Towards Incremental End-of-Utterance Detection in Dialogue Systems

Michaela Atterer, Timo Baumann, David Schlangen
Institute of Linguistics
University of Potsdam, Germany
{atterer,timo,das}@ling..uni-potsdam.de

Abstract

We define the task of *incremental* or *0-lag utterance segmentation*, that is, the task of segmenting an ongoing speech recognition stream into utterance units, and present first results. We use a combination of hidden event language model, features from an incremental parser, and acoustic / prosodic features to train classifiers on real-world conversational data (from the Switchboard corpus). The best classifiers reach an F-score of around 56%, improving over baseline and related work.

1 Introduction

Unlike written language, speech—and hence, automatic speech transcription—does not come segmented into units. Current spoken dialogue systems simply wait for the speaker to turn silent to segment their input. This necessarily reduces their responsiveness, as further processing can only even commence a certain duration after the turn has ended (Ward et al., 2005). Moreover, given the typically simple domains, such work mostly does not deal with the problem of segmenting the turn into utterances, i.e. does not distinguish between *utterance* and *turn* segmentation. However, as our corpus shows (see below), multi-utterance turns are the norm in natural dialogues. The work that does treat intra-turn utterance segmentation does so in an offline context, namely the post-processing of automatic transcripts of recorded speech such as meeting protocols (Fung et al., 2007), and relies heavily on right-context pause information.

In this paper, we define the task of *incremental* or *0-lag utterance segmentation*, that is, the task of segmenting an ongoing speech recognition stream into utterance units using only left-context information.[1] This work is done in the context of developing an incremental dialogue system architecture (as proposed among others by (Aist et al., 2007)), where, ideally, a considerable part of the analysis has already been done while the speaker still speaks. The incremental parser and other components of such a system need to be reset at turn-internal utterance-boundaries with as little delay as possible. Hence it is of vital importance to predict the end-of-utterance while the last word of a sentence is processed (or even earlier). We investigate typical features an incremental system can access, such as partial parse trees and parser internal information. These experiments are a first important step towards online endpointing in an incremental system.

2 Data

We used section 2 of the Switchboard corpus (Godfrey et al., 1992) for our experiments. Section 3 was used for training the language models and the parser that we used. Some of the Switchboard dialogues are of a very low quality. We excluded those where transcription notes indicated high rate of problems due to static noise, echo from the other speaker or background noise. As our parser became very slow for long sentences, we excluded sentences that were longer than 25 words from the

[1] 0-lag here refers to the time where feature extraction starts. As the modules on which feature extraction is based require processing time themselves, a complete absence of prediction delay is of course not possible.

Coling 2008: Companion volume – Posters and Demonstrations, pages 11–14
Manchester, August 2008

analysis (4% of the sentences). We also excluded back-channel utterances (typically one-word turns) from the corpus.

Of the remaining corpus we only used the first 100,000 instances to reduce the computational load for training the classifiers. 80 % of those were used as a training corpus, and 20 % as a test corpus. For follow-up experiments that investigated turn-initial or turn-internal utterance boundaries only (see below), we used the relevant subsets of the first 200,000 instances.

3 Feature Extraction

Our features comprise prosodic features, and syntactic features.

Prosodic features are *pitch*, *logarithmized signal energy* and derived features, extracted from the audio every 10 ms. In order to track changes over time, we derive features by windowing over past values of pitch, energy, and energy in voiced regions only, with window sizes ranging from 50 ms to 5000 ms. We calculate the arithmetic *mean* and the *range* of the values, the *mean difference* between values within the window and the *relative position* of the minimum and maximum. We also perform a linear regression and use its *slope*, the *MSE* of the regression and the *error* of the regression for the last value in the window.

As classification was done word-wise (final vs. non-final word), each word was attributed the prosodic features of the last corresponding 10-ms-frame.

For the extraction of **syntactic features** we used both n-gram models and a parser. The parser was a modified version of Amit Dubey's `sleepy` parser,[2] which can produce syntactic structure incrementally online. The n-gram model was a hidden event model as typically used in the sentence unit detection literature (see e.g. (Fung et al., 2007)). For the time being, all features based on word identities are computed on gold standard transcriptions. We trained n-gram models both based on words and on words plus POS-information that was incrementally obtained from the parser.[3] We calculated the log-probability of trigrams with the last token in the n-gram being a place-holder for end-of-utterance (i.e. the probability of (*I,would,end-of-utterance*) or (*Thank,you,end-of-utterance*). We also calculated log probabilities for trigrams such as (*I, end-of-utterance-1,end-of-utterance*). Thirdly, the log probability was also computed for a string consisting of 4 word/POS-pairs followed by an end marker.

Further syntactic features can be roughly divided into two classes: *parser-based features*, which are related to internal states of the parser, and *structure-based features* which refer to properties of the syntactic tree. The former try to capture the expectation of there being more incomplete edges towards the beginning of a sentence than towards the end. We also might expect a relative decrease in the overall number of edges towards the end of a sentence. Therefore we track a selection of numbers referring to the various kinds of edges stored in the chart. Moreover, we utilize some of the parser's information about the best intermediate edge, and use the category after the dot of this edge as an estimate for the most probable category to come next. Furthermore, we use the forward probability of the best tree as a rounded log probability.

The structure-based features are simple features such as the part-of-speech category of the current word and the number of the word in the current utterance, and more complex features that try to (roughly) approximate semantic notions of completeness by counting the number of verbs or number of nominal phrases encountered, as we would usually expect a sentence to be incomplete if we haven't heard a verb or nominal phrase yet. For example, in sentences of the structure *(NP) (VP (V NP))* or *(NP) (VP (V NP (N PP)))*, humans would typically be aware that the last phrase has probably been reached during the last noun phrase or prepositional phrase (cf. (Grosjean, 1983)). However, the length and internal structure of these phrases can vary a great deal. We try to capture some of this variation by features referring to the last non-terminal seen, the second-to-last non-terminal seen and the number of words seen since the last non-terminal. A number of features (the *count features*) are simple features that record the number of words since the turn or utterance started and the time elapsed since the utterance started. They are also subsumed under syntactic features.

We also used dialogue act features like the previous dialogue act, and the previous dialogue act of the last speaker. Those currently come from

[2] `http://homepages.inf.ed.ac.uk/adubey/software/`

[3] The models were trained using the SRILM-tools (Stolcke, 2002) for $n = 3$ using Chen and Goodman's modified Kneser-Ney discounting (Chen and Goodman, 1998).

the gold standard. We assume that in a dialogue system the system would at least have information about its own dialogue acts.

4 Experimental Settings

We tested a number of classifiers as implemented in the Weka toolkit (Witten and Frank, 2005), and found that the JRip-classifier, an implementation of the RIPPER-algorithm (Cohen, 1995), performed best. A number of attribute selection algorithms also did not result in a significant change of performance. Therefore, we only report the plain results by JRip. We also tested the impact each of our information sources had on the results. The aim was to find out how important parser, part-of-speech information and pitch and energy features are, respectively.

As turn-internal utterance-ends might be more difficult to detect than those that coincide with turn-ends, we repeat the experiment with turn-internal utterances only. Deleting turn-final utterances from our initial 200,000-instance corpus resulted in 128,686 remaining word instances, 80 % of which were used for training. For a third experiment, where we look at turn-initial utterances, we use again a subset of those 200,000 word-instances.

For clarity, we simply use precision/recall for evaluation; see (Liu and Shriberg, 2007) for a discussion of other metrics. As a baseline we assume non-existent utterance segmentation, which results in a recall of 0 and a precision of 100 %.

5 Results

	Precision	Recall	F
baseline	100	0	0
all features	73.8	45.0	55.9
all syntactic features	74.8	44.0	55.4
word/POS n-gram features	73.4	45.8	56.4
word n-gram features	66.9	34.7	45.7
only count features	59.3	7.7	13.6
prosodic features only	49.5	8.3	14.2
pitch features	100	0	0
energy features	48.2	7.4	12.8

Table 1: Results for end-of-utterance classification for all utterances.

Tables 1 and 2 show the results for the experimental settings described above. Dialogue act features were included in the syntactic features, but JRip did not use them in its rules eventually. Table 1 shows that the overall F-score is best when n-grams with POS information are used. Adding a parser, however, increases precision. Prosody features in general do not seem to have much of an influence on end-of-utterance prediction in our data, with energy features contributing more than pitch features. Table 2 indicates, as expected, that

	Precision	Recall	F
baseline	100	0	0
all features	71.2	40.3	51.4
all syntactic features	72.7	38.2	50.0
word/POS n-gram features	70.5	41.1	51.9
word n-gram features	70.9	26.4	38.5
only count features	60.4	1.0	2.0
prosodic features only	41.7	1.7	3.3
pitch features	100	0	0
energy features	31.6	1.2	2.3

Table 2: Results for end-of-utterance classification for utterances which are not turn-final.

the end of an utterance is harder to predict when it is not turn-final. Performance drops compared to the results shown in Table 1. Note that some of the performance drop must be attributed to the use of a different data set. However, the performance drop is much more dramatic for the experiments where only prosody is used than for those where syntax is used. We speculate that the count features also loose their impact because one-word utterances like 'Okay' are usually turn-initial.

The results shown in Tables 1 and 2 can be regarded as an upper bound for a dialogue system, because our experiments so far work with gold standard sentence boundaries for creating syntactic features (e.g., for resetting our parser). Strictly speaking, they are only realistic for turn-initial utterances. For the remaining 14,737 of our 26,401 utterances, we therefore report a lower bound, where we use only features that do not have knowledge about the beginning of the sentence (Table 3). No count and parser-based features were used for this experiment, only n-gram (word-based without POS information) and pitch features. The Table also shows the results for the 11,664 turn-initial utterances, where we use all features.[4] We then derive the overall performance from the fractions of initial and non-initial utterances.

Future work aims at putting together a system where the parser is restarted using predictions based on its own output.

[4]Note that turn-initial utterances can at the same time be turn-final.

	Precision	Recall	F
non-initial	65.0	28.6	39.7
initial	74.5	55.1	63.3
overall	70.4	40.3	51.3

Table 3: Results for end-of-utterance classification for utterances which are not turn-initial (reduced feature set), and utterances that are turn-initial (full feature set) and the derived overall performance.

6 Related Work

(Fuentes et al., 2007) report F-measures of 84% using prosodic features only, but they use left–right-windows for feature calculation, where our processing is truly incremental and more suitable for real-time usage in a dialogue system. Moreover, they only seem to use one-utterance turns, which makes the task easier when prosodic features are used. In our dialogue corpus (Switchboard, section 2), however, each turn contains on average 2.5 utterances, and turn-internal utterances also need to be recognized. (Fung et al., 2007) reach an F-score of 75.3% , but report that the best feature was pause-duration—a feature we don't use because we want to find out how well we can predict the end of a sentence before a pause makes this clear. Similarly, (Ferrer et al., 2002) rely largely on pause features. (Schlangen, 2006) investigates incremental prediction of end-of-utterance and end-of-turn for various pause-lengths, and achieves an F-score of 35.5% for pause length 0, on which we can improve here.

7 Discussion and Conclusion

We investigated *0-lag end-of-utterance detection* for incremental dialogue systems. In our setup, we aim to recognise the end of an utterance as soon as possible, while the potentially last word is processed, without the help of information about subsequent silence. We investigate a number of features an incremental system would be able to access, such as information from an incremental parser. We find that remaining (non-pause) prosodic information is not as helpful as in non-incremental studies, especially for non-turn-final utterances. Syntactic information, on the other hand, increases performance. Future work aims at more sophisticated prosodic modelling and at testing the impact of using real or simulated speech recognition output. We also intend to implement end-of-utterance prediction in the context of a real incremental system we are building.

8 Acknowledgement

This work was funded by DFG ENP SCHL845/3-1.

References

Aist, Gregory, James Allen, Ellen Campana, Carlos Gomez-Gallo, Scott Stoness, Mary Swift, and Michael Tanenhaus. 2007. Incremental understanding in human-computer dialogue and experimental evidence for advantages over non-incremental methods. In *Proc. of the 2007 Workshop on the Semantics and Pragmatics of Dialogue (DECALOG)*.

Chen, S.F. and J. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Center for Research in Computing Technology (Harvard University).

Cohen, William W. 1995. Fast effective rule induction. In *Machine Learning: Proceedings of the Twelfth International Conference*.

Ferrer, L., E. Shriberg, and A. Stolcke. 2002. Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody in human-computer dialog. In *Proc. Intl. Conf. on Spoken Language Processing*, Denver.

Fuentes, Olac, David Vera, and Thamar Solorio. 2007. A filter-based approach to detect end-of-utterances from prosody in dialog systems. In *Proc. Human Language Technologies 2007*, Rochester, New York.

Fung, J., D. Hakkani-Tur, M. Magimai-Doss, E. Shriberg, S. Cuendet, and N. Mirghafori. 2007. Prosodic features and feature selection for multi-lingual sentence segmentation. In *Proc. Interspeech*, pages 2585–2588, Antwerp.

Godfrey, John J., E. C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proc. of ICASSP-1992*, pages 517–520, San Francisco, USA, March.

Grosjean, François. 1983. How long is the sentence? Prediction and prosody in the on-line processing of language. *Linguistics*, 21:501–529.

Liu, Y. and E. Shriberg. 2007. Comparing evaluation metrics for sentence boundary detection. In *Proc. IEEE ICASSP*, Honolulu,USA.

Schlangen, David. 2006. From reaction to prediction: Experiments with computational models of turn-taking. In *Proc. Interspeech 2006, Panel on Prosody of Dialogue Ac ts and Turn-Taking*, Pittsburgh, USA.

Stolcke, Andreas. 2002. SRILM – an extensible language modeling toolkit. In *Proc. ICSLP 2002*.

Ward, Nigel G., Anais G. Rivera, Karen Ward, and David G. Novick. 2005. Root causes of lost time and user stress in a simple dialog system. In *Proc. of Interspeech*, El Paso, USA.

Witten, Ian H. and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.