Proceedings of the Second SIGHAN Workshop on Chinese
Language Processing, July 2003, pp. 16-23.

Utterance Segmentation Using Combined Approach Based on Bi-directional N-gram and Maximum Entropy

Ding Liu
National Laboratory of Pattern Recognition
Institute of Automation
Chinese Academy of Sciences
Beijing 100080, China.
dliu@nlpr.ia.ac.cn

Chengqing Zong
National Laboratory of Pattern Recognition
Institute of Automation
Chinese Academy of Sciences
Beijing 100080, China.
cqzong@nlpr.ia.ac.cn

Abstract

This paper proposes a new approach to segmentation of utterances into sentences using a new linguistic model based upon Maximum-entropy-weighted Bi-directional N-grams. The usual N-gram algorithm searches for sentence boundaries in a text from left to right only. Thus a candidate sentence boundary in the text is evaluated mainly with respect to its left context, without fully considering its right context. Using this approach, utterances are often divided into incomplete sentences or fragments. In order to make use of both the right and left contexts of candidate sentence boundaries, we propose a new linguistic modeling approach based on Maximum-entropy-weighted Bi-directional N-grams. Experimental results indicate that the new approach significantly outperforms the usual N-gram algorithm for segmenting both Chinese and English utterances.

1 Introduction

Due to the improvement of speech recognition technology, spoken language user interfaces, spoken dialogue systems, and speech translation systems are no longer only laboratory dreams. Roughly speaking, such systems have the structure shown in Figure 1.

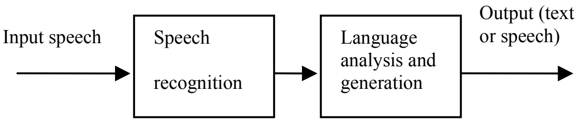

Figure 1. System with speech input.

In these systems, the language analysis module takes the output of speech recognition as its input, representing the current utterance exactly as pronounced, without any punctuation symbols marking the boundaries of sentences. Here is an example: *这边请您坐电梯到 9 楼服务生将在那里等您并将您带到913 号房间* . (*this way please please take this elevator to the ninth floor the floor attendant will meet you at your elevator entrance there and show you to room 913.*) As the example shows, it will be difficult for a text analysis module to parse the input if the utterance is not segmented. Further, the output utterance from the speech recognizer usually contains wrongly recognized words or noise words. Thus it is crucial to segment the utterance before further language processing. We believe that accurate segmentation can greatly improve the performance of language analysis modules.

Stevenson et al. have demonstrated the difficulties of text segmentation through an experiment in which six people, educated to at least the Bachelor's degree level, were required to segment into sentences broadcast transcripts from which all punctuation symbols had been removed. The experimental results show that humans do not always agree on the insertion of punctuation symbols, and that their segmentation performance is not very good (Stevenson and Gaizauskas, 2000). Thus it is a great challenge for computers to perform the task automatically. To solve this problem, many methods have been proposed, which can be roughly classified into two categories. One approach is based on simple acoustic criteria, such as non-speech intervals (e.g. pauses), pitch and energy. We can call this approach *acoustic segmentation*. The other approach, which can be called *linguistic segmentation*, is based on linguistic clues, including lexical knowledge, syntactic structure, semantic information etc. Acoustic segmentation can not always work well, because utterance boundaries do not always correspond to acoustic criteria. For example: *您好<pause>请问<pause>明天的单人间还有吗<pause>或者<pause>标准间也行.* Since the simple acoustic criteria are inadequate, linguistic clues play an indispensable role in utterance segmentation, and many methods relying on them have been proposed.

This paper proposes a new approach to linguistic segmentation using a Maximum-entropy-weighted Bi-directional N-gram-based algorithm (MEBN). To evaluate the performance of MEBN, we conducted experiments in both Chinese and English. All the results show that MEBN outperforms the normal N-gram algorithm. The remainder of this paper will focus on description of our new approach for linguistic segmentation. In Section 2, some related work on utterance segmentation is briefly reviewed, and our motivations are described. Section 3 describes MEBN in detail. The experimental results are presented in Section 4. Finally, Section 5 gives our conclusion.

2 Related Work and Our Motivations

2.1 Related Work

Stolcke et al. (1998, 1996) proposed an approach to detection of sentence boundaries and disfluency locations in speech transcribed by an automatic recognizer, based on a combination of prosodic cues modeled by decision trees and N-gram language models. Their N-gram language model is mainly based on part of speech, and retains some words which are particularly relevant to segmentation. Of course, most part-of-speech taggers require sentence boundaries to be pre-determined; so to require the use of part-of-speech information in utterance segmentation would risk circularity. Cettolo et al.'s (1998) approach to sentence boundary detection is somewhat similar to Stolcke et al.'s. They applied word-based N-gram language models to utterance segmentation, and then combined them with prosodic models. Compared with N-gram language models, their combined models achieved an improvement of 0.5% and 2.3% in precision and recall respectively.

Beeferman et al. (1998) used the CYBERPUNC system to add intra-sentence punctuation (especially commas) to the output of an automatic speech recognition (ASR) system. They claim that, since commas are the most frequently used punctuation symbols, their correct insertion is by far the most helpful addition for making texts legible. CYBERPUNC augmented a standard trigram speech recognition model with lexical information concerning commas, and achieved a precision of 75.6% and a recall of 65.6% when testing on 2,317 sentences from the Wall Street Journal.

Gotoh et al. (1998) applied a simple non-speech interval model to detect sentence boundaries in English broadcast speech transcripts. They compared their results with those of N-gram language models and found theirs far superior. However, broadcast speech transcripts are not really spoken language, but something more like spoken written language. Further, radio broadcasters speak formally, so that their reading pauses match sentence boundaries quite well. It is thus understandable that the simple non-speech interval model outperforms the N-gram language model under these conditions; but segmentation of natural utterances is quite different.

Zong et al. (2003) proposed an approach to utterance segmentation aiming at improving the performance of spoken language translation (SLT) systems. Their method is based on rules which are oriented toward key word detection, template matching, and syntactic analysis. Since this approach is intended to facilitate translation of Chinese-to-English SLT systems, it rewrites long sentences as several simple units. Once again, these results cannot be regarded as general-purpose utterance segmentation. Furuse et al. (1998) similarly propose an input-splitting method for translating spoken language which includes many long or ill-formed expressions. The method splits an input into well-balanced translation units, using a semantic dictionary.

Ramaswamy et al. (1998) applied a maximum entropy approach to the detection of command boundaries in a conversational natural language user interface. They considered as their features words and their distances to potential boundaries. They posited 400 feature functions, and trained their weights using 3000 commands. The system then achieved a precision of 98.2% in a test set of 1900 commands. However, command sentences for conversational natural language user interfaces contain much smaller vocabularies and simpler structures than the sentences of natural spoken language. In any case, this method has been very helpful to us in designing our own approach to utterance segmentation.

There are several additional approaches which are not designed for utterance segmentation but which can nevertheless provide useful ideas. For example, Reynar et al. (1997) proposed an approach to the disambiguation of punctuation marks. They considered only the first word to the left and right of any potential sentence boundary, and claimed that examining wider context was not beneficial. The features they considered included the candidate's prefix and suffix; the presence of particular characters in the prefix or suffix; whether the candidate was honorific (e.g. Mr., Dr.); and whether the candidate was a corporate designator (e.g. Corp.). The system was tested on the Brown Corpus, and achieved a precision of 98.8%. Elsewhere, Nakano et al. (1999) proposed a method for incrementally understanding user utterances whose semantic boundaries were unknown. The method operated by incrementally finding plausible sequences of utterances that play crucial roles in the task execution of dialogues, and by utilizing beam search to deal with the ambiguity of boundaries and with syntactic and semantic ambiguities. Though the method does not require utterance segmentation before discourse processing, it employs special rule tables for discontinuation of significant utterance boundaries. Such rule tables are not easy to maintain, and experimental results have demonstrated only that the method outperformed the method assuming pauses to be semantic boundaries.

2.2 Our motivations

Though numerous methods for utterance segmentation have been proposed, many problems remain unsolved.

One remaining problem relates to the language model. The N-gram model evaluates candidate sentence boundaries mainly according to their left context, and has achieved reasonably good results, but it can't take into account the distant right context to the candidate. This is the reason that N-gram methods often wrongly divide some long sentences into halves or multiple segments. For example: *小王病了一个星期.* The N-gram method is likely to insert a boundary mark between "了" and "一", which corresponds to our everyday impression that, if reading from the left and not considering several more words to the right of the current word, we will probably consider "小王病了" as a whole sentence. However, we find that, if we search the sentence boundaries from right to left, such errors can be effectively avoided. In the present example, we won't consider "一个星期" as a whole sentence, and the search will be continued until the word "小" is encountered. Accordingly, in order to avoid segmentation errors made by the normal N-gram method, we propose a reverse N-gram segmentation method (RN) which does seek sentence boundaries from right to left. Further, we simply integrate the two N-gram methods and propose a bi-directional N-gram method (BN), which takes into account both the left and the right context of a candidate segmentation site. Since the relative usefulness or significance of the two N-gram methods varies depending on the context, we propose a method of weighting them appropriately, using parameters generated by a maximum entropy method which takes as its features information about words in the context. This is our Maximum-Entropy-Weighted Bi-directional N-gram-based segmentation method. We hope MEBN can retain the correct segments discovered by the usual N-gram algorithm, yet effectively skip the wrong segments.

3 Maximum-Entropy-Weighted Bi-directional N-gram-based Segmentation Method

3.1 Normal N-gram Algorithm (NN) for Utterance Segmentation

Assuming that $W_1W_2...W_m$ (where *m* is a natural number) is a word sequence, we consider it as an *n* order Markov chain, in which the word $W_i (1 \le i \le m)$ is predicted by the *n-1* words to its left. Here is the corresponding formula:

$$P(W_i \mid W_1W_2...W_{i-1}) = P(W_i \mid W_{i-n+1}...W_{i-1})$$

From this conditional probability formula for a word, we can derive the probability of a word sequence $W_1W_2...W_i$:

$$P(W_1W_2..W_i) = P(W_1W_2..W_{i-1}) \times P(W_i \mid W_1W_2..W_{i-1})$$

Integrating the two formulas above, we get:

$$P(W_1W_2..W_i) = P(W_1W_2..W_{i-1}) \times P(W_i \mid W_{i-n+1}..W_{i-1})$$

Let us use SB to indicate a sentence boundary and add it to the word sequence. The value of $P(W_1W_2...W_iSBW_{i+1})$ and $P(W_1W_2...W_iW_{i+1})$ will determine whether a specific word $W_i (1 \le i \le m)$ is the final word of a sentence. We say W_i is the final word of a sentence if and only if $P(W_1W_2...W_iSBW_{i+1}) > P(W_1W_2...W_iW_{i+1})$.

Taking the trigram as our example and considering the two cases where W_{i-1} is and is not the final word of a sentence, $P(W_1W_2...W_iSBW_{i+1})$ and $P(W_1W_2...W_iW_{i+1})$ is computed respectively by the following two formulas:

$$\begin{aligned} P(W_1W_2...W_iSBW_{i+1}) &= P(W_1W_2...SBW_i) \times P(SB \mid SBW_i) \times P(W_{i+1} \mid W_iSB) \\ &+ P(W_1W_2...W_{i-1}W_i) \times P(SB \mid W_{i-1}W_i) \times P(W_{i+1} \mid W_iSB) \\ P(W_1W_2...W_iW_{i+1}) &= P(W_1W_2...SBW_i) \times P(W_{i+1} \mid SBW_i) \\ &+ P(W_1W_2...W_{i-1}W_i) \times P(W_{i+1} \mid W_{i-1}W_i) \end{aligned}$$

In the normal N-gram method, the above iterative formulas are computed to search the sentence boundaries from W_1 to W_m.

3.2 Reverse N-gram Algorithm (RN) for Utterance Segmentation

In the *reverse* N-gram segmentation method, we take the word sequence $W_1W_2...W_m$ as a reverse Markov chain in which $W_i (1 \le i \le m)$ is predicted by the *n-1* words to its right. That is:

$$P(W_i \mid W_mW_{m-1}...W_{i+1}) = P(W_i \mid W_{i+n-1}...W_{i+1})$$

As in the N-gram algorithm, we compute the occurring probability of word sequence $W_1W_2...W_m$ using the formula:

$$P(W_mW_{m-1}..W_i) = P(W_mW_{m-1}..W_{i+1}) \times P(W_i \mid W_mW_{m-1}..W_{i+1})$$

Then the iterative computation formula is:

$$P(W_mW_{m-1}..W_i) = P(W_mW_{m-1}..W_{i+1}) \times P(W_i \mid W_{i+n-1}..W_{i+1})$$

By adding SB to the word sequence, we say W_i is the final word of a sentence if and only if $P(W_mW_{m-1}...W_{i+1}SBW_i) > P(W_mW_{m-1}...W_{i+1}W_i)$. Similar to NN, $P(W_mW_{m-1}...W_{i+1}SBW_i)$ and $P(W_mW_{m-1}...W_{i+1}W_i)$ are computed as follows in the trigram:

$$\begin{aligned} P(W_mW_{m-1}..W_{i+1}SBW_i) &= P(W_mW_{m-1}..SBW_{i+1}) \times P(SB \mid SBW_{i+1}) \times P(W_i \mid W_{i+1}SB) \\ &+ P(W_mW_{m-1}..W_{i+2}W_{i+1}) \times P(SB \mid W_{i+2}W_{i+1}) \times P(W_i \mid W_{i+1}SB) \\ P(W_mW_{m-1}..W_{i+1}W_i) &= P(W_mW_{m-1}..SBW_{i+1}) \times P(W_i \mid SBW_{i+1}) \\ &+ P(W_mW_{m-1}..W_{i+2}W_{i+1}) \times P(W_i \mid W_{i+2}W_{i+1}) \end{aligned}$$

In contrast to the normal N-gram segmentation method, we compute the above iterative formulas to seek sentence boundaries from W_m to W_1.

3.3 Bi-directional N-gram Algorithm for Utterance Segmentation

From the iterative formulas of the normal N-gram algorithm and the reverse N-gram algorithm, we can see that the normal N-gram method recognizes a candidate sentence boundary location mainly according to its left context, while the reverse N-gram method mainly depends on its right context. Theoretically at least, it is reasonable to suppose that, if we synthetically consider both the left and the right context by integrating the NN and the RN, the overall segmentation accuracy will be improved.

Considering the word sequence $W_1W_2...W_m$, the candidate sites for sentence boundaries may be found between W_1 and W_2, between W_2 and W_3, …, or between W_{m-1} and W_m. The number of candidate sites is thus *m-1*. We number those *m-1* candidate sites 1, 2 … *m-1* in succession, and we use $P_{is}(i)$ $(1 \le i \le m-1)$ and $P_{no}(i)$ $(1 \le i \le m-1)$ respectively to indicate the probability that the current site *i* really is, or is not, a sentence boundary. Thus, to compute the word sequence segmentation, we must compute $P_{is}(i)$ and $P_{no}(i)$ for each of the *m-1* candidate sites. In the bi-directional BN, we compute $P_{is}(i)$ and $P_{no}(i)$ by combining the NN results and RN results. The combination is described by the following formulas:

$$P_{is\_BN}(i) = P_{is\_NN}(i) \times P_{is\_RN}(i)$$

$$P_{no\_BN}(i) = P_{no\_NN}(i) \times P_{no\_RN}(i)$$

where $P_{is\_NN}(i)$, $P_{no\_NN}(i)$ denote the probabilities calculated by NN which correspond to $P(W_1W_2...W_iSBW_{i+1})$ and $P(W_1W_2...W_iW_{i+1})$ in section 3.1 respectively and $P_{is\_RN}(i)$, $P_{no\_RN}(i)$ denote the probabilities calculated by RN which correspond to $P(W_mW_{m-1}...W_{i+1}SBW_i)$ and $P(W_mW_{m-1}...W_{i+1}W_i)$ in section 3.2 respectively.

We say there exits a sentence boundary at site i $(1 \le i \le m-1)$ if and only if $P_{is\_BN}(i) > P_{no\_BN}(i)$.

3.4 Maximum Entropy Approach for Utterance Segmentation

In this section, we explain our maximum-entropy-based model for utterance segmentation. That is, we estimate the joint probability distribution of the candidate sites and their surrounding words. Since we consider information concerning the lexical context to be useful, we define the feature functions for our maximum method as follows:

$$f_{j10}(b,c) = \begin{cases} 1 & if\ (include\ (\Pr efix\ (c), S_j)\ \&\ \&\ b == 0) \\ 0 & else \end{cases}$$

$$f_{j11}(b,c) = \begin{cases} 1 & if\ (include\ (\Pr efix\ (c), S_j)\ \&\ \&\ b == 1) \\ 0 & else \end{cases}$$

$$f_{j20}(b,c) = \begin{cases} 1 & if\ (include\ (Suffix\ (c), S_j)\ \&\ \&\ b == 0) \\ 0 & else \end{cases}$$

$$f_{j21}(b,c) = \begin{cases} 1 & if\ (include\ (Suffix\ (c), S_j)\ \&\ \&\ b == 1) \\ 0 & else \end{cases}$$

S_j denotes a sequence of one or more words which we can call the Matching String. (Note that S_j may contain the sentence boundary mark 'SB'.) The candidate c's state is denoted by b, where $b=1$ indicates that c is a sentence boundary and $b=0$ indicates that it is not a boundary. *Prefix(c)* denotes all the word sequences ending with c (that is, c's left context plus c) and *Suffix(c)* denotes all the word sequences beginning with c (in other words, c plus its right context). For example: in the utterance: *去<c1>机<c2>场<c3>怎<c4>么<c5>走,* '场', '机场', and '去机场' are *c3*'s Prefix, while '怎', '怎么'and '怎么走' are *c3*'s Suffix. The value of function $include(\Pr efix(c), S_j)$ is *true* when word sequence S_j is one of c's Prefixes, and the value of function $include(Suffix(c), S_j)$ is *true* when S_j is one of c's Suffixes.

Corresponding to the four feature functions $f_{j10}(b,c)$, $f_{j11}(b,c)$, $f_{j20}(b,c)$, $f_{j21}(b,c)$ are the four parameters α_{j10}, α_{j11}, α_{j20}, α_{j21}. Thus the joint probability distribution of the candidate sites and their surrounding contexts is given by:

$$P(c,b) = \pi \prod_{j=1}^{k} (\alpha_{j10}^{f_{j10}(b,c)} \times \alpha_{j11}^{f_{j11}(b,c)} \times \alpha_{j20}^{f_{j20}(b,c)} \times \alpha_{j21}^{f_{j21}(b,c)})$$

where k is the total number of the Matching Strings and π is a parameter set to make *P(c,1)* and *P(c,0)* sum to 1. The unknown parameters $\alpha_{j10}, \alpha_{j11}, \alpha_{j20}, \alpha_{j21}$ are chosen to maximize the likelihood of the training data using the *Generalized Iterative Scaling* (Darroch and Ratcliff, 1972) algorithm. In the maximum entropy approach, we say that a candidate site is a sentence boundary if and only if P(c, 1) > P(c, 0). (At this point, we can anticipate a technical problem with the maximum approach to utterance segmentation. When a Matching String contains SB, we cannot know whether it belongs to the Prefixes or Suffixes of the candidate site until the left and right contexts of the candidate site have been segmented. Thus if the segmentation proceeds from left to right, the lexical information in the right context of the current candidate site will always remain uncertain. Likewise, if it proceeds from right to left, the information in the left context of the current candidate site remains uncertain. The next subsection will describe a pragmatic solution to this problem.)

3.5 Maximum-Entropy-Weighted Bi-directional N-gram Algorithm for Utterance Segmentation

In the bi-directional N-gram based algorithm, we have considered the left-to-right N-gram algorithm and the right-to-left algorithm as having the same significance. Actually, however, they should be assigned differing weights, depending on the lexical contexts. The combination formulas are as follows:

$$P_{is}(i) = W_{n\_is}(C_i) \times P_{is\_NN}(i) \times W_{r\_is}(C_i) \times P_{is\_RN}(i)$$

$$P_{no}(i) = W_{n\_no}(C_i) \times P_{no\_NN}(i) \times W_{r\_no}(C_i) \times P_{no\_RN}(i)$$

$W_{n\_is}(C_i)$, $W_{n\_no}(C_i)$, $W_{r\_is}(C_i)$, $W_{r\_no}(C_i)$ are the functions of the context surrounding candidate site i which denotes the weights of $P_{is\_NN}(i)$, $P_{no\_NN}(i)$, $P_{is\_RN}(i)$ and $P_{no\_RN}(i)$ respectively. Assuming that the weights of $P_{is\_NN}(i)$ and $P_{no\_NN}(i)$ depend upon the context to the left of the candidate site, and that the weights of

$P_{is\_RN}(i)$ and $P_{no\_RN}(i)$ depend on the context to the right of the candidate site, the weight functions can be rewritten as: $W_{n\_is}(LeftC_i)$, $W_{n\_no}(LeftC_i)$, $W_{r\_is}(RightC_i)$, $W_{r\_no}(RightC_i)$. It is reasonable to assume that as the joint probability $P(LeftC_i, i = SB)$ rises, $P_{is\_NN}(i)$ will increase in significance. (The joint probability in question is the probability of the current candidate's left context, taken together with the probability that the candidate is a sentence boundary.) Therefore the value of $W_{n\_is}(LeftC_i)$ is given by $W_{n\_is}(LeftC_i) = P(LeftC_i, i = SB)$. Similarly we can give the formulas for computing $W_{n\_no}(LeftC_i)$, $W_{r\_is}(RightC_i)$, and $W_{r\_no}(RightC_i)$ as follows:

$$W_{n\_no}(LeftC_i) = P(LeftC_i, i != SB)$$

$$W_{r\_is}(RightC_i) = P(RightC_i, i = SB)$$

$$W_{r\_no}(RightC_i) = P(RightC_i, i != SB)$$

We can easily get the values of $P(LeftC_i, i = SB)$, $P(LeftC_i, i != SB)$, $P(RightC_i, i = SB)$, and $P(RightC_i, i != SB)$ using the method described in the maximum entropy approach section. For example:

$$P(LeftC_i, i = SB) = \pi \prod_{j=1}^{k} \alpha_{j11}^{f_{j11}(1,i)}$$

$$P(LeftC_i, i != SB) = \pi \prod_{j=1}^{k} \alpha_{j10}^{f_{j10}(0,i)}$$

As mentioned in last subsection, we need segmented contexts for maximum entropy approach. Since the maximum entropy parameters for MEBN algorithm are used as modifying NN and RN, we just estimate the joint probability of the candidate and its surrounding contexts based upon the segments by NN and RN. Using NLeftC$_i$ indicate the left context to the candidate *i* which has been segmented by NN algorithm and RRightC$_i$ indicate the right context to *i* which has been segmented by RN, the combination probability computing formulas for MEBN are as follows:

$$P_{is\_MEBN}(i) = P(NLeftC_i, i = SB) \times P_{is\_NN}(i) \times P(RRightC_i, i = SB) \times P_{is\_RN}(i)$$

$$P_{no\_MEBN}(i) = P(NLeftC_i, i != SB) \times P_{no\_NN}(i) \times P(RRightC_i, i != SB) \times P_{no\_RN}(i)$$

We evaluate site *i* as a sentence boundary if and only if $P_{is\_MEBN}(i) > P_{no\_MEBN}(i)$.

4 Experiment

4.1 Model Training

Our models are trained on both Chinese and English corpora, which cover the domains of hotel reservation, flight booking, traffic information, sightseeing, daily life and so on. We replaced the full stops with "SB" and removed all other punctuation marks in the training corpora. Since in most actual systems part of speech information cannot be accessed before determining the sentence boundaries, we use Chinese characters and English words without POS tags as the units of our N-gram models. Trigram and reverse trigram probabilities are estimated based on the processed training corpus by using *Modified Kneser-Ney Smoothing* (Chen and Goodman, 1998). As to the maximum entropy model, the Matching Strings are chosen as all the word sequences occurring in the training corpus whose length is no more than 3 words. The unknown parameters corresponding to the feature functions are generated based on the training corpus using the *Generalized Iterative Scaling algorithm*. Table 1 gives an overview of the training corpus.

Corpus	SIZE	SB Number	Average Length of Sentence
Chinese	4.02MB	148967	8 Chinese characters
English	4.49MB	149311	6 words

Table 1. Overview of the Training Corpus.

4.2 Testing Results

We test our methods using open corpora which are also limited to the domains mentioned above. All punctuation marks are removed from the test corpora. An overview of the test corpus appears in table 2.

Corpus	SIZE	SB Number	Average Length of Sentence
Chinese	412KB	12032	10 Chinese characters
English	391KB	10518	7 words

Table 2. Overview of the Testing Corpus.

We have implemented four segmentation algorithms using NN, RN, BN and MEBN respectively.

If we use “RightNum” to denote the number of right segmentations, “WrongNum” denote the number of wrong segmentations, and “TotalNum” to denote the number of segmentations in the original testing corpus, the precision (P) can be computed using the formula *P=RightNum/(RightNum+WrongNum)*, the recall (R) is computed as *R=RightNum/TotalNum*, and the F-Score is computed as *F-Score* $= \frac{2 \times P \times R}{P + R}$. The testing results are described in Table 3 and Table 4.

Methods	Total Num	Right Num	Wrong Num	Precision	Recall	F-Score
NN	12032	10167	2638	79.4%	84.5%	81.9%
RN	12032	10396	2615	79.9%	86.4%	83.0%
BN	12032	10528	2249	82.4%	87.5%	84.9%
MEBN	12032	10348	1587	86.7%	86.0%	86.3%

Table 3. Experimental Results for Chinese Utterance Segmentation.

Methods	Total Num	Right Num	Wrong Num	Precision	Recall	F-Score
NN	10518	8730	3164	73.4%	83.0%	77.9%
RN	10518	9014	3351	72.9%	85.7%	78.8%
BN	10518	9056	3019	75.0%	86.1%	80.2%
MEBN	10518	8929	2403	78.8%	84.9%	81.7%

Table 4. Experimental Results for English Utterance Segmentation.

From the result tables it is clear that RN, BN, and MEBN all outperforms the normal N-gram algorithm in the F-score for both Chinese and English utterance segmentation. MEBN achieved the best performance which improves the precision by 7.3% and the recall by 1.5% in the Chinese experiment, and improves the precision by 5.4% and the recall by 1.9% in the English experiment.

4.3 Result analysis

MEBN was proposed in order to maintain the correct segments of the normal N-gram algorithm while skipping the wrong segments. In order to see whether our original intention has been realized, we compared the segments as determined by RN with those determined by NN, compare the segments found by BN with those of NN and then compare the segments found by MEBN with those of NN. For RN, BN and MEBN, suppose TN denotes the number of total segmentations, CON denotes the number of correct segmentations overlapping with those found by NN; SWN denotes the number of wrong NN segmentations which were skipped; WNON denotes the number of wrong segmentations not overlapping with those of NN; and CNON denotes the number of segmentations which were correct but did not overlap with those of NN. The statistical results are listed in Table 5 and Table 6.

Methods	TN	CON	SWN	WNON	CNON
RN	13011	9525	1098	1077	870
BN	12777	9906	753	355	622
MEBN	11935	9646	1274	223	678

Table 5. Chinese Utterance Segmentation Results Comparison.

Methods	TN	CON	SWN	WNON	CNON
RN	12365	8223	1077	1271	792
BN	12075	8565	640	488	491
MEBN	11332	8370	1247	486	559

Table 6. English Utterance Segmentation Results Comparison.

Focusing upon the Chinese results, we can see that RN skips 1098 incorrect segments found by NN, and has 9525 correct segments in common with those of NN. It verifies our supposition that RN can effectively avoid some errors made by NN. But because at the same time RN brings in 1077 new errors, RN doesn’t improve much in precision. BN skips 753 incorrect segments and brings in 355 new segmentation errors; has 9906 correct segments in common with those of NN and brings in 622 new correct segments. So by equally integrating NN and RN, BN on one hand finds more correct segments, on the other hand brings in less wrong segments than NN. But in skipping incorrect segments by NN, BN still performs worse than RN, showing that it only exerts the error skipping ability of RN to some extent. As for MEBN, it skips 1274 incorrect segments and at the same time brings in only 223 new incorrect segments. Additionally it maintains 9646 correct segments in common with those of NN and brings in 678 new correct segments. In recall MEBN performs a little worse than BN, but in precision it achieves a much better performance than BN, showing that modified by the maximum entropy weights, MEBN makes use of the error skipping ability of RN more effectively. Further, in skipping wrong segments by NN, MEBN even outperforms RN, which indicates the weights we set on NN and RN not only act as modifying parameters, but also have direct beneficial affection on utterance segmentation.

5 Conclusion

This paper proposes a reverse N-gram algorithm, a bi-directional N-gram algorithm and a Maximum-entropy-weighted Bi-directional N-gram algorithm for utterance segmentation. The experimental results for both Chinese and English utterance segmentation show that MEBN significantly outperforms the usual N-gram algorithm. This is because MEBN takes into account both the left and right contexts of candidate sites: it integrates the left-to-right N-gram algorithm and the right-to-left N-gram algorithm with appropriate weights, using clues on the sites' lexical context, as modeled by maximum entropy.

Acknowledgements

This work is sponsored by the Natural Sciences Foundation of China under grant No.60175012, as well as supported by the National Key Fundamental Research Program (the 973 Program) of China under the grant G1998030504.

The authors are very grateful to Dr. Mark Seligman for his very useful suggestions and his very careful proofreading.

References

Beeferman D., A. Berger, and J. Lafferty. 1998. CYBERPUNC: A lightweight punctuation annotation system for speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA. pp. 689-692.

Beeferman D., A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning* 34, pp 177-210.

Berger A., S. Della Pietra, and V. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1), pp. 39-71.

Cettolo M. and D. Falavigna. 1998. Automatic Detection of Semantic Boundaries Based on Acoustic and Lexical Knowledge. *ICSLP* 1998, pp. 1551-1554.

Chen S. F. and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. *Technical Report* TR-10-98, Center for Research in Computing Technology, Harvard University. pp.243-255.

Darroch J. N. and D. Ratcliff. 1972. Generalized Iterative Scaling for Log-Linear Models. The *Annals of Mathematical Statistics*, 43(5), pp. 1470-1480.

Furuse O., S. Yamada, and K. Yamamoto. 1998. Splitting Long or Ill-formed Input for Robust Spoken-language Translation. *COLING-ACL* 1998, pp. 421-427.

Gotoh Y. and S. Renals. 2000. Sentence Boundary Detection in Broadcast Speech Transcripts. In *Proc. International Workshop on Automatic Speech Recognition*, pp. 228-235.

Nakano M., N. Miyazaki, J. Hirasawa, K. Dohsaka, and T. Kawabata. 1999. Understanding Unsegmented User Utterances in Real-Time Spoken Dialogue Systems. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, College Park, MD, USA, pp. 200-207.

Ramaswamy N. G. and J. Kleindienst. 1998. Automatic Identification of Command Boundaries in a Conversational Natural Language User Interface. *ICSLP* 1998. pp. 401-404.

Reynar J. and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the 5th Conference on Applications of Natural Language Processing (ANLP)*, Washington DC, pp. 16-19.

Seligman M. 2000. Nine Issues in Speech Translation. *In Machine Translation,* 15, pp. 149-185.

Stevenson M. and R. Gaizauskas. 2000. Experiments on sentence boundary detection. In *Proceedings of the Sixth Conference on Applied Natural Language Processing and the First Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 24-30.

Stolcke A. and E. Shriberg. 1996. Automatic linguistic segmentation of conversational speech. *Proc. Intl. Conf. on Spoken Language Processing,* Philadelphia, PA, vol. 2, pp. 1005-1008.

Stolcke A., E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, and Y. Lu. 1998. Automatic Detection of Sentence Boundaries and Disfluencies based on Recognized Words. *Proc. Intl. Conf. on Spoken Language Processing*, Sydney, Australia, vol. 5, pp. 2247-2250.

Zong, C. and F. Ren. 2003. Chinese Utterance Segmentation in Spoken Language translation. In *Proceedings of the 4th international conference on intelligent text processing and Computational Linguistics (CICLing)*, Mexico, Feb 16-22. pp. 516-525.

Zhou Y. 2001. Utterance Segmentation Based on Decision Tree. Proceedings of the *6th National joint Conference on Computational Linguistics*, Taiyuan, China, pp. 246-252.