Resource-Limited Sentence Boundary Detection

David Carter and Ian Gransden

Speech Machines, Merebrook Business Park,
Hanley Road, Malvern WR13 6NP, England
`david.carter@speechmachines.com, ian.gransden@speechmachines.com`

Abstract

We examine the practical constraints imposed on the task of sentence boundary detection in speech recognizer output, by the requirements of a system that supports large-scale commercial off-line transcription of dictations. We develop and evaluate a method that observes these constraints, reformulating the best technique previously reported in order to allow the use a smoothing technique directly tailored to boundary prediction. We then show how this method can be generalized and improved upon, demonstrating significantly better performance in three different domains.

1. Introduction

Production environments involving automatic speech recognition can impose resource limitations that present interesting challenges to the application of language modelling technology. Speech Machines' DictationNet (TM) system is an example of such an environment. DictationNet handles the recording, routing, transcribing and distribution of medical dictations. Recognition can be run on a recorded dictation to provide the (human) transcriptionist with a draft to edit, which, when accuracy is sufficiently high, can offer a significant time saving over transcribing from the audio alone.

Sentence boundary detection (SBD) is one of a number of operations that can be carried out to improve the quality of the draft presented to the transcriptionist. However, the environment imposes a number of constraints on how this can be done. These constraints may be regarded as in many ways typical of those that will be faced in any off-line dictation processing environment, or more generally, by any attempt to recognize large quantities of recorded speech from multiple sources without any human interaction during recognition.

The main constraints are:

- The speech recognizer used is likely to be a standard commercial product that must be treated as a "black box". Thus, SBD must operate on the recognizer's output, rather than being interleaved with the recognition process itself.
- The recognizer output may contain only limited prosodic information. For example, the recognizer we use offers only the durations of the recognized words, and the lengths of the pauses (if any) detected between them.
- The "reference" text available for training language models may be far from a verbatim transcription of what was actually said. Professional transcriptionists will often reorganize what is said to make it conform to the required format of a report. This may involve expanding some expressions, moving material around, and entering some of the spoken content in database fields associated with the transcription, so that it does not appear in the text at all. This means that automatic alignment of reference text with recognizer output can only ever be partially successful.
- Because the speech being recognized comes from many different domains (in our case, reports from many hospitals on many medical specialties), it will not be practical to devote any significant effort to manual preprocessing of training material. For example, large-scale editing of transcriptions to make them closer to verbatim, hand-labelling them with prosodic information, or annotating seed material for them to be tagged by part of speech, is out of the question.
- The statistical information required for SBD – and, indeed, for language modeling more generally – will be different for each of the many domains to be handled, and so it must be compiled in a (nearly) automatic fashion for each one.
- Moore's law notwithstanding, the sheer volume of material being processed in DictationNet, and the many stages it has to go through, mean that the CPU and memory requirements of each stage, including SBD, are important considerations.
- The style of the dictations processed is in some ways intermediate between those of read and spontaneous speech. This means that while sentence boundaries must be inferred, there is often room for debate about their "correct" placement. The punctuation choices made by transcriptionists (when working from the audio alone) are often mutually inconsistent, and frequently wrong from the standpoint of the grammar and punctuation rules of more formal written language. Thus our training data can be viewed as somewhat noisy; but on the up side, a boundary decision made by an algorithm may be acceptable to the end user even if it is not what a particular transcriptionist would have done, or if it would be inappropriate in normal written language.

With these constraints in mind, we have concentrated on algorithms with the following run-time characteristics.

- We take as input only what our recognizer offers us. Currently, we use only the words recognized and the lengths of any intervening pauses. Future work may also take account of the length of the word before each (putative) boundary, but pause duration is likely to be the most important clue ([1]).
- We train the purely lexical component of our model on reference text alone. Reference text is available in quite

large quantities, while training involving prosody additionally requires producing, storing, and aligning recognition output. This is only feasible (and, fortunately, only necessary, since far fewer parameters need training) for smaller quantities of material.

- Because we can never fully align recognized and transcribed material, we adopt a "piecemeal" approach to prosodic training, rather than trying to train HMM's (or other models) on whole documents as e.g. [1, 2] do. We align at the character (not word) level, and take each successful alignment of two word boundaries, with any associated pause length and the few words on either side, as a separate training datum.

- Run-time memory limits mean that we only look at a two-word context on either side of a word boundary: using word trigrams would involve too many parameters, and in any case turns out to buy us very little.

- Rather than imposing a fixed threshold on the probability-estimate (or other scoring) value to decide on a boundary, as might be required in interactive recognition, we take advantage of the off-line nature of our processing to impose a threshold that will break the dictation as a whole into sentences of an average length that is characteristic of the domain from which it comes. As well as ensuring that we end up with what will appear to be neither a large surfeit nor a large deficit of boundaries, this also allows us to consider only relative probabilities, and thus to forget about many normalization issues. In future, it will also mean we can bring into play stylistic constraints, to keep sentence lengths within reasonable upper and lower limits.

We evaluated several algorithms with these characteristics. All take a Bayesian approach as their starting point. One algorithm in the sequence corresponds to the current state of the art, and the final algorithm improves significantly on it.

2. Bayesian Approaches

2.1. Derivation

At each inter-word position in our recognizer output, we require an estimate of the probability $P(B = 1|W_1, W_2, L, W_3, W_4)$ of a sentence boundary (B), given the length L of the (possibly zero-length) pause there and the identity of the two words on each side: W_1 and W_2 to the left and W_3 and W_4 to the right. We simplify by taking the underlying linguistic event sequence as W_1, W_2, B, W_3, W_4, where B is unknown, and treat L and $\underline{W} = (W_1, W_2, W_3, W_4)$ as independent except via B. By Bayes' rule, we have:

$$\begin{aligned} P(B|\underline{W}, L) &= \frac{P(\underline{W}, B|L)}{P(\underline{W}|L)} \\ &= \frac{P(\underline{W}, B|L)}{P(\underline{W}, B|L) + P(\underline{W}, \neg B|L)} \\ &= \frac{f_B}{f_B + f_{\neg B}} \end{aligned} \qquad (1)$$

where, using our independence assumption and Bayes' rule,

$$\begin{aligned} f_B &= P(\underline{W}, B|L) \\ &= \frac{P(\underline{W}, B, L)}{P(L)} \\ &\approxeq P(\underline{W}, B)\frac{P(L|B)}{P(L)} \\ &= P(\underline{W}, B)\frac{P(B|L)}{P(B)} \end{aligned} \qquad (2)$$

Our trigram approximation on the linguistic events allows us to further approximate the linguistic term here, $P(\underline{W}, B)$, as

$$P(W_1, W_2)P(B|W_1, W_2)P(W_3|W_2, B)P(W_4|B, W_3) \qquad (3)$$

which again using Bayes' rule we can rewrite as

$$\begin{aligned} &P(W_1, W_2) \times P(B|W_1, W_2) \\ &\times \tfrac{P(B|W_2, W_3)P(W_2, W_3)}{P(B=i|W_2)P(W_2)} \\ &\times \tfrac{P(B|W_3, W_4)P(W_3, W_4)}{P(B=i|W_3)P(W_3)} \end{aligned} \qquad (4)$$

Omitting terms not involving B because they will cancel between numerator and denominator in Equation 1, we have

$$\begin{aligned} f_B \quad \propto \quad & \frac{P(B|L)}{P(B)} \times \frac{P(B|W_1, W_2)}{P(B|W_2)} \\ & \times P(B|W_2, W_3) \times \frac{P(B|W_3, W_4)}{P(B|W_3)} \end{aligned} \qquad (5)$$

All the terms in this expression can be estimated by training on suitable corpora.

If we ignore the possibility of multiple sentence boundaries occurring within a trigram – i.e. of sentences of length one, which are very rare (less than 0.4% of sentences) in our data – then because of the use of the approximation in Equation 3, this approach is in fact equivalent to the HMM-based event language model of [2], apart from the following differences. Firstly, we are only trying to distinguish sentence boundaries and non-boundaries, whereas [2] also look for repairs. Secondly, we use a trigram model, in contrast to the 4-gram model of [2], because we found no significant advantage in the latter for our data and methods. Thirdly – and this may actually give us an advantage over [2] – our final estimate, Equation 5, only requires us to estimate probabilities of boundaries, not of words, so we will be able (see below) to adopt smoothing techniques more directly suited to estimating a variable with only two values, rather than Katz backoff [3] which only makes sense when we are predicting a variable that ranges over the whole vocabulary.

2.2. Estimating Component Probabilities

We estimate the pause-based and word-based components of Equation 5 in different ways.

The estimates of $P(B|L)$ were derived as follows. First, to allow for the fact that different speakers will tend to produce different pause lengths, we in fact use the relative pause length for L rather than the absolute value. This is defined as follows: a zero-length pause has relative length zero, and the Kth shortest pause out of N non-zero pauses will have relative length $\frac{K-\frac{1}{2}}{N}$, approximating a uniform distribution over [0,1]. Secondly, each pause in the training set was assigned to one of 26 "buckets": bucket 0 for all the zero-length pauses, and buckets 1 to 25 for the other relative lengths, in order. An estimate for a run-time pause of length L could then be derived directly from bucket 0 for $L = 0$, or by interpolation between the nearest buckets for $L > 0$.

In contrast, the linguistic estimates were smoothed by interpolating higher-order N-grams with lower. We interpolated $P(B|W_2)$ and $P(B|W_3)$ with the zerogram estimate $P(B|.)$; $P(B|W_1, W_2)$ with $P(B|W_2)$; $P(B|W_3, W_4)$ with $P(B|W_3)$; and the "cross-boundary" value $P(B|W_2, W_3)$ with $\sqrt{P(B|W_2)P(B|W_3)}$.

Rather than use Katz backoff, which is inapplicable when the variable being predicted has only two possible values, we chose the interpolation weights to be sensitive in a different way to the numbers of observations of the relevant contexts. Thus, when we approximate our estimate

$$\hat{P}(X|C) = (1 - \lambda)\hat{P}_{backoff}(X) + \lambda N_{CX}/N_C \tag{6}$$

we allow λ to be an increasing function of the number of observations N_C of the context C, so that the maximum likelihood higher-order estimate N_{CX}/N_C is given more weight the more reliable it is. Experiments on the data available to us suggested that $\lambda(N) = \frac{N}{N+\overline{S}}$ is close to optimal, where $\overline{S}$ is the average sentence length, and is certainly better than using any fixed λ.

2.3. Experimental Protocol

We evaluated our approaches on three separate domains, A, B and C, with rather different amounts of available training material. Domain A is relatively free-form material, while domains B and C are more typical of the kinds of reports DictationNet usually handles, having many headings and frequently-occurring fixed expressions. After all punctuation symbols other than periods had been deleted, domain A had 31.5M words of (reference) training material, domain B had 0.9M words, and domain C had 1.5M. Vocabulary sizes were around 14,000 for A and 8,000 for the other two domains. Trigram perplexities on held-out test data (to be described below), using Chen-Goodman smoothing [4], were 38 for A, 108 for B, and 62 for C. However, these differences are largely due to the very different training set sizes; when each is restricted to 0.9M words, the perplexities come out as 102, 108 and 71.

For each domain, we trained the linguistic components of Equation 5 on the reference material. We trained $P(B|L)$ on (partially) aligned reference material and recognizer output from our test jobs, holding out five successive roughly equal-sized segments of it (from distinct dictations) each time for actual testing and averaging the results. Our overall test (and prosody-training) material consisted of 190 jobs from 7 speakers for domain A, yielding 50117 aligned boundaries; 159 jobs from 2 speakers for B, yielding 27428 boundaries; and 322 jobs from 3 speakers for C, with 47240 boundaries. A minority of sentence boundaries in the test material were in fact marked by the speaker saying the word "period"; where this word had been recognized, we deleted it, along with the shorter of the pauses (if any) to its immediate left and right.

The threshold for a positive decision on a sentence boundary was set so as to make the proportion of such boundaries in each test segment roughly equal to that in the held-out prosody-training segments, rather than in the larger linguistic training set. This single parameter can be safely and more accurately estimated from the smaller, more speaker-specific set.

Because a large majority of word boundaries in our material are not sentence boundaries, even a default strategy of never hypothesizing a boundary can achieve over 90% accuracy. Rather than reporting accuracy statistics, therefore, we follow [5] in using the F-score measure from information retrieval, defined as $F = \frac{2PR}{P+R}$ for precision P (the proportion of selected boundaries that actually were sentence boundaries) and recall R (the proportion of actual boundaries that were selected). Our thresholding turned out to be accurate enough that P and R were always within a few per cent of each other, so that F alone is a good measure of the performance of the algorithm. One can also view $1 - 2\beta F$, for some $\beta \leq 1$, as the effective error rate, where β is the proportion of the algorithm's disagreements with the reference data that a human reader would view as errors on the part of the algorithm. Because of characteristics of our environment already alluded to, we believe β may be around 0.5 to 0.7 in our case, although we have not evaluated it formally.

Data set	Domain	Method			
		0	1	2	3
Rec	A	*63.60*	70.15	70.08	**72.03**
	B	*46.68*	50.34	50.92	51.74
	C	*67.21*	69.65	70.18	**71.32**
INC	A	*78.95*	*81.51*	82.70	**83.65**
	B	*81.00*	83.20	83.83	84.77
	C	*78.06*	81.36	81.50	**83.03**
Ref	A	*77.75*	81.55	81.97	**83.53**
	B	*80.23*	81.05	81.38	**82.31**
	C	*80.56*	*82.81*	83.15	**84.29**

Table 1: F scores (%) for all conditions

We report results for three versions of each domain's test data. Firstly, for all the (successfully aligned) boundaries; secondly, just for those boundaries whose two immediate neighbouring words were correctly recognized; and thirdly, for all the boundaries, but simulating perfect word recognition by using the reference words. These datasets will be called "Rec" (recognized), "INC" (immediate neighbours correct), and "Ref" (reference), respectively. "INC" is of interest because it contains just those boundaries which the transcriptionist will not have to visit anyway to correct the words around them; inserting or deleting a period by hand is less costly when it is part of a larger operation involving word correction.

At several points below, we claim that one result is significantly different from another. We tested significance by collecting the positions in the test data where one method predicted a boundary and the other did not, and carrying out a sign test (two tail, p=0.05, approximating binomials to normals) on the null hypothesis that each method was equally likely to be right.

2.4. Results

The F scores for each domain and dataset are given in Table 1.

"Method 1" in the table is the Bayesian approach we have been discussing, while "Method 0" is the same but without any pause information being used. The other methods will be explained as we proceed. For now, it is worth noting that overall performance ("Rec") on domain B is significantly worse than on A and C, but where word correctness can be partially ("INC") or fully guaranteed ("Ref"), the difference disappears. This is because recognition accuracy for B happens to be much less good.

As a first possible improvement, we derive Method 2, by generalizing Equation 5 to include a model combination weight

([1], p15), replacing $\frac{P(B|L)}{P(B)}$ in Equation 5 by $(\frac{P(B|L)}{P(B)})^\lambda$. We optimize λ on each held-out fourth-fifths of the test data and test on the other one-fifth, and average the results. The figures (again in Table 1) suggest that for some conditions, the contribution of the pause data is being weighted wrongly in the pure Bayesian approach (Method 1), and that when this is corrected, small improvements can result. We found an improvement in eight out of the nine cases, two of which were statistically significant. Interestingly, the average λ value over the nine conditions was 0.78, suggesting that the pure Bayesian approach (with λ fixed at 1) may tend to overvalue the pause data slightly.

Because of the similarities and differences already noted, Method 2 can be taken as at least equivalent to the existing state of the art, represented by [1, 2], although the absolute values are rather different because of the different material. We therefore use Method 2 as the standard of comparison for all the other methods we evaluate. In Table 1, italics are used for results that are significantly worse than those of Method 2, and bold for results – from Method 3, to which we will now turn – that are significantly better.

3. An Improved Approach

Like the approach of [1, 2], the Bayesian approaches just evaluated may turn out to be suboptimal because the linguistic parts of their estimators are trained on the assumption of correct recognizer output; and, given our constraints, they have to be so trained, because we cannot recognize, align and use the quantities of material that would be required to base them on actual recognizer output. However, it is likely that bigram-based estimators will be more errorful than unigram-based, and the situation is also complicated by the facts that we smooth from bigrams to unigrams, and that since the unigram estimators appear in the denominator of Equation 5, this expression is effectively using them to offset the predictions of the bigram-based ones. Furthermore, recognition accuracy may differ in the pre-, cross- and post-boundary positions. A simple way to model all of these possibilities is to generalize Equation 5 still further to redefine f_i as proportional to

$$\frac{P(B|L)^{\lambda_1}}{P(B)^{\lambda_2}} \times \frac{P(B|W_1,W_2)^{\lambda_3}}{P(B|W_2)^{\lambda_4}} \times P(B|W_2,W_3)^{\lambda_5} \times \frac{P(B|W_3,W_4)^{\lambda_6}}{P(B|W_3)^{\lambda_7}} \quad (7)$$

for positive weights λ_i. The results from using the expression in Equation 7, again with the λ_i's optimized on held-out sets and results averaged, are shown as Method 3 in Table 1. For seven conditions, these show a significant improvement; for the other two, there was an improvement, but it was not (on its own) significant. On average, there was an absolute improvement of 1.31% for "Rec", 1.14% for "INC", and 1.21% for "Ref". For the "INC" datasets, which contain the cases of most interest, where a sentence boundary error means an extra task for the transcriptionist, our F scores are over 80%, which means there should be a 6-7% reduction in the number of sentence boundary errors they need to correct. We therefore conclude that Method 3 is a useful advance on the state of the art.

The only clear trend to be seen in the weight values was that the pause-length weight, λ_1, was consistently higher for "Rec", where recognition errors caused the linguistic components to be less reliable. There was a less marked tendency for unigram components to receive slightly smaller weights than the bigram ones, and for the cross-boundary bigram predictor, $P(B|W_2,W_3)$, to be weighted more heavily than the other two bigram predictors.

We verified that including pause lengths does benefit Method 3 as well as Method 2. We found that the F-scores over all nine conditions drop by an average of 4.2% absolute when pause lengths are excluded from Method 3, compared to 3.5% for Method 2. We concluded that pause data is still very useful, and that in fact Method 3 may make slightly better use of it than Method 2 does.

Finally, we tried reducing the number of bigram parameters using an entropy-based criterion, in order not to make excessive memory demands at run-time. Space limitations prevent the details being presented here, but we found that limiting the size of any one bigram parameter set to being twice the vocabulary size (rather than forty times, in the case of domain A) worsened our scores by only 0.5-1%.

4. Summary and Conclusions

We have shown how the practical constraints imposed by the requirements of a large-scale dictation transcription system determine many of the characteristics of possible sentence boundary detection algorithms. Within this environment, we first developed and evaluated a method at least equivalent to the previous state of the art, formulated in such a way as to allow a smoothing method directly tailored to boundary detection rather than word prediction. We then showed how this can be improved upon by a more general method that delivers significantly better performance in three different domains. Our improved method is robust against drastically reducing the memory footprint required for the parameter sets.

Future work will involve determining whether there are useful predictions to be extracted from the durations of the words on either side of a possible boundary, and from their syntactic parts of speech (if we can extract these automatically). There are also almost certain to be important inter-speaker differences in the words and pause lengths occurring at boundaries, and it will doubtless be effective to model them explicitly. It may be that some of the improvement of our Method 3 over the prior state of the art is because it is better able to model the speaker-specific characteristics in the relatively small speaker sets our test data is drawn from, an effect we could exploit further.

5. References

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

[2] Stolcke, A., Shriberg., A., Bates, R., Ostendorf, M., Hakkani, D., Plauche, M., Tür, G., and Lu, Y. "Automatic detection of sentence boundaries and disfluencies based on recognized words". Proceedings of ICSLP, Sydney, 1998.

[3] Katz, S.M. "Estimation of probabilities from sparse data for the language model component of a speech recognizer." IEEE ASSP, 35(3), pp400-401, 1987.

[4] Chen, S., and Goodman, J. "An empirical study of smoothing techniques for language modeling". Technical Report TR-10-98, Harvard University, 1998.

[5] Beeferman, D., Berger, A., and Lafferty, J. "Cyberpunc: a lightweight punctuation annotation system for speech". Proceedings of ICASSP, pp689-692, Seattle, WA, 1998.