High Performance Segmentation of Spontaneous Speech Using Part of Speech and Trigger Word Information

Marsal Gavaldà
Interactive Systems Labs.
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
marsal+@cs.cmu.edu

Klaus Zechner
Interactive Systems Labs.
Comp. Ling. Program
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213, USA
zechner+@andrew.cmu.edu

Gregory Aist
Comp. Ling. Program
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213, USA
aist+@andrew.cmu.edu

Abstract

We describe and experimentally evaluate an efficient method for automatically determining small clause boundaries in spontaneous speech. Our method applies an artificial neural network to information about part of speech and trigger words.

We find that with a limited amount of data (less than 2500 words for the training set), a small sliding context window (+/−3 tokens) and only two hidden units, the neural net performs extremely well on this task: less than 5% error rate and F-score (combined precision and recall) of over .85 on unseen data.

These results prove to be better than those reported earlier using different approaches.

1 Introduction

In the area of machine translation, one important interface is that between the speech recognizer and the parser. In the case of human-to-human dialogues, the speech recognizer's output is a sequence of *turns* (a contiguous segment of a single speaker's utterance) which in turn can consist of multiple clauses.

Lavie *et al.* (1996) discuss that using smaller units rather than whole turns can greatly facilitate the task of the parser since it reduces the complexity of its input.

The problem is thus how to correctly segment an utterance into clauses.

The segmentation procedure described in Lavie *et al.* (1996) uses a combination of acoustic information, statistical calculation of boundary-trigrams, some highly indicative keywords and also some heuristics from the parser itself.

Stolcke and Shriberg (1996) studied the relevance of several word-level features for segmentation performance on the Switchboard corpus (see Godfrey *et al.* (1992)). Their best results were achieved by using part of speech n-grams, enhanced by a couple of trigger words and biases.

Another, more acoustics-based approach for turn segmentation is reported in Takagi and Itahashi (1996).

Palmer and Hearst (1994) used a neural network to find sentence boundaries in running text, i.e. to determine whether a period indicates end of sentence or end of abbreviation. The input to their network is a window of words centered around a period, where each word is encoded as a vector of 20 reals: 18 values corresponding to the word's probabilistic membership to each of 18 classes and 2 values representing whether the word is capitalized and whether it follows a punctuation mark. Their best result of 98.5% accuracy was achieved with a context of 6 words and 2 hidden units.

In this paper we bring their idea to the realm of speech and investigate the performance of a neural network on the task of turn segmentation using parts of speech, indicative keywords, or both of these features to hypothesize segment boundaries.

2 Data preparation

For our experiments we took as data the first 1000 turns (roughly 12000 words or 12 full dialogues) of transcripts from the Switchboard corpus in a version that is already annotated for parts of speech (e.g. *noun*, *adjective*, *personal pronoun*, etc.).

The definition of a *small clause* which we wanted the neural network to learn the boundaries of is as follows: Any finite clause that contains an inflected verbal form and a subject (or at least either of them, if not possible otherwise). However, common phrases such as *good bye*, *and stuff like that*, etc. are also considered small clauses.

Preprocessing the data involved (i) expansion of some contracted forms (e.g. *I'm* $\rightarrow$ *I am*), (ii) correction of frequent tagging errors, and (iii) generation of segment boundary candidates using some simple heuristics to speed up manual editing.

Thus we obtained a total of 1669 segment boundaries, which means that on average approximately after every seventh token (i.e. 14% of the text) there is a segment boundary.

3 Features and input encoding

3.1 Features

The transcripts are tagged with part of speech (POS) data from a set of 39 tags[1] and were processed to extract *trigger* words, i.e. words that are frequently near small clause boundaries (<b>). Two scores were assigned to each word w in the transcript according to the following formulae:

$$\text{score}_{\text{pre}}(w) = C(w\texttt{<b>})\ \hat{P}(w\texttt{<b>}|w)$$
$$\text{score}_{\text{post}}(w) = C(\texttt{<b>}w)\ \hat{P}(\texttt{<b>}w|w)$$

where C is the number of times w occurred as the word (before/after) a boundary, and $\hat{P}$ is the Bayesian estimate for the probability that a boundary occurs (after/before) w.

This score is thus high for words that are likely (based on $\hat{P}$) and reliable (based on C) predictors of small clause boundaries.

The pre- and post-boundary trigger words were then merged and the top 30 selected to be used as features for the neural network.

3.2 Input encoding

The information generated for each word consisted of a data label (a unique tracking number, the actual word, and its part of speech), a vector of real values $x_1, ..., x_c$ and a label ('+' or '−') indicating whether a segment boundary had preceded the word in the original segmented corpus.

The real numbers $x_1, ..., x_c$ are the values given as input to the first layer of the network. We tested three different encodings:

1. Boolean encoding of POS: x_i $(1 \leq i \leq c = 39)$ is set to 0.9 if the word's part of speech is the i^{th} part of speech, and to 0.1 otherwise.
2. Boolean encoding of triggers: x_i $(1 \leq i \leq c = 30)$ is set to 0.9 if the word is the i^{th} trigger, and to 0.1 otherwise.
3. Concatenation of boolean POS and trigger encodings ($c = 39 + 30 = 69$).

4 The neural network

We use a fully connected feed-forward three-layer (input, hidden, and output) artificial neural network and the standard backpropagation algorithm to train it (with learning rate $\eta = 0.3$ and momentum $\alpha = 0.3$).

Given a window size of W and c features per encoded word, the input layer is dimensioned to $c \times W$ units, that is W blocks of c units.

The number of hidden units (h) ranged in our experiments from 1 to 25.

[1]The tagset is based on the standard tagsets of the Penn Treebank and the Brown Corpus.

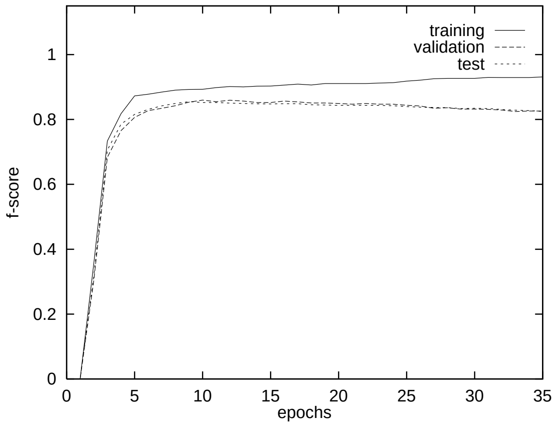

Figure 1: Training the neural network. (Net with POS and trigger encoding, $W = 6$, $h = 2$, $\theta = 0.7$)

As for the output layer, in all the experiments it was fixed to a single output unit which indicates the presence or absence of a segment boundary just before the word currently at the middle of the window. The actual threshold to decide between segment boundary and no segment boundary is the parameter θ which we varied from 0.1 to 0.9.

The data was presented to the network by simulating a sliding window over the sequence of encoded words, that is by feeding the input layer with the $c \times W$ encodings of, say, words $w_i ... w_{i+W-1}$ and then, as the next input to the network, shifting the values one block (c units) to the left, thereby admitting from the right the c values corresponding to the encoding of w_{i+W}. Note that at the beginning of each speaker turn or utterance the first $c \times (\frac{W}{2} - 1)$ input units need be padded with a "dummy" value, so that the first word can be placed just before the middle of the window. Symmetrically, at the end of each turn, the last $c \times (\frac{W}{2} - 1)$ input units are also padded.

5 Results and discussion

We created two data sets for our experiments, all from randomly chosen turns from the original data: (i) the "small" data set (a 20:20:60(%) split between training, validation, and test sets), and (ii) the "large" data set (a 60:20:20(%) split).

First, we ran 180 experiments on the "small" data set, exhaustively exploring the space defined by varying the following parameters:

- encoding scheme: POS only, triggers only, POS and triggers.
- window size: $W \in \{2, 4, 6, 8\}$
- number of hidden units: $h \in \{2, 10, 25\}$
- output threshold: $\theta \in \{\ 0.1, 0.3, 0.5, 0.7, 0.9\ \}$

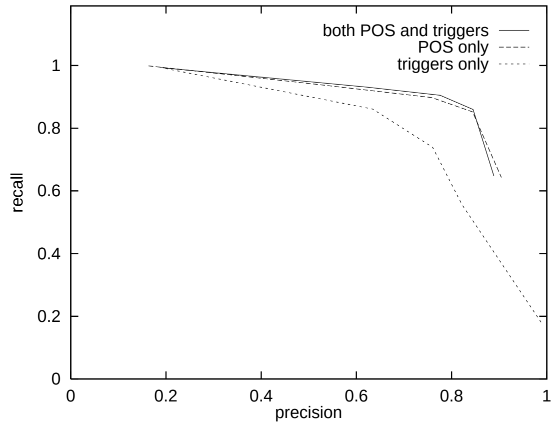

Figure 2: Precision vs. recall tradeoff. (On unseen data, net with $W = 6$, $h = 2$, $0.1 \leq \theta \leq 0.9$)

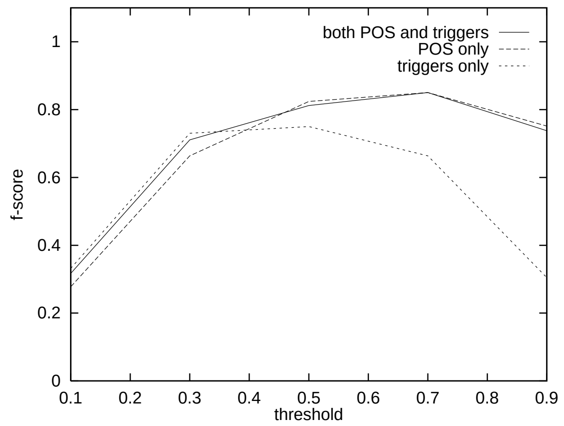

Figure 3: F-scores as a function of the output unit threshold θ. (On unseen data, net with $W = 6$, $h = 2$)

Precision (number of correct boundaries found by the neural network divided by total number of boundaries found by the neural network), *recall* (number of correct boundaries found by the neural network divided by true number of boundaries in the data) and *F-score* (defined as $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$) were computed for each training, validation and test sets.

To be fair, we chose to take the epoch with the maximum F-score on the validation set as the best configuration of the net, and we report results from the test set only. Figure 1 shows a typical training/learning curve of a neural network.

The best performance was obtained using a net with 2 hidden units, a window size of 6 and the output unit threshold set to 0.7. The following results were achieved.

classification rate	precision	recall	F-score
95.8%	0.845	0.860	0.852

Some general trends are observed:

- As the window size gets larger, the performance increases, but it seems to peak at around size 6.
- Fewer hidden units yield better results; generally we get the best results for just two hidden units.
- The global performance as measured by the proportion of correct classifications (i.e. both '+' and '−') increases as the F-score increases.
- High performance (correct classifications >95%, F-score >0.85) is easily achieved.
- The optimal threshold for a high F-score lies in the $0.5 \leq \theta \leq 0.7$ interval.
- Varying the threshold leads to a tradeoff of precision vs. recall.

To illustrate the last point, we present a graph that shows a comparison between the three encoding methods used, for a window size of 6 (Figure 2). The combined method is only slightly better than the POS method, but they both are clearly superior to the trigger-word method. Still it is interesting to note that quite a reasonable performance can be obtained just by looking at the 30 most indicative pre- and post-boundary trigger-words. Noteworthy is also the behavior of the precision–recall curves: with our method a high level of recall can be maintained even as the output threshold is increased to augment precision.

In Figure 3, we plot the F-score against the threshold. Whereas for the encodings POS only and POS and triggers, the peaks are in the region between 0.5 and 0.7, for the triggers only encoding, the best F-scores are achieved between 0.3 and 0.5.

We also ran another 30 experiments with the "large" data set focusing on the region defined by the parameters that achieved the best results in the preceding experiments (i.e. window size 6 or 8, threshold between 0.5 and 0.7, number of hidden units between 1 and 10). Under these constraints, F-scores vary slightly, always remaining between .85 and .88 for both validation and test sets.

Within this region, therefore, several neural nets yield extremely good performance.

While Lavie *et al.* (1996) just report an improvement in the end-to-end performance of the JANUS speech-to-speech translation system when using their segmentation method but do not give details the performance of the segmentation method itself, Stolcke and Shriberg (1996) are more explicit and provide precision and recall results. Moreover Lavie *et al.* (1996) deal with Spanish input whereas Stolcke and Shriberg (1996), like us, drew their data from the Switchboard corpus.

Type	Harmful?	Reason	Context
false positive	no	trigger word	_to work_ <b> _and * when I had_
false positive	yes	non-clausal _and_	_work off * and on_
false negative	yes	speech repair	<b> _but * and they are_
false positive	?	trigger word	_he you know * gets to a certain_
false positive	yes	non-clausal _and_	_if you like trip * and fall or something_
false negative	yes	speech repair	<b> _we * that's been_
false positive	no	CORRECT	<b> _but i think * its relevance_
false negative	no	CORRECT	<b> _and she * she was_
false negative	yes	embedded relative clause	_into nursing homes * die very quickly_
false positive	no	trigger word	_wait lists * and all_

Table 1: Sample of misclassifications (on unseen data, net with encoding of POS and triggers, $W = 6$, $h = 2$, $\theta = 0.7$). *False positive* indicates an instance where the net hypothesizes a boundary where there is none. *False negative* indicates an instance where the net fails to hypothesize a boundary where there is one. A '<b>' indicates a small clause boundary. A '*' indicates the location of the error.

Thus here we compare our approach with that of Stolcke and Shriberg (1996). They trained on 1.4 million words and in their best system, achieved precision .69 and recall .85 (which corresponds to an F-score of .76). We trained on 2400 words (i.e. over 500 times less training data), and we achieved an F-score of .85 (i.e. a 12% improvement).

6 Error analysis

Table 1 shows 10 representative errors that one of the best performing neural network made on the test set. 25 randomly selected errors were used to do the error analysis, which consisted of 14 false positives and 11 false negatives. 8 of the errors were errors we considered to be harmful to the parser, 3 were errors of unknown harmfulness, and the remaining 14 were considered harmless.

Of the harmful errors, three were due to the word *and* being used as a conjunction in a non-clausal context, two were due to a failure to detect a speech repair, and one was due to an embedded relative clause (*most people that move into nursing homes * die very quickly*).

The network was also able to correctly identify some mistagged data (marked as CORRECT in Table 1).

These results suggest that adding features relevant to speech repairs (such as whether words were repeated) or features relevant to detecting the use of *and* as a non-clausal conjunct might be useful in achieving better accuracy.

7 Conclusion

We have shown that using neural networks for automatically segmenting turns in conversational speech into small clauses reaches a level of less than 5% error rate and achieves good precision/recall performance as measured by an F-score of more than .85.

These results outperform those obtained by other methods as reported in the literature.

Future work on this problem includes issues such as optimizing the set of POS tags, adding acoustic/prosodic features to the neural network, and using it for *pro*-drop languages like Spanish to assess the relative importance of POS vs. trigger word weights and to examine the performance of the system for languages where POS tags may not be as informative as they are for English.

8 Acknowledgements

The work reported in this paper was funded in part by grants from ATR – Interpreting Telecommunications Research Laboratories of Japan, the US Department of Defense, and the Verbmobil Project of the Federal Republic of Germany.

This material is based on work supported under a National Science Foundation Graduate Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development. In *Proceedings of the ICASP-92*, vol. I, pp. 517–520.

A. Lavie, D. Gates, N. Coccaro, and L. Levin. 1996. Input segmentation of spontaneous speech in JANUS: a speech-to-speech translation system. In *Proceedings of the ECAI-96*.

D. D. Palmer and M. A. Hearst. 1994. Adaptive sentence boundary disambiguation. In *Proceedings of the ANLP-94*.

A. Stolcke and E. Shriberg. 1996. Automatic linguistic segmentation of conversational speech. In *Proceedings of the ICSLP-96*, pp. 1005–1008.

K. Takagi and S. Itahashi. 1996. Segmentation of spoken dialogue by interjections, disfluent utterances and pauses. In *Proceedings of the ICSLP-96*, pp. 697–700.