AUTOMATIC LINGUISTIC SEGMENTATION OF CONVERSATIONAL SPEECH

Andreas Stolcke *Elizabeth Shriberg*

Speech Technology and Research Laboratory
SRI International, Menlo Park, CA 94025
stolcke@speech.sri.com ees@speech.sri.com

ABSTRACT

As speech recognition moves toward more unconstrained domains such as conversational speech, we encounter a need to be able to segment (or resegment) waveforms and recognizer output into linguistically meaningful units, such a sentences. Toward this end, we present a simple automatic segmenter of transcripts based on N-gram language modeling. We also study the relevance of several word-level features for segmentation performance. Using only word-level information, we achieve 85% recall and 70% precision on linguistic boundary detection.

1. INTRODUCTION

Today's large-vocabulary speech recognizers typically prefer to process a few tens of seconds of speech at a time, to keep the time and memory demands of the decoder within bounds. For longer inputs, the waveform is usually presegmented into shorter pieces based on simple acoustic criteria, such as nonspeech intervals (e.g., pauses) and turn boundaries (when several speakers are involved). We refer to such segmentations as *acoustic segmentations*.

Acoustic segmentations generally do not reflect the linguistic structure of utterances. They may fragment sentences or semantic units, or group together spans of unrelated units. We examine several reasons why such behavior is undesirable, and propose that *linguistic segmentations* be used instead. This requires algorithms for automatically finding linguistic units. In this paper we report on first results from our ongoing efforts toward such an automatic linguistic segmentation. In all further discussion, unless otherwise noted, the terms 'segment,' 'segmentation,' etc. will refer to *linguistic* segmentations.

2. THE IMPORTANCE OF LINGUISTIC SEGMENTATION

Acoustic segmentations are inadequate in cases where the output of a speech recognizer is to serve as input for further processing based on syntactically or semantically coherent units. This includes most natural language (NL) parsers or NL understanding or translation systems. For such systems, the fragmented recognition output would have to be put back together and large spans of unrelated material would need to be resegmented into linguistic units.

Automatic detection of linguistic segments could also improve the user interface of many speech systems. A spoken language system could use the knowledge incorporated in an automatic segmenter to help end-point a user's speech input. A speech indexing and retrieval system (such as for transcribed broadcast audio) could process its data in more meaningful units if the locations of linguistic segment boundaries were known.

Our main motivation for the work reported here comes from speech language modeling. Experiments at the 1995 Johns Hopkins Language Modeling Workshop showed that the quality of a language model (LM) can be improved if both training and test data are segmented linguistically, rather than acoustically [8]. We showed in [10] and [9] that proper modeling of filled pauses requires knowledge of linguistic segment boundaries. We found for example that segment-internal filled pauses condition the following words quite differently from segment-initial filled pauses. Finally, recent efforts in language modeling for conversational speech, such as [8], attempt to capitalize on the internal structure of utterances and turns. Such models are formulated in terms of linguistic units and therefore require linguistic segmentations to be applicable.

3. METHOD

Our main goal for this work was to examine to what extent various kinds of lexical (word-based) information were useful for automatic linguistic segmentation. This precluded a study based on the output of existing speech recognition systems, which currently achieve about 40-50% word error rate on the type of data used in our experiments. At such high error rates, the analysis of any segmentation algorithm and the features it uses would likely be confounded by the unreliable nature of the input data. We therefore chose to eliminate the problem of inaccurate speech recognition and tested our algorithms on hand-transcribed word-level transcripts of spontaneous speech from the Switchboard corpus [4]. An additional benefit of this approach is that the models employed by the segmentation algorithms can also be directly used as language models for speech recognizers for the same type of data, an application we are pursuing as well.

The segmentation approaches we investigated all fell within the following framework. We first trained a statistical language model of the N-gram variety to model the distribution of both words and segment boundaries. (For this purpose, segment boundaries were represented as special tokens `<s>` within the text.) The segmentation information was removed from the test data, and the language model was used to hypothesize the most probable locations of segment boundaries. The resulting segmentations were then evaluated along a number of metrics.

As training data, we used 1.4 million words of Switchboard transcripts annotated for linguistic segmentations by the UPenn Treebank project [7], comprising a total of 193,000 segments. One half of the standard Switchboard development test set, totaling 10,000 words and 1,300 segments, was used for testing.

The hand-annotated segments encompassed different kinds of linguistic structures, including

- Complete sentences
- Stand-alone phrases
- Disfluent sentences aborted in mid-utterance[1]
- Interjections and back-channel responses

The following excerpt illustrates the character of the data. Linguistic segment boundaries are marked <s>, whereas acoustic segmentations are indicated by //.

```
B.44:  Worried that they're not going to
get enough attention?  <s> //

A.45:  Yeah, <s> and, uh, you know, colds
and things like that <laughter> get -- //

B.46:  Yeah.  <s> //

A.47:  -- spread real easy and things,
<s> but, // and they're expensive <s> and,
// <lipsmack> // course, // there's a lot
of different types of day care available,
too, // you know, where they teach them
academic things.  <s> //

B.48:  Yes.  <s> //
```

This short transcript shows some of the ubiquitous features of spontaneous speech affecting segmentation, such as

- Mismatch between acoustic and linguistic segmentations (A.47)
- segments spanning several turns (A.45 and A.47)
- backchannel responses (B.46)

4. THE MODEL

The language models used were of the N-gram type commonly used in speech recognition [5]. In N-gram models, a word w_n from a $n-1$ word history $w_1 \dots w_{n-1}$. If the history contains a segment boundary <s>, it is truncated before that location. During testing, the model is run as a *hidden segment model*, hypothesizing segment boundaries between any two words and implicitly computing the probabilities of all possible segmentations.

Associated with each word position are two states, **S** and **NO-S**, corresponding to a segment starting or not before that word. A forward computation yields the likelihoods of the states at each position k:

$$\begin{aligned}
P_{\textbf{NO-S}}(w_1 \dots w_k) &= P_{\textbf{NO-S}}(w_1 \dots w_{k-1}) \times \\
&\quad p(w_k | w_{k-2} w_{k-1}) \\
&\quad + P_{\textbf{S}}(w_1 \dots w_{k-1}) \times \\
&\quad p(w_k | \texttt{<s>} w_{k-1}) \\
P_{\textbf{S}}(w_1 \dots w_k) &= P_{\textbf{NO-S}}(w_1 \dots w_{k-1}) \times \\
&\quad p(\texttt{<s>} | w_{k-2} w_{k-1}) p(w_k | \texttt{<s>}) \\
&\quad + P_{\textbf{S}}(w_1 \dots w_{k-1}) \times \\
&\quad p(\texttt{<s>} | \texttt{<s>} w_{k-1}) p(w_k | \texttt{<s>})
\end{aligned}$$

A corresponding Viterbi algorithm is used to find the most likely sequence of **S** and **NO-S** (i.e., a segmentation) for a given word string. This language model is a full implementation of the model approximated in [8]. The hidden disfluency model of [10] has a similar structure. As indicated in the formulae above, we currently use at most two words of history in the local conditional probabilities $p(\cdot|\cdot)$. Longer N-grams can be used if more state information is kept.

The local N-gram probabilities are estimated from the training data by using Katz backoff with Good-Turing discounting [6].

5. RESULTS

5.1. Baseline Segmentation Model

The first model we looked at models only plain words and segment boundaries in the manner described. It was applied to the concatenation of all turns of a conversation side, with no additional contextual cues supplied. During testing, this model thus operates with very minimal information, i.e., with only the raw word sequence to be segmented. Table 1 shows results for bigram and trigram models. The performance metrics used are defined as follows. *Recall* is the percentage of actual segment boundaries hypothesized. *Precision* is the percentage of hypothesized boundaries that are actual. *False Alarms (FA)* are the fraction of potential boundaries incorrectly hypothesized as boundaries. *Segment Error Rate (SER)* is the percentage of actual segments identified without intervening false alarms.

Table 1: Baseline model performance

Model	Recall	Precision	FA	SER
Bigram	65.5%	56.9%	1.9%	58.9%
Trigram	70.2%	60.7%	2.0%	53.1%

As can be seen, word context alone can identify a majority of segment boundaries at a modest false alarm rate of about 2%. The trigram model does better than the bigram, but this is expected since it has access to a larger context around potential segment boundaries. to use in its decision. Given these results, we only consider trigram models in all following experiments.

[1] Although complete and disfluent sentences were marked differently in the corpus, we modeled these with a single type of boundary token.

5.2. Using Turn Information

Next we examined a richer model that incorporated information about the turn-taking between speakers.[2] Note that turn boundaries are already present in acoustic segmentations, but in this case we will only use them as a cue to the identification of linguistic segments. Turn information is easily incorporated into the segmentation model by placing special tags at turn boundaries (in both training and testing). Model performance is summarized in Table 2.

Table 2: Segmentation performance using turn information

Model	Recall	Precision	FA	SER
Baseline	70.2%	60.7%	2.0%	53.1%
Turn-tagged	76.9%	66.9%	1.8%	44.9%

As can be seen, adding turn information improves performance on all metrics. This improvement occurs even though turn boundaries are far from perfectly correlated with segment boundaries. As illustrated earlier, turns can contain multiple segments, or segments may span multiple turns.

5.3. Using Part-of-Speech Information

So far we have used only the identity of words. It is likely that segmentation is closely related to syntactic (as opposed to lexical) structure. Short of using a full-scale parser on the input we could use the parts of speech (POS) of words as a more suitable representation from which to predict segment boundaries. Parts of speech should also generalize much better to contexts containing N-grams not observed in the training data (assuming the POS of the words involved is known).

We were able to test this hypothesis by using the POS-tagged version of the Switchboard corpus. We built two models based on POS from this data. Model I had all words replaced by their POS labels during training and test, and also used turn boundary information. Model II also used POS labels, but retained the word identities of certain word classes that were deemed to be particularly relevant to segmentation. These retained words include filled pauses, conjunctions, and certain discourse markers such as “okay,” “so,” “well,” etc. Results are shown in Table 3.

Table 3: Segmentation performance using POS information

Model	Recall	Precision	FA	SER
Word-based	76.9%	66.9%	1.8%	44.9%
POS-based I	68.9%	58.5%	2.0%	59.3%
POS-based II	79.6%	73.5%	0.9%	39.9%

We see that POS tags alone (Model I) do not result in better segmentations than words. The fact that Model II performs better than both the all-word based model and the pure POS model indicates that certain function words that tend to occur in the context of segment boundaries provide some of the strongest cues for these boundaries. Apart from these strong lexical cues, it seems to be helpful to abstract from word identity and use POS information instead. In other words, the tag set could be optimized to provide the right level of resolution for the segmentation task.

It should be noted that the results for POS-based models are optimistic in the sense that for an actual application one would first have to tag the input with POS labels, and then apply the segmentation model. The actual performance would be degraded by tagging errors.

5.4. Error Trade-offs

As an aside to our search for useful features for the segmentation task, we observe that we can optimize any particular language model by trading off recall performance for false alarm rate, or vice versa. We did this by biasing the likelihoods of **S** states by some constant factor, causing the Viterbi algorithm to choose these states more often. Table 4 compares two bias values, and shows that the bias can be used to increase both recall and precision, while also reducing the segment error rate.

Table 4: Biasing segmentation

Model	Recall	Precision	FA	SER
Bias = 1	76.9%	66.9%	1.8%	44.9%
Bias = 2	85.2%	69.2%	2.7%	37.4%

6. DISCUSSION

6.1. Error Analysis

To understand what type of errors the segmenter makes, we hand-checked a set of 200 false alarms generated by the baseline trigram model. The most frequent type (34%) of false alarm corresponded to splitting of segments at sentence-internal clause boundaries, e.g., false alarms triggered by a conjunction that would be likely to start a segment. For example, the `<s>` in the segmentation

```
i'm not sure how many active volcanos
there are now <s> and and what the amount
of material that they do uh put into the
atmosphere
```

represents a false alarm, presumably triggered by the following coordinating conjunction “and.”

5% of the false alarms could be attributed to filled pauses at the end of segments, which were often attached to the following segment. This actually reflects a labeling ambiguity that should not be counted as an error. Another 7% of the false alarm we deemed to be labeling errors. Thus, a total of 12% of false alarms could be considered to be actually correct.

6.2. Other Segmentation Algorithms

Our language-model-based segmentation algorithm is only one of many that could be used to perform the linguistic segmentation task, given a set of features. Conceptually, segmentation is just another

[2] Speakers can talk over each other. We did not model this case separately; instead, we adopted the serialization of turns implied by the transcripts.

classification problem, in which each word transition must be labeled as either a segment boundary or a within-segment transition. Two natural choices for alternative approaches are decision trees and a transformation-based, error-driven classifier of the type developed by Eric Brill for other tagging problems [2]. Both of these methods would make it easier to combine diverse input features that are not readily integrated into a single probabilistic language model, e.g., if we wanted to use both POS and word identity for each word.[3] Our approach, on the other hand, has the advantage of simplicity and efficiency. Furthermore, the language model used for segmentation can also be used for speech decoding or rescoring.

We already mentioned that if POS information is to be used for segmentation, an automatic tagging step is required. This presents somewhat of a chicken-and-egg problem, in that taggers typically rely on segmentations. An appealing solution to this problem in the statistical tagging framework [3] would be to model both segmentation and tag assignment as a single hidden Markov process.

6.3. Other Features for Segmentation

All of our experiments were based on lexical information only. To further improve segmentation performance, and to make it less dependent on accurate speech recognition, we plan to combine the LM approach with a model for various acoustic and prosodic correlates of segmentation. These include:

- Unfilled pause durations
- Fundamental frequency patterns
- Phone durations
- Glottalization

Our current segmentation model deals with each conversation side in isolation. An alternative approach is to model the two sides jointly, thereby allowing us to capitalize on correlations between the segment structure of one speaker and what is said by the other. It is likely, for example, that backchannel responses would be modeled better this way.

7. CONCLUSIONS

We have argued for the need for automatic speech segmentation algorithms that can identify linguistically motivated, sentence-level units of speech. We have shown that transcribed speech can be segmented linguistically with good accuracy by using an N-gram language model for the locations of the hidden segment boundaries. We studied several word-level features for possible incorporation in the model, and found that best performance so far was achieved with a combination of function ‘cue’ words, POS labels, and turn markers.

Acknowledgments

This research was supported by DARPA and NSF, under NSF Grant IRI-9314967. The views herein are those of the authors and should not be interpreted as representing the policies of DARPA or the NSF.

[3] Such an integration can be achieved in a language model using the maximum entropy paradigm [1], but this would make the estimation process considerably more expensive.

8. REFERENCES

1. A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.
2. E. Brill. Some advances in transformation-based part of speech tagging. In *Proceedings of the 12th National Conference on Artificial Intelligence*, Seattle, WA, 1994. AAAI Press.
3. K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Second Conference on Applied Natural Language Processing*, pages 136–143, Austin, Texas, 1988.
4. J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume I, pages 517–520, San Francisco, March 1992.
5. F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*. Morgan Kaufmann, San Mateo, Ca., 1990.
6. S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, March 1987.
7. M. Meteer et al. Dysfluency annotation stylebook for the Switchboard corpus. Distributed by LDC, February 1995. Revised June 1995 by Ann Taylor.
8. M. Meteer and R. Iyer. Modeling conversational speech for speech recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, May 1996.
9. E. Shriberg and A. Stolcke. Word predictability after hesitations: A corpus-based study. In *Proceedings International Conference on Spoken Language Processing*, Philadelphia, PA, October 1996.
10. A. Stolcke and E. Shriberg. Statistical language modeling for speech disfluencies. In *Proceedings IEEE Conference on Acoustics, Speech and Signal Processing*, volume I, pages 405–408, Atlanta, GA, May 1996.