Multi-pass sentence-end detection of lecture speech

Madina Hasan, Rama Doddipatla and Thomas Hain

The University of Sheffield, Sheffield, United Kingdom.

[m.hasan,r.doddipatla,t.hain]@sheffield.ac.uk

Abstract

Making speech recognition output readable is an important task. The first step here is automatic sentence end detection (SED). We introduce novel F0 derivative-based features and sentence end distance features for SED that yield significant improvements in slot error rate (SER) in a multi-pass framework. Three different SED approaches are compared on a spoken lecture task: hidden event language models, boosting, and conditional random fields (CRFs). Experiments on reference transcripts show that CRF-based models give best results. Inclusion of pause duration features yields an improvement of 11.1% in SER. The addition of the F0-derivative features gives a further reduction of 3.0% absolute, and an additional 0.5% is gained by use of backward distance features. In the absence of audio, the use of backward features alone yields 2.2% absolute reduction in SER.

Index Terms: Sentence end detection, punctuation, capitalisation, conditional random fields.

1. Introduction

With the advances in computational capabilities, large amounts of digital audio and video data are produced on a daily basis. Applying automatic speech recognition (ASR) to these valuable resources of information allows their content to be easily usable and retrievable. However, typically the standard ASR output is a stream of single words, where sentence boundary, punctuation and case information are not available. Enriching the ASR output with such information improves the readability and intelligibility of the transcribed text. This step is also useful for further text based natural language processing (NLP) which typically requires formatted text containing sentence information, punctuation and capitalisation. The need for sentence end information in NLP tasks was discussed by several researchers. For instance, the work in [1] showed that knowing sentence boundaries in text can improve unsupervised dependency parsing. Jones [2] demonstrated that enhancing text with periods improves the readability of ASR transcripts. In the field of information extraction, Makhoul et al. [3], Favre et al. [4], and many others reported that punctuation marks (specifically, commas and periods) can significantly improve accuracy. Mrozinski et al. [5] studied the impact of sentence segmentation on the readability and usability of ASR output, and consequently on summarisation accuracy.

Common punctuation marks in a spoken text may contain *full stop*, *comma*, *question mark*, *exclamation mark*, *colon* and *semicolon*. The occurrence of these marks varies widely with the type of the speaking domain. For instance, conversational speech contains more questions when compared with broadcast news. A study on the Wall Street Journal corpus showed that *comma* and *full stop* are the dominant punctuation marks [6]. As less frequent events are more difficult to predict, most studies focus on the detection of *full stop* and *comma*.

The work in this paper addresses automatic sentence end detection (SED) for lecture transcripts. Experiments in this paper compare the performance of boosting, hidden event language models (HELM) and conditional random fields (CRFs). Notably systems based on CRFs give the best performance. However, all of these approaches are not capable of including long range statistics, such as the length of the current sentence. Consequently, distance features are introduced, which require a two-pass strategy for SED. We examined the effect of these distance features firstly using only text features to see the effect on systems where only text features are available. Secondly, using different combinations of prosodic features to see the effect on systems with audio data available in addition to text (audio data allows the extraction of additional prosodic information). The proposed distance features showed significant performance improvement over both our text-only baseline systems and systems that combine both text and prosodic features.

Though the frame-level fundamental frequency conveys long term information about speakers' pitch, using it as a raw feature is not robust as it fails to capture the variability of the pitch across the speakers and speaking context [7]. We introduce derivative based F0 feature extraction, which results in significant performance improvement.

The rest of the paper is organised as follows: §2 discusses previous work on SED, while §3 discusses the approaches and §4 describes the features used in experiments presented in this paper. All experimental work is described in §5, which includes the description of data, features, and experimental set up. Conclusions summarise the findings at the end.

2. Previous work

Restoration of punctuation was addressed by many studies for both textual and spoken language. Punctuation restoration of spoken language is generally more challenging and typically makes use of both text and speech related types of information.

Previous studies on punctuation in speech used lexical information [6, 8, 9], prosodic information [10] or both [11, 12]. In [9], only text features are used for detecting sentence boundaries on Switchboard reference transcripts. A window of three words and the associated parts-of-speech tags in both directions of the current word are used, together with individual words in the current window. These text features are then fed into Boostexter [13] which implements one of boosting family algorithms. They reported a recall of 58.5%, and 63.8% precision. The prosodic and the text features were combined in [11, 12] using Hidden Markov Models (HMM) framework, for detecting sentence ends in read and spontaneous speech. Results showed that systems using both prosodic and text features always outperform the use of either feature. On reference transcripts, they reported a boundary error rate (BER) of 3.3% for a broadcast

news task and 4.0% on telephone speech corpora. Liu et al. [14] compared the performance of maximum entropy, CRFs, and HMM models for sentence boundary detection on broadcast news (BN) and conversational telephone speech (CTS), using lexical and prosodic features. CRFs models were shown to outperform the other models, but the best performance on reference transcripts (CTS:26.43%, BN:48.21%) and ASR transcripts (CTS:36.26%, BN: 57.23%) were obtained with a majority voting of the three approaches. For punctuation prediction on 50 hours of French and English broadcast news and podcast data, from the Quaero project [15], Kolar et.al. [16] used both textual and prosodic features to train boosting models. The textual features were extracted using a word n-gram language model, up to 4-grams. Prosodic features included pause duration, pitch features, and durations of vowel and final rhymes. It was shown that for both languages textual-based models outperformed the prosodic-based models. Best performance was achieved when both feature types were used, achieving a slot error rate (SER) of 65.3% on average when considering English reference transcripts.

3. Approaches

In the following, the key algorithmic approaches used in this paper are described.

3.1. Hidden Event Language Model: Baseline

As a first baseline hidden event language model (HELM) approach was implemented, which originally was proposed for disfluencies detection [17]. In the HELM framework, sentence ends are treated as hidden events, as they are not observable in the spoken content. While standard language models are typically used to predict the next word, given word history, the LM here is used to estimate the probability of the occurrence of sentence end at the end of each observed word, given its context. Given a sequence of words, $W = w_0\ w_1\ w_2 \cdots\ w_n$, the model predicts the sequence of inter-words events, $E = e_0\ e_1\ e_2 \cdots e_n$, using a quasi-HMM framework. Word/event pairs represent the states of the model, while the event type represents the hidden state. The observations are previous words, and the probabilities are obtained through a standard language model.

3.2. Boosting

Boosting [18] is a machine learning classification technique that combines simple rules (also called weak classifiers) into a single more accurate rule. Those classifiers are built sequentially, such that the each new classifier focuses on the training examples that were misclassified before. Given a set of labelled training examples pairs $(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$, where, y_i is the label associated with the observation instance x_i, boosting maintains a set of weights over the set of training examples, for each iteration $t = 1, 2, \cdots, T$. The distribution of those weights reflects the importance of each training example. With respect to those weights, a weak hypothesis with the lowest error is chosen. Updating the weights is done such that the weak learner is forced to focus on the difficult examples. In this work, the Adaboost [13] algorithm implementation provided by the ICSIboost [19] tool, is used.

3.3. Conditional Random Fields

Linear-Chain conditional random fields (CRFs) are discriminative models that have been intensively used for sequence labelling and segmentation purposes [20, 21]. The model aims to estimate and directly optimise the posterior probability of the label sequence, given a sequence of features (hence the frequently used term *direct model*). Let $\mathbf{x}$ be the observation and $\mathbf{y}$ be the label sequence, a first-order linear-chain CRF model is defined by

$$p_\lambda(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_t \sum_k \lambda_k f_k(\mathbf{x}, y_{t-1}, y_t, t)\right), \quad (1)$$

where $Z(\mathbf{x})$ is a normalisation term, λ_k are the model parameter (one weight for each feature), and the set of feature functions f_k can be defined over the entire observations with time step t. The CRF++ [22] toolkit was used in this work.

4. Features

Boosting and CRFs allow the use of feature functions. While both boosting and random field theories allow for continuous-valued features to be included, best performance is often obtained by quantisation of continuous valued input. In the following we briefly introduce the features used in the experiments in §5.

4.1. Textual features

Text-based features include n-grams for $n = 2, 3,$ and 4, and cover up to two following words (also called post words). We define an m-gram feature as

$$h_i^m = (w_{-m}, \cdots, w_{-1}, w_0, \cdots, w_{i-1}, w_i). \quad (2)$$

Here m represents the n-gram count and i represents the overlap into the feature, i.e. the number of post-words. Experiments in this paper make use of 2, 3 and 4 gram contexts with one or two post-words.

4.2. Prosodic features

Prosodic cues are known to be relevant to discourse structure across languages and can therefore play an important role in various information extraction tasks [23]. These cues change within sentences and paragraphs. Thus, they are good indicators of sentence boundaries. In English and related languages, such information is indicated with pausing and change in pitch. Hence, these values are widely used in the literature for detecting sentence boundaries [10, 16].

For the set of experiments presented in this paper, pause duration (PD) and pitch-based (F0) features were used for the SED task. To extract these prosodic features, the reference transcripts were aligned to the audio data using the AMI meeting transcription systems [24]. The exact word timings allows us to compute the duration of pauses at the end of each word, the pause duration feature. The duration feature was extracted on training, development and evaluation sets.

Pitch information was extracted using the ESPS[25] `get_f0` function. Pitch estimates were averaged for a whole word to yield a single value (the F0 feature). Typically, fundamental frequency varies within a sentence. Hence, a first order derivative (F0D) should hold relevant information. Experiments with different methods for extraction of the F0D values were conducted. The best results were obtained by first computing short and long term averages (window lengths 7 and 100 words respectively), over past word-F0 estimates, referred to as $F0_{s,i}$ and $F0_{l,i}$, respectively. Then the first order derivative of the

ratio of long and short term averages is computed using regression window of order 4. In particular, the process of finding the F0D feature for a word with index i, ($F0D_i$), can be described by the following:

1. Normalisation: This includes normalising the values of both small window S_i and large window L_i, using moving average sliding window technique. That is,

$$S_i = F0_{s,i} \circledast w_{s,i}, \quad L_i = F0_{l,i} \circledast w_{l,i},$$

where $w_{s,i}$ and $w_{l,i}$ are the long and short averaging windows, respectively.

2. Differentiation: In this step, the delta coefficients of the ratio $R_i = \frac{S_i}{L_i}$ is computed,

$$F0D_i = R_i \circledast G,$$

where $G = [-q, -q+1, \cdots, q, q+1]$ is the regression window, and q is the order of the regression window.

All continuous-valued features were quantised using CART regression class trees, as available in the *scikit* machine learning tools [26]. The optimal tree depth was found to be 4 and was fixed for all further experiments.

4.3. Multi-pass features

In addition to features discussed above, forward and backward distance features are proposed in this paper. These features are introduced to include long range statistics, measuring how far the current word is from its neighbouring sentence ends in the word sequence, in both directions. The forward (FD) and backward (BD) distance features are quantised using the CART regression tree approach.

The position of the periods is only available for the training data, and is initially meaningless in the test set. For this reason, a multi-pass approach is used to compute the distance features. A first recognition run using a model trained without distance features gives initial boundaries estimates. Based on these estimates, the distance features can then be calculated. For consistency, equivalent operations have to be performed both on training, development and test sets.

5. Experiments

The following section presents our experiments on E-corner data using the approaches discussed in §3 for performing sentence boundary detection.

5.1. Data

E-corner is a corpus of conversational speech consisting of lecture recordings. The corpus is divided into training, development, and evaluation sets. The distribution of punctuations as well as the number of words are summarised in Table 1. The first stage of data processing includes extensive text normalisation as is standard for ASR. For example, this step converts entries such as dates, currency values and any numbers to words. It also makes sure that dots in between abbreviations will not be interpreted as sentence boundaries. Finally, since the task is motivated to recognise sentence boundaries, all occurrences of question and exclamation marks are mapped to periods.

The primary stages in our experiments include feature extraction, model training, tuning the parameters on the development set and finally evaluating the models on the test set. Before proceeding with the experiments, a brief summary of various error measures used for evaluation are presented in the following section.

Set	Words	%Periods	% ? Mark	% ! Mark
Train	831017	5.1	0.57	0.023
Eval	135196	5.1	0.45	0.024
Dev	129716	4.9	0.68	0.031

Table 1: E-Corner data statistics.

Approach	n-gram	%Rec	%Prec	%BER	%SER
HELM	2	40.0	58.8	5.2	88.0
	3	48.4	63.1	4.7	79.9
	4	46.8	64.2	4.7	79.3

Table 2: HELM results on E-corner data.

5.2. Metrics

A variety of metrics [27] are commonly used for evaluating the performance of sentence end detection. These include *precision* (P), *recall* (R), *boundary error rate* (BER) and *slot error rate* (SER). The definitions of these error measures are given below.

$$\text{Prec} = \frac{TP}{TP+FP}, \quad \text{Rec} = \frac{TP}{TP+FN},$$

$$\text{SER} = \frac{FP+FN}{TP+FN}, \quad \text{BER} = \frac{I+M}{N}.$$

Where I, M and N denote the number of insertion, misses and the total number of words, respectively. TP, TF, FP and FN refer to true positive, true false, false positive, and false negative counts, in that order.

5.3. Baseline experiments

The baseline experiments use only text features which include various n-grams. The performance is evaluated using HELM, boosting and CRFs. The results are presented in Tables 2 and 3. One can observe that in all cases, as the n-gram order increases, the performance improves. It is also important to note that the performance of all systems is relatively poor. However, using only this information, both HELM and CRFs seem to perform better than boosting. Based on these results, only 4-grams are used as our baseline models for all further experiments.

5.4. Extending the feature set

In addition to n-gram features, the feature set is extended by a variety of features such as: increasing the number of post-words (PW), pause duration (PD) between words, pitch (F0), differential pitch (F0D) extracted from the audio signals and distance features. The effects of adding these features to the text features, using boosting and CRFs approaches, are presented in Table 4.

One can observe that increasing the post-word depth from one to two improves the performance in both boosting and CRFs, indicating that context has an important role in improving the classifier performance. Surprisingly, including raw pitch (F0) appears to help. However, F0D feature gives significant gains for both boosting and CRFs approaches, when compared with F0 feature. The best performance is achieved using the pause durations between words.

Approach	Feat	%Rec	%Prec	%BER	%SER
Boosting	$(h_0^2+h_1^2)$	37.4	63.2	5.0	84.4
CRF	$(h_0^2+h_1^2)$	42.8	64.1	4.8	81.2
Boosting	$(h_0^3+h_1^3)$	36.3	64.4	5.0	83.8
CRFs	$(h_0^3+h_1^3)$	40.9	65.5	4.8	80.7
Boosting	$(h_0^4+h_1^4)$	36.2	64.9	4.9	83.4
CRFs	$(h_0^4+h_1^4)$	42.4	67.8	4.6	77.8

Table 3: Baseline results on E-corner data comparing boosting and CRFs.

Approach	Feat	%Rec	%Prec	%BER	%SER
Boosting	$(h_0^4+h_1^4+h_2^4)$	38.9	67.0	4.8	80.3
	+F0	40.9	67.5	4.7	78.8
	+F0D	45.1	70.3	4.4	74.0
	+PD	52.0	73.8	3.9	66.5
CRFs	$(h_0^4+h_1^4+h_2^4)$	45.3	70.0	4.4	74.1
	+F0	45.9	72.5	4.2	71.5
	+F0D	48.5	74.3	4.0	68.3
	+PD	54.4	75.8	3.7	63.0

Table 4: Results comparing CRFs and boosting when adding prosodic features

To add the distance features, an additional segmentation pass is needed to compute these features based on the output of the first decoding run. The results for distance features using only CRFs approach are presented, as CRFs always have better performance when compared with boosting approach (see Table 4). The results using the proposed features are presented in Table 5 and Table 6. In Table 5, the performance of both the distance features is studied using the $(h_0^4+h_1^4+h_2^4)$ text-only CRFs baseline model as the initial model for segmenting the data, to show the effect of the distance features for the applications where no audio data is available (and hence no prosodic features can be included). One can observe that the BD feature improves the performance by about 3% relative, while there is hardly any improvement using the FD feature. Since the position of periods in the test data is crucial in extracting the distance feature, another set of experiments using a better baseline model is performed. The experiments were conducted with a PD feature based model ($h_0^4+h_1^4+h_2^4$ + PD). The results are presented in Table 6. There is a small but consistent gain in performance using the BD feature. This shows that distance features are helpful in sentence boundary detection.

5.5. Feature Combination

In previous sections, experiments have shown how various features perform in detecting sentence boundaries. It will be interesting to see whether combining these features can further improve the system performance. In this direction, a variety of feature combination experiments have been performed using the features discussed in previous sections; the results are presented in Table 7. One can observe that PD+F0+F0D gives the best performance without using distance features. Using this model, the distance features are integrated using the multi-pass approach. It can be seen that the backward distance (BD) feature provides the best result, giving a relative gain of 20% in SER when compared to the baseline.

Feat	%Rec	%Prec	%BER	%SER
$(h_0^4+h_1^4+h_2^4)$	45.3	70.0	4.4	74.1
+BD	45.8	72.1	4.3	71.9
+FD	45.3	70.0	4.4	74.1
+BD+FD	45.3	70.0	4.4	74.1

Table 5: Results comparing Distance Features using CRFs and $(h_0^4+h_1^4+h_2^4)$ as the initial model.

Feat	%Rec	%Prec	%BER	%SER
$(h_0^4+h_1^4+h_2^4)$+ PD	54.4	75.8	3.7	63.0
+ BD	55.2	75.7	3.7	62.5
+ FD	54.4	75.8	3.7	63.0
+ BD+FD	54.4	75.8	3.7	63.0

Table 6: Results comparing Distance Features using CRFs and $(h_0^4+h_1^4+h_2^4)$+ PD as the initial model

6. Conclusions

The paper addressed the problem of sentence end detection for lecture transcripts. Experiments were conducted using three different approaches, namely HELM, boosting and CRFs, using a variety of text, prosodic and distance features. A new derivative-based F0 feature was introduced. From our experiments, it is clear that the F0D feature provides better performance gains when compared with F0, consistently for both CRFs and boosting approaches. The paper also proposed a novel distance feature to include long range statistics, such as the length of the current sentence. A multi-pass approach was introduced for computation of the distance features, since they are unknown a-priori. It is shown that the backward distance (BD) feature consistently improves the performance and is also the best result we achieved in feature combination, with a relative gain of 20% in SER. Moreover, BD feature provided a substantial benefit (3% relative), in case where no audio data is available.

7. Acknowledgment

We thank Dr. John Dines from IDIAP Research Institute for sharing the E-corner data and also helping with the configurations. This work is in part supported by the EU FP7 DocuMeet Project `http://www.documeet.eu/the-project`.

8. References

[1] V. I. Spitkovsky, H. Alshawi, and D. Jurafsky, "Punctuation: making a point in unsupervised dependency pars-

Feat	%Rec	%Prec	%BER	%SER
$(h_0^4+h_1^4+h_2^4)$	45.3	70.0	4.4	74.1
+PD	54.4	75.8	3.7	63.0
+PD+F0	55.1	76.1	3.7	62.2
+PD+F0D	56.5	77.4	3.6	60.0
+PD+F0+F0D	56.9	77.5	3.5	59.6
+PD+F0+F0D+FD	56.9	77.5	3.5	59.6
+PD+F0+F0D+BD	57.5	77.5	3.5	59.2
+PD+F0+F0D+BD+FD	56.9	77.5	3.5	59.6

Table 7: Feature Combination Experiments using CRF approach.

ing," in *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, ser. CoNLL '11, 2011, pp. 19–28.

[2] D. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. Reynolds, and M. Zissman, "Measuring the readability of automatic speech-to-text transcripts," in *in Proc. Eurospeech*, 2003, pp. 1585–1588.

[3] J. Makhoul, A. Baron, I. Bulyko, L. Nguyen, L. A. Ramshaw, D. Stallard, R. M. Schwartz, and B. Xiang, "The effects of speech recognition and punctuation on information extraction performance," in *INTERSPEECH'05*, 2005, pp. 57–60.

[4] B. Favre, R. Grishman, D. Hillard, H. Ji, D. Hakkani-Tur, and M. Ostendorf, "Punctuating speech for information extraction," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-april 4 2008, pp. 5013 –5016.

[5] J. Mrozinski, E. Whittaker, P. Chatain, and S. Furui, "Automatic Sentence Segmentation of Speech for Automatic Summarization," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, may 2006, p. I.

[6] D. Beeferman, A. Berger, and J. Lafferty, "Cyberpunc: a lightweight punctuation annotation system for speech," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 2, may 1998, pp. 689 –692 vol.2.

[7] M. K. Sönmez, E. Shriberg, L. P. Heck, and M. Weintraub, "Modeling dynamic prosodic variation for speaker verification." in *ICSLP*. Citeseer, 1998.

[8] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10, 2010, pp. 177–186.

[9] N. K. Gupta and S. Bangalore, "Extracting clauses for spoken language understanding in conversational systems," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, ser. EMNLP '02, 2002, pp. 273–280.

[10] M. Haase, W. Kriechbaum, G. Möhler, and G. Stenzel, "Deriving document structure from prosodic cues." in *INTERSPEECH*, 2001, pp. 2157–2160.

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

[12] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech communication*, vol. 32, no. 1, pp. 127–154, 2000.

[13] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine learning*, vol. 39, no. 2-3, pp. 135–168, 2000.

[14] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1526–1540, 2006.

[15] J. Kolár and L. Lamel, "On Development of Consistently Punctuated Speech Corpora," in *INTERSPEECH*, 2011, pp. 833–836.

[16] ——, "Development and Evaluation of Automatic Punctuation for French and English Speech-to-Text," in *INTERSPEECH*, 2012.

[17] A. Stolcke and E. Shriberg, "Statistical language modeling for speech disfluencies," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1, 1996, pp. 405–408 vol. 1.

[18] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[19] B. Favre, D. Hakkani-Tür, and S. Cuendet, "ICSIBOOST," http://code.google.come/p/icsiboost, 2007.

[20] H. Tseng, "A conditional random field word segmenter," in *In Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[21] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03, 2003, pp. 134–141.

[22] T. Kudoh, "CRF++," http://crfpp.sourceforge.net/, 2007.

[23] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, no. 1-2, pp. 127–154, 2000.

[24] T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grezl, A. el Hannani, M. Huijbregts, M. Karafiat, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech and Language Processing*, Aug. 2011.

[25] "Entropic, ESPS Version 5.0 Programs Manual," 1993.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] Y. Liu and E. Shriberg, "Comparing evaluation metrics for sentence boundary detection," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–185.