They ...
2) " It comes at a bad time , " said Ormston . <s> " A singularly bad time ...

Figure 1. Typical ambiguity for the position of a sentence boundary token <s> in the context of a period followed by quotes.

The rest of the paper is organized as follows. The following section presents related work. Section 3 then introduces the methodology of hidden-event language models and the features used for the maximum entropy approach. Experiments and results are presented in Section 4 and conclusions are drawn in Section 5.

2 Related Work

To the knowledge of the author no prior work in the domain of handwritten text recognition exists. However, the problem of sentence boundary detection has been addressed in various settings in the domain of speech and language processing. For the segmentation of raw text decision trees were used by [20, 18] where [18] also investigated neural networks. More recently, a maximum entropy approach for the segmentation of printed text was presented in [19] with the advantage that it achieves a comparable performance to other state-of-the-art systems, does not depend on syntactical part of speech (POS) tags, and requires significantly less training data.

In the case of the segmentation of the output of speech recognition systems the use of language models that also model hidden events (e.g. sentence boundary tokens <s>) has been proposed in [22]. Later, these word based

hidden-event language models have been extended by integrating prosodic cues (duration, pitch, and energy) modeled by decision trees [21]. As shown in [28], the performance can further be improved by replacing the decision trees with maximum entropy models that more tightly integrate words and prosodic features.

3 Methodology

This section presents the techniques that are investigated for the sentence boundary detection. The task is to find the best sequence of boundary events $T = (T_1, \dots T_n)$ for a given word input stream $W = (W_1, \dots W_n)$ where $T_i \in \{<\text{s}>, \emptyset\}$.

The first two subsections cover the hidden-event language model technique, the maximum entropy modeling, and a motivation for the features used in the proposed approach. Then, the integration of maximum entropy models into the hidden-event language modeling technique is explained. Finally, the handwritten text recognition system is described.

3.1 Hidden-Event Language Models

Hidden-event language models for text segmentation were introduced in [22]. They can be considered a variant of the widely used statistical n-gram language models [10]. The difference arises from the fact that during the training of the hidden-event language models the events to detect (sentence boundary tokens <s> in our case) are explicitly present, while they are missing (or hidden) during the recognition phase. For the segmentation of an input word sequence with missing sentence boundaries, this language model is then used in a hidden Markov modeling (HMM) framework. The states represent the presence or absence of a sentence boundary event for each word and the transition probabilities are given by the n-gram probabilities. For the generation of the final sequence of boundary events and non-boundary events the forward-backward algorithm [3] is used to compute the most likely overall sequence of boundary events T^*.

$$T^* = \underset{T}{argmax}\, p(T|W) \tag{1}$$

For the experiments in this paper 4-gram language models with interpolated Kneser-Ney smoothing [8, 14] are first trained from different text sources. The various language models are then linearly interpolated where the interpolation weights are computed using expectation maximization according to [11].

3.2 Maximum Entropy

Maximum entropy models have been successfully used in a wide variety of applications as they can easily handle thousands of features and the model training procedure is proved to be able to converge to the uniquely defined global optimum. See [4] for an excellent introduction. The model that is trained in the maximum entropy framework is of the following exponential form:

… 12	tomorrow	.	Mr.	Michael …
w_{i-2}	w_{i-1}	w_i	w_{i+1}	w_{i+2}

Feature Set	Features at Position i
Word	$w_{i-2}(12)$, $w_{i-1}(tomorrow)$, $w_i(.)$, $w_{i+1}(Mr.)$, $w_{i+2}(Michael)$
Bigram	$b_{i-1}(12\ \ tomorrow)$, $b_i(tomorrow\ \ .)$, $b_{i+1}(.\ \ Mr.)$, $b_{i+2}(Mr.\ \ Michael)$
Capital	$c_5(0a.AA)$, $c_3(a.A)$, $c_l(0a)$ $c_r(AA)$
Line Break	$l(none)$

Figure 2. The feature sets used for maximum entropy modeling. The example corresponds to a sentence boundary after word w_i. Capital refers to the features derived from capitalization of the words.

$$p_\lambda(c|x) = \frac{1}{Z_\lambda(x)} \exp \sum_i \lambda_i f_i(x, c) \tag{2}$$

where $p_\lambda(c|x)$ represents the posterior probability of class c ($c \in \{<\text{s}>, \emptyset\}$ in our case) given the context x. The $f_i(x, c) \in \{1, 0\}$ correspond to the binary features that are derived from both the context x and the class label c. The feature weights λ_i are estimated on the training data. These weights represent the only free parameters of a maximum entropy model for a given set of features. Finally, $Z_\lambda(x)$ normalizes the exponential part of the above equation as follows $\sum_c p_\lambda(c|x) = 1$. In its standard form, all features in a maximum entropy model are of a binary form indicating either the presence or absence of a feature.

For the maximum entropy model of this paper four different feature sets are used that are extracted from the context of five consecutive words surrounding each potential sentence boundary location (i.e. after every word of the word input stream). See Fig. 2 for an example. Please note, that the features shown in Fig. 2 follow the usual convention where only those features are shown that have a value of 1. The simplest feature set directly uses the individual words in the window and the bigram feature set consists of the four word bigrams that can be found in the same context. The third feature set ('Capital' in Fig. 2) maps all five words of a context into a single word. Each word is represented as a single character 'A' or 'a' depending on the capitalization of the first character of each word. Numbers are mapped to '0' and other words (such as sentence final words ., ..., !,?,:, or ") are preserved as they are. The previous feature set is motivated from the observation that the correct capitalization is often preserved even in the case of misrecognized words. These features can be particularly valuable when a sentence final word has been deleted in the recognition process but they also serve as backup features in the case of unknown words or words that have been observed only very few times during the training of the maximum entropy model. Finally, the layout of the written text should also be represented in a set of appropriate features. The presence and positions of

titles, paragraphs, lists etc. does not seem to be very hard to detect but can provide very strong cues for the ending of sentences. As a weak indication of the end of a paragraph only the presence or absence of a line break after word w_i is used as a feature in this paper.

The experimental setup used in this paper differs in a number of ways of the methods described in [19]. Most importantly, we have to segment recognized words (instead of knowing the true words). As a result, sentence boundaries can appear after each word and not only at sentence final words. For the features we also include word bigrams to take advantage of frequent sentence endings or sentence starts. We also attempt to exploit the layout of the document by taking into account the position of line breaks.

3.3 Model Combination

For the combination of the hidden-event language model with the maximum entropy based sentence boundary detection system the integrated HMM scheme described in [21] is used. The original task of finding the optimal sequence T^* for a given word sequence W is extended to take into account additional information $X = (X_1, \ldots X_n)$ related to the input word sequence.

$$T^* = \underset{T}{argmax}\, p(T|W, X) \tag{3}$$

In contrast to a HMM based hidden-event language model the states of the integrated model do not only emit words, but information gained from additional knowledge sources in the form of likelihoods $p(X_i|T_i, W)$ as well. In [21] the required likelihoods are obtained from the outputs of decision trees computed from the prosodic features extracted around word boundaries. In our case the required likelihoods are derived from the posterior probabilities estimated by the maximum entropy model. This concept has also been successfully used for the joint segmentation and classification of dialog acts [28].

3.4 Handwritten Text Recognizer

The recognizer for unconstrained handwritten English texts used in the experiments reported below is based on hidden Markov models (HMM). It is derived from a system described in [16]. The main recognition step consists in Viterbi decoding [24] supported by a word bigram language model. For language model smoothing we use the Good-Turing technique[7] together with Katz-backoff to lower order models[13].

Substantial performance improvements over [16] were achieved through the following extensions: We use mixtures of eight Gaussians instead of a single Gaussian and optimize the number of model states individually per character [25] instead of using a single global number of states per character model. In contrast to other works in the domain of handwritten text recognition, the integration of the word bigram language model is optimized as described in [26].

Table 1. Cross validation set definition. All meta parameters will be optimized on one set and applied on the other set and vice versa leading to average performance measures over the 400 sentences.

Name	Sentences	Words	Recognition Accuracy
Set 1	200	3995	70.5%
Set 2	200	3936	76.5%
Total	400	7931	73.5%

4 Experiments and Results

The description of the experimental setup in the first subsection covers the handwritten material and the metrics involved in system optimization and evaluation. Section 4.2 explains various optimization steps of the applied techniques. The final Section 4.3 reports cross-validated results of the best configuration of the hidden-event language modeling technique, the maximum entropy approach, as well as the combined system.

4.1 Experimental Setup

For the text segmentation experiments the same 400 sentences extracted from the offline IAM Database [17] have been used as in previous work [26] . For this paper the 400 sentences are divided into two cross validation sets of 200 sentences each, written by two disjunct sets of persons. As the recognizer has been trained on handwritten text written by writers which are not present in the two validation sets the experimental setup is writer independent. By using both the transcriptions and the best recognizer output all experiments can be carried out for two conditions. First, the *true words condition* refers to a setup that assumes a recognition accuracy of 100% by using the transcriptions of the sentences. When the 1-best output of the handwritten text recognition system as described in [26] is used, the experiments can be carried out under the *recognized words condition*. See Table 1 for details of the cross validation sets and its corresponding word recognition accuracies. For all segmentation experiments the 200 sentences of each cross validation set (under both true words conditions and recognized words conditions) are concatenated to a single stream of words that is then fed into the sentence boundary detection systems.

For the optimization and evaluation of sentence boundary detection systems appropriate performance metrics have to be defined. As this paper concentrates on the sentence boundary detection alone, and does not consider a specific application for which the sentence segmentation could be optimized, two different metrics will be used to allow a more detailed analysis of the systems performance. The first metric *NIST-SU* [1] evaluates the segmentation error based on the reference sentence unit boundaries. It is defined as the number of missed and inserted sentence boundaries (i.e. false alarms, or FA) normalized by the number of reference boundaries. The second metric is the *F-Measure* that is widely used in the domain of information retrieval. It is defined as

Reference	w.w.w|w|w.w|w.w.w.w|
System	w.w|w.w.w.w|w.w.w.w|
Counts	...F.M.M...C.......C

Metric	Errors/Counts	Reference	Rate
NIST-SU	1 FA, 2 miss	4 boundaries	75%
Recall	2 correct	4 boundaries	50%
Precision	2 correct	3 boundaries	67%
F-Measure			57%

Figure 3. Performance metrics used for the evaluation of the sentence boundary detection systems. An F corresponds to a false alarm (FA), while missed sentence boundaries are indicated with an M. For correctly recognized boundaries the letter C is used.

Table 2. Validation set perplexities for hidden-event language models that have been trained on different corpora. The perplexity values are computed under both true words condition (True) and recognized words condition (Rec.)

Cond.	Rec.	IAM	LOB	WC	Brown	Int.
True		152	276	299	322	175
Rec.	108	209	363	433	406	230

$\frac{2 \times Recall \times Precision}{Recall + Precision}$ where *Recall* measures the percentage of the detected sentence boundaries and *Precision* is computed as the percentage of the true sentence boundaries among all sentence boundaries hypothesized by the segmentation system. See Fig. 3 for an illustration of the performance metrics defined above.

4.2 System Optimization

This section reports the system optimization steps that have been carried out for both the hidden-event language model based segmentation as well as the segmentation system relying on the maximum entropy approach. For the training of the hidden-event language models various text sources were used. In addition to the transcriptions of the IAM Database, the entire LOB corpus [12] is used after all sentences contained in the cross validation sets have been removed from both the transcriptions and the LOB corpus. Furthermore, language models were also trained on the Brown corpus [6] and the Wellington corpus (WC) [2]. The perplexity of the corresponding hidden-event language models was then measured on the first validation set. Table 2 reports the measured perplexity values under both true words condition (row 'True') and recognized words condition (row 'Rec.'). As expected, the perplexity values under the recognized words conditions are substantially higher than the perplexity values under the true words conditions [1]. The best hidden-event language models are obtained through linear interpolation of the individual 4-gram language models (column 'Int.' in Table 2). For the true words condition the average interpolation weights found by the optimization procedure (see Section 3.1) are 0.37 for the transcriptions of the IAM database, 0.33 for the LOB corpus, 0.15 for the Wellington corpus, and 0.15 for the Brown corpus. In the case of the recognized word conditions the lowest perplexity rates are obtained when the 1-best output of handwritten text recognizer is also used for language model training. The average interpolation weights are 0.08 for the recognized texts, the Wellington corpus, and the Brown corpus. Higher weights of 0.37 are assigned to both the transcriptions and to the LOB corpus.

Table 3. The Effect of adding feature sets for maximum entropy modeling using recognized words. The first validation set was used for training and and the second validation set for testing. W: Word features, B: Bigram features, C: Capitalization features, and L: Line break feature.

Features	NIST-SU	F-Measure
W	60.5%	71.3%
W + B	54.5%	72.3%
W + B + C	41.5%	74.8%
W + B + C + L	35.0%	80.1%

For the optimization of the maximum entropy based segmentation system the use of the different feature sets introduced in Section 3.2 is measured and the resulting NIST-SU error rates and F-Measure scores under recognized words condition are reported in Table 3. The simplest system relies on single word features only and achieves a NIST-SU segmentation error of 60.5% and a corresponding F-Measure of 71.3%. The results provided in Table 3 indicate how each feature set further improves the segmentation performance. It is interesting to see that a substantial improvement results from the addition of the line break feature even in the presence of many line breaks that do not correlate with the end of a sentence. This observation confirms the importance of features that represent the layout of a document for a sentence boundary detection system.

4.3 Evaluation

The evaluation of the individual sentence boundary detection systems is carried out under the both true words condition and recognized words condition. All performance scores reported in Table 4 are averaged over the two validation sets as follows. In a first experiment the first validation set is used to optimize parameters and the performance of this system is measured on the second validation set. The validation sets are then switched for the second experiment.

For the system based on hidden-event language models only the final language models interpolated from all

[1] The very low perplexity of the language model trained on the recognized words of the second validation set (column Rec. of Table 2) does not directly reflect the quality of that language model. It is the result of not taking into account the out of vocabulary words found in the first validation set.

Table 4. Final results for both true words and recognized words. The first table reports the performance under true words condition while the second table provides the corresponding scores under the recognized words condition. HE-LM, MaxEnt, and Comb. refer to hidden-event language model, maximum entropy, and combination.

True Words	NIST	F-Msr.	Recall	Prec.
HE-LM	4.5	97.8	97.3	98.3
MaxEnt	6.0	97.0	97.3	96.8
Comb.	2.5	98.8	99.0	98.5

Rec. Words	NIST	F-Msr.	Recall	Prec.
HE-LM	43.0	77.6	74.5	81.0
MaxEnt	41.3	76.9	69.3	86.9
Comb.	40.5	79.8	80.0	79.6

available text sources are used. The maximum entropy system incorporates all investigated feature sets. Finally, in the case of the sentence boundary detection system integrating the hidden-event language model and the maximum entropy based technique equal weights are used for the two techniques. As no attempt was made to optimize these weights, the reported performance of the combined system can be interpreted as a conservative estimate. The measured performance under true words condition demonstrates the effectiveness of the techniques investigated in this paper and confirms the very high accuracy rates reported in the literature for this task.

The comparison of the performance of the hidden-event language model with the maximum entropy based approach under recognized word condition suggests that the maximum entropy based approach might be more robust in handling recognition errors than the hidden-event language model. This impression is further supported by the fact that (due to resource constraints) the maximum entropy based method is only trained on recognizer output, the transcriptions of the IAM database and the LOB corpus (in contrast to the hidden-event language model, that is also trained on the Wellington corpus and the Brown corpus). The achieved performance of the combined system under recognized words condition is very encouraging. Even in the presence of a significant amount of recognition errors it is possible to detect 80% of the sentence boundaries without introducing an excessive amount of false alarms.

5 Conclusions and Outlook

This paper addresses the problem of the detection of sentence boundaries in the output of a state-of-the-art handwritten text recognition system. Two sentence boundary detection techniques depending on hidden-event language models and maximum entropy are investigated. First, a separate optimization of the two systems is performed leading to encouraging segmentation rates under recognized text condition. The integration of the output of the maximum entropy approach into the hidden-event language model framework is then shown to outperform both individual models.

Future work will involve larger validation sets and more text data to train the hidden-event language models. Instead of integrating the output of the maximum entropy approach into the hidden-event language model, an integration of the output of the hidden-event language model into the maximum entropy framework should be investigated as well. Finally, the use of conditional random fields as suggested in [15] seems to be promising.

6 Acknowledgment

This work was supported by the Swiss National Science Foundation through the research network IM2.

References

[1] NIST website, RT-03 fall rich transcription. http://www.nist.gov/speech/tests/rt/rt2003/fall/, 2003.

[2] L. Bauer. *Manual of Information to accompany The Wellington Corpus of Written New Zealand English, for use with Ditigal Computers*. Department of Linguistics, Victoria University, Wellington, New Zealand, 1993.

[3] L. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, 41(1):164–171, 1970.

[4] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[5] D. D'Amato, E. Kuebert, and A. Lawson. Results from a performance evaluation of handwritten address recogntion systems for the united states postal service. In *7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 189–198, Amsterdam, The Netherlands, 2000.

[6] W. N. Francis and H. Kucera. *Brown Corpus Manual, Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Ditigal Computers*. Department of Linguistics, Brown University, Providence RI, USA, 1979.

[7] I. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

[8] J. T. Goodman. A bit of progress in language modeling. Msr-tr-2001-72, Machine Learning and Applied Statistics Group, Microsoft, Redmond, USA, 2001.

[9] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximor. Industrial bank check processing: the A2iA check reader. *Int. Journal on Document Analysis and Recognition*, 3:196–206, 2001.

[10] F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann, 1990.

[11] F. Jelinek and R. Mercer. Interpolated estimation of markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice*, pages 381–402. North Holland, Amsterdam, 1980.

[12] S. Johansson, G. Leech, and H. Goodluck. *Manual of Information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers*. Department of English, University of Oslo, Oslo, 1978.

[13] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 35(3):400–401, 1987.

[14] R. Kneser and H. Ney. Improved backing-off for m-gram language models. In *Int. Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184, Massachusetts, USA, 1995.

[15] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper. Using conditional random fields for sentence boundary detection in speech. In *43rd Annual Meeting of the ACL*, pages 451–458, Ann Arbor, USA, 2005.

[16] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[17] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for off-line handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[18] D. D. Palmer and M. A. Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational Linguistics*, 23(2):241–267, 1997.

[19] J. C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *5th Conference on Applied Natural Language Processing*, pages 16–19, Washington, USA, 1997.

[20] M. D. Riley. Some applications of tree-based modelling to speech and language. In *DARPA Speech and Language Technology Workshop*, pages 339–352, Massachusetts, USA, 1989.

[21] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür. Prosody-based segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154, 2000.

[22] A. Stolcke and E. Shriberg. Automatic linguistic segmentation of conversational speech. In *Int. Conference on Spoken Language Processing*, volume 2, pages 1005–1008, Philadelphia, USA, 1996.

[23] A. Vinciarelli. Application of information retrieval techniques to single writer documents. *Pattern Recognition Letters*, 26(14-15):2262–2271, 2002.

[24] A. Viterbi. Error bounds for convolutional codes and an simptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[25] M. Zimmermann and H. Bunke. Hidden Markov model length optimization for handwriting recognition systems. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 369–374, Niagra-on-the-Lake, Canada, Aug. 2002.

[26] M. Zimmermann and H. Bunke. Optimizing the integration of statistical language models in HMM based offline handwritten text recognition. In *17th Int. Conf. on Pattern Recognition*, volume 2, pages 541 – 544, Cambridge, England, 2004.

[27] M. Zimmermann, J.-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):818–821, 2006.

[28] M. Zimmermann, A. Stolcke, and E. Shriberg. Joint segmentation and classification of dialog acts in multiparty meetings. In *Int. Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 581–584, Toulouse, France, 2006.