Noise-matched training of CRF based sentence end detection models

Madina Hasan, Rama Doddipatla, Thomas Hain

The University of Sheffield, Sheffield, United Kingdom
m.hasan@sheffield.ac.uk, drsanand@gmail.com, t.hain@sheffield.ac.uk

Abstract
Sentence end detection (SED) is an important task for many applications and has been studied on written text and automatic speech recognition (ASR) transcripts. In previous work it was shown that conditional random fields models gave best SED performance on a range of tasks, with and without the inclusion of prosodic features. So far, true transcripts were used for both training and evaluation of SED models. However, in the context of noisy ASR transcripts the performance degrades significantly, especially at medium to high ASR error rates. In this work we demonstrate the correlation of SED performance with word error rate (WER), at different ASR system performance levels. A new method is introduced for transferring SED labels onto noisy ASR transcripts for model training of noise-matched SED models. The proposed method significantly improves the performance of SED models, and provides 11% relative gain in slot error rate when compared with models trained on true transcripts. This paper further investigates the effect of noise-matched trained SED with different features. It is observed that the impact of textual features reduces significantly with low ASR performance. However, prosodic features still have noticeable impact.

Index Terms: Metadata extraction, endpoint detection, Enriching transcription, conditional random field.

1. Introduction
The output from an Automatic Speech Recognition (ASR) is a stream of words that typically has no case or punctuation information, as the necessary information is not spoken, but inferred. Enriching the Speech-To-Text (STT) output with this information can greatly improve the readability and usability for downstream Natural Language Processing (NLP) tasks. 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. They demonstrated that proper sentence segmentation is crucial to improve the performance of summarisation.

In our previous work [6], we presented a study on Sentence End Detection (SED) of lecture speech. It was shown that the performance can be greatly improved using prosodic and so-called multi-pass distance features. The study primarily focused on improving the performance of SED, and only used reference or true transcripts as test data for evaluating the performance of SED models. However, in practice, we rarely have access to true transcripts and test data is obtained using an ASR system. As reported by several studies [7, 8, 9], the performance of SED may drastically degrade even when applied to ASR transcripts with relatively low Word Error Rates (WER). Naturally the amount of errors in ASR output, and their types are the cause. Thus, the performance of SED models is expected to drop more significantly when applied to spontaneous speech, where ASR word error rates are still relatively high.

This paper focuses on improving the performance of the SED models on ASR transcripts at medium to higher word error rates. In particular we train Conditional Random Fields (CRF)-based SED models on noisy ASR transcripts of the training data. So far, to the best of our knowledge, this approach was not reported in the literature.

2. Previous work
Restoration of sentence ends has been addressed by many studies for both written and spoken language. A large number of these studies reported a degradation in the performance of SED when using ASR transcripts, even when the word error rates are low.

In [7, 8] a Boundary Error Rate (BER) of 3.3% for Broadcast News (BN) and 4.0% on Conversational Telephone Speech (CTS) tasks were reported respectively, when using reference transcripts. However, the results degraded significantly to 10.8% and 22.5%, respectively, when processing errorful ASR transcripts with WER of 30.5% for BN and 46.7% for CTS. Liu et al. [10] also reported Sentence Unit (SU) boundary detection results on broadcast news and conversational telephone speech tasks and showed that with WERs of 11.7% for BN and 14.9% for CTS, the performance on ASR transcripts (36.26% for CTS and 57.23% for BN) degrades when compared to performance on reference transcripts (26.43% for CTS and 48.21% for BN).

Kolar et. al [9] reported to achieve the best performance for punctuation prediction, using textual and prosodic features. The experiment was conducted on 50 hours of French and English broadcast news and podcast data, from the Quaero project [11]. They reported a Slot Error Rate (SER) of 65.3% on average when considering English reference transcripts. However, this result deteriorated with ASR transcripts to 77.2%, with a WER of 17.3%.

The studies reported in literature, to the best of our knowledge, train the SED models on reference transcripts even when performing SED on ASR transcripts of the test data. This paper proposes to train the SED models using ASR transcripts of the training data, and show that it improves the performance of SED on ASR transcripts of the test data.

This style of training performs better on both systems; text features only and text plus prosody features. The next section describes the proposed training of SED models on ASR tran-
scripts and the features used in this study.

3. Modeling noise-matched CRF-based SED

In previous work [6] it was shown that conditional random fields (CRFs) [12] based classifiers were outperforming boosting [13] and Hidden Event Language model (HELM) [14] classifiers, with and without the inclusion of prosodic features. Thus in this work we only use the CRFs approach, as implemented in the CRF++ toolkit [15]. Linear-Chain CRFs are discriminative models that have been intensively used for sequence labelling and segmentation purposes [16, 17]. Unlike the Hidden Markov Model (HMM) approach, which models the joint distribution of the label sequence and the observations, CRFs model aims and directly optimises the posterior probability of the label sequence, given a sequence of features (hence the frequently used term direct model).

The degradation in the performance of SED models when applied to ASR transcripts has been reported by several researchers, as discussed in the previous section. We hypothesise that this degradation is due to training SED models on true transcripts and using these models to detect sentence ends on ASR transcripts, which results in a mismatch for detecting sentence boundaries on ASR transcripts. To reduce this mismatch we propose model training of noise-matched SED models. These SED models are trained on ASR transcripts of the training data, and tuned on ASR transcripts of the development data, then used to restore the sentence boundaries on the test set.

3.1. Training SED models on ASR transcripts

Supervised training of SED models requires associating each word in the training data with a label indicating if that word is followed by a sentence end or not. However, these labels do not exist in the ASR output of the training data. The challenge in the proposed method is therefore to restore these labels into the errorful ASR output. The following steps summarise the training procedure for training SED models on ASR transcripts:

1. Prepare reference transcripts: The punctuated reference transcriptions of the training and development sets are processed to tag each word with a label indicating if that word is followed by a sentence end or not. Each segment in the reference transcripts has a unique identifier.

2. Decode the training data: An ASR system is used to perform recognition on the training and development data. This provides lists of the recognised words with their start and end time information.

3. Normalise the ASR transcripts: The recognition output (ASR transcript) is normalised to split all joined words (i.e. by hyphens or underscores).

4. Utterance matching and alignment: The utterances IDs in the ASR transcripts are then matched with the reference segments labels.

5. Align the recognised words with the words in the reference (with the associate tags), for each matched pair. This alignment helps identifying the correct, incorrect, substituted, and inserted words.

6. A new list is then created such that:
   - The correctly and incorrectly (substitution errors) recognised words are kept, but associated with the timing information from the recognised words list and SEs tags from the reference list (tagged reference list).
   - The deleted words are removed, while the inserted words are kept with their recognised timing information from the hypothesised list, but with a tag that indicates not SEs, as physically there was no SE at this location in the reference list.

7. Train the CRFs-based SED models, using the new tagged list. The prosodic features can be extracted using the timing information produced with this list.

Following such an approach allows the system to learn sentence boundaries with errors in ASR transcriptions. Figure 1 illustrates the system used for training the SED models on ASR transcripts. For further discussion in this paper, SED_{crf} is used to refer to models trained on reference or true transcripts and SED_{asr} to refer to models trained on ASR transcripts.

3.2. Features

Two different types of features were used for building the SED models and these include: textual features, prosodic features. All continuous features are modeled using CART style regression trees [18].

Text features include n-grams and the next two words (post words). Generally, we define the textual features as follows:

\[
    \tilde{h}^m_i = \{w_{i-m}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_i\}.
\]

Here \(m\) represents the n-gram order and \(i\) represents the location in the reference list.

Prosodic features include pause duration (PD) and pitch-based (F0) features. In reference transcripts, pause duration features were extracted by aligning the reference transcripts with the audio data using forced alignment. The timing information available with the force alignment process is also used for extracting the pitch values for each word, obtained by averaging the raw pitch values of all the frames. Pitch values are obtained using the ESPS [19] using the get_f0 function. For ASR transcripts, both PD and F0 features were extracted from the timing information provided by the respective ASR models described in the previous section.

4. Experiments and Discussion

This section presents the experiments for performing SED using the proposed approach to train SED models on ASR transcripts. In order to understand the influence of ASR errors on SED, three different ASR systems trained on different data are
used for generating the ASR transcripts both in training and evaluation. The experiments presented in this section look at the performance of SED in three different scenarios: performance of SED models trained on reference transcripts (SED$_{ref}$) and are used for evaluating reference transcripts, performance of SED$_{asr}$ on ASR transcripts and finally the performance of SED models trained on ASR transcripts (SED$_{asr}$) and used for evaluating ASR transcripts. In all cases, the SED models are trained and evaluated on the E-corner lecture data. The data statistics for punctuation are presented in Table 1. The purpose of these experiments is to show that noise-matched training of SED models using ASR transcripts reduces the mismatch between train and test conditions.

The performance of SED is evaluated using the most common metric metrics such as: recall (R), precision (P) and F1, in addition to SER, as described in [9]. Regardless of the ASR system used to produce the ASR transcripts of E-corner test set, the SED models were trained on the train set, and tuned to work at the equal error point on the development set of E-corner data set. Before we proceed further with the discussion, a brief description of the ASR acoustic model training is presented in the next section.

### 4.1. Acoustic models for ASR

Three different ASR models are trained using data from meetings, TED talks and E-corner data. The first model (ASR$_1$) is trained on TED talks data and the second model trained using meeting data (AMI, AMIDA and ICSI) and TED talks (ASR$_2$), and the third model (ASR$_3$) is trained on meeting data, TED-talks and E-corner data. These different models provide variation in ASR transcripts and facilitate us to study the performance of SED to different levels of ASR errors. It is important to note that the performance of SED is still reported only on the E-corner data.

The ASR models are trained as GMM-HMM systems, using bottleneck (BN) features derived from a deep neural network (DNN). The input to the DNN uses 31 adjacent frames of the log filter-bank outputs, which are concatenated and decorrelated with Discrete Fourier Transform (DCT) to form a 368 dimensional feature vector. The filter-bank inputs are mean and variance normalised at the speaker level. Global mean and variance normalisation is performed on each dimension before feeding the input for training the DNN. For our experiments, we chose to use 5 hidden layers with the first 4 hidden layer having 65 units (more details in [20]). The bottleneck (BN) layer is placed just before the output layer and has 39 units. We set aside 15% of the training data for cross validation and use the rest to train the DNN. The training automatically stops once the frame accuracy of the cross validation set falls below 0.1%. DNNs are trained using the TNET toolkit [21]. The GMM-HMM system trained using the BN feature is a tied-state triphone system with 16 mixture Gaussians for each state.

The performance of the different ASR systems on the E-corner train, development (Dev) and test data sets are presented in Table 2. One can observe the %WERs is very high and we believe that this should greatly influence the performance of SED on ASR transcripts. However, it can be noted that using more data to train the DNN helped improve the performance other than adding in-domain E-corner training data for training ASR$_3$. In the next section, we will present the performance of SED on the different scenarios which were discussed in the previous section.

<table>
<thead>
<tr>
<th>Set</th>
<th>Words</th>
<th>% Periods</th>
<th>% ? Mark</th>
<th>% ! Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>831017</td>
<td>5.1</td>
<td>0.57</td>
<td>0.023</td>
</tr>
<tr>
<td>Test</td>
<td>135196</td>
<td>5.1</td>
<td>0.45</td>
<td>0.024</td>
</tr>
<tr>
<td>Dev</td>
<td>129716</td>
<td>4.9</td>
<td>0.68</td>
<td>0.031</td>
</tr>
</tbody>
</table>

Table 1: E-Corner data statistics.

### 4.2. Case 1: SED$_{asr}$ model for performing SED on true transcripts of the test set.

This is the ideal scenario where the true transcripts (i.e. 0 %WER) are used instead of ASR transcripts of the test data. The performance of SED is presented in Table 3 using textual and prosodic features. Refer to [6] for a more detailed study of various features and how they influence the performance of SED using true transcripts. One can observe that the best performance is achieved when combining textual and prosodic features.

**Table 3:** Performance of SED$_{asr}$ model on true transcripts of E-corner test set (Case 1).

<table>
<thead>
<tr>
<th>Features</th>
<th>%Rec</th>
<th>%Prec</th>
<th>%BER</th>
<th>%SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>(h$^2_1$+h$^2_2$+h$^2_3$)</td>
<td>45.3</td>
<td>70.0</td>
<td>4.4</td>
<td>74.1</td>
</tr>
<tr>
<td>+PD</td>
<td>40.9</td>
<td>67.5</td>
<td>4.7</td>
<td>78.8</td>
</tr>
<tr>
<td>+PD+F0</td>
<td>54.4</td>
<td>75.8</td>
<td>3.7</td>
<td>63.0</td>
</tr>
<tr>
<td>+PD+F0</td>
<td>55.1</td>
<td>76.1</td>
<td>3.7</td>
<td>62.2</td>
</tr>
</tbody>
</table>

### 4.3. Case 2: SED$_{asr}$ model for performing SED on ASR transcripts of the test set.

Here the SED models are still trained on the reference or true transcripts and are used for performing SED on ASR transcripts of the test data. Table 4 presents the results only using the textual features and Table 5 presents the results combining textual and pause duration features.

One can easily notice the huge degradation in %SER for SED using ASR transcripts instead of true transcripts as presented in Table 3. It is interesting to observe that the pause duration feature can still help recover some of the information, as it is derived from the audio data and does not get influenced with errors in text. Using a better ASR model helps improve the system performance. However, we believe that this degradation in performance is due to the mismatch in training the SED models on true transcripts and using them to perform SED on ASR transcripts of the test data. In order to reduce this mismatch we propose to train SED models on ASR transcripts and use them to perform SED. The results with the proposed approach are presented in the next section.
Table 4: Performance of SED\textsubscript{asr} on E-corner test set using textual features (Case 2).

<table>
<thead>
<tr>
<th>ASR</th>
<th>% R</th>
<th>% P</th>
<th>% F1</th>
<th>% Acc</th>
<th>% BER</th>
<th>% SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR\textsubscript{1}</td>
<td>22.6</td>
<td>37.0</td>
<td>28.1</td>
<td>94.0</td>
<td>6.0</td>
<td>116.0</td>
</tr>
<tr>
<td>ASR\textsubscript{2}</td>
<td>26.3</td>
<td>42.3</td>
<td>32.4</td>
<td>94.3</td>
<td>5.6</td>
<td>109.7</td>
</tr>
<tr>
<td>ASR\textsubscript{3}</td>
<td>27.6</td>
<td>45.7</td>
<td>34.4</td>
<td>94.5</td>
<td>5.5</td>
<td>105.0</td>
</tr>
</tbody>
</table>

Table 5: Performance of SED\textsubscript{asr} on E-corner test set using textual and pause duration features (Case 2).

<table>
<thead>
<tr>
<th>ASR</th>
<th>% R</th>
<th>% P</th>
<th>% F1</th>
<th>% Acc</th>
<th>% BER</th>
<th>% SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASR\textsubscript{1}</td>
<td>17.0</td>
<td>62.0</td>
<td>26.5</td>
<td>95.2</td>
<td>4.8</td>
<td>93.4</td>
</tr>
<tr>
<td>ASR\textsubscript{2}</td>
<td>14.6</td>
<td>68.7</td>
<td>24.1</td>
<td>95.2</td>
<td>4.8</td>
<td>92.0</td>
</tr>
<tr>
<td>ASR\textsubscript{3}</td>
<td>14.9</td>
<td>69.4</td>
<td>24.6</td>
<td>95.6</td>
<td>4.8</td>
<td>91.7</td>
</tr>
</tbody>
</table>

4.4. Case 3: SED\textsubscript{asr} model for performing SED on ASR transcripts of the test set.

This section presents the results using the proposed approach to perform SED on ASR transcripts using SED models trained on ASR transcripts.

Table 6 presents the results using only textual features and Table 7 presents results using textual, pause duration and pitch features. Here we present results only using ASR transcripts from ASR\textsubscript{asr} which provides the best %WER. Comparing the results with Table 4 and Table 5, one can observe that the performance of SED\textsubscript{asr} has significantly improved by using models trained on ASR transcripts. We observe a relative gain of 18% using 4-gram textual features and 9.3% when adding the pause duration features respectively. Adding the pitch features to the pause duration and 4-gram textual features further improves the results by a relative gain of 2.4%.

Another interesting aspect to observe is the influence on SED performance with the increase in n-gram with the presence of ASR errors. Though, increasing the n-gram context helps improve the performance when reference transcripts are used, they seem to have a minimal influence when used with ASR transcripts. This might happen, as ASR transcripts do not preserve the words sequence information due to the insertion, deletion and substitution errors and to a certain degree are correlated with %WER levels. Hence, it is expected that the increase in the n-gram will not have same effect in improving the SED results as in the reference case. Tables 6 and 7 confirm this hypothesis. For instance, compare to the results reported in [6], the transition from 2-gram to 3-gram text features adds about 8% absolute improvement to the SER measure, while this improvement dropped to 3.5% absolute when used for ASR transcripts.

These results corroborate our hypothesis that reducing the mismatch between training and evaluation conditions improves the system performance, and reduces the effect of the noise produced by the ASR system. The relation between the level of ASR measure (WER) and the SED measures (F1 and SER), was studied using ASR\textsubscript{r}, and ASR\textsubscript{asr}, for both mismatched (case 2) and matched (case 3) scenarios. Table 8 presents this study, using 4-gram textual features and pause duration features, for both scenarios. The WER is highly correlated with (SER, in the positive direction), and with (F1, in the negative direction), in both scenarios, with an intra-class correlation coefficients of 0.89 and 0.90, respectively. Table 8 shows that the effect of the error-full ASR transcripts is less in the matched case than the mismatched case.

5. Conclusion

The paper showed that the performance of SED degrades using models trained on true transcripts for detecting sentence boundaries on ASR transcripts. This degradation is correlated with WER on the ASR transcripts. We hypothesised that using the models trained on true transcripts might provide a mismatch for detecting sentence boundaries on ASR transcripts and proposed to train models on ASR transcripts. By doing so, the models could learn sentence boundaries on error full ASR outputs and might be robust to ASR errors in test data. Experimental results presented on the E-corner lecture data showed that the proposed approach indeed improved the performance of SED. Since there is a wide gap in performance of SED using reference and ASR transcripts, there is still large scope for improving the system performance.

6. Acknowledgements

This work was partially supported by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology).

7. Data Access Statement

The speech data used in this paper was obtained from the following sources: ICSI Meetings corpus (LDC number LDC2004S02), AMI corpus (DOI number 10.1007/11677482_3), TedTalks data (data freely available as part of the IWSLT evaluations), E-corner data was harvested from http://ecorner.stanford.edu/index.html).

The specific file lists used for training and testing in the experiments in this paper, as well as result files can be downloaded from http://mini.dcs.shef.ac.uk/publications/papers/is15-hasan.
8. References


