Available online at www.sciencedirect.com

ScienceDirect

Computer Speech & Language 55 (2019) 187−199

www.elsevier.com/locate/csl

System-independent ASR error detection and classification using Recurrent Neural Network

Rahhal Errattahi[a], Asmaa EL Hannani*[,a], Thomas Hain[b], Hassan Ouahmane[a]

[a] *Laboratory of Information Technologies, National School of Applied Sciences, University of Chouaib Doukkali, Morocco*
[b] *Speech and Hearing Group, Department of Computer Science, University of Sheffield, UK*

Received 4 July 2018; received in revised form 16 November 2018; accepted 11 December 2018
Available online xxx

Abstract

This paper addresses errors in continuous Automatic Speech Recognition (ASR) in two stages: error detection and error type classification. Unlike the majority of research in this field, we propose to handle the recognition errors independently from the ASR decoder. We first establish an effective set of generic features derived exclusively from the recognizer output to compensate for the absence of ASR decoder information. Then, we apply a variant Recurrent Neural Network (V-RNN) based models for error detection and error type classification. Such model learn additional information to the recognized word classification using label dependency. As a result, experiments on Multi-Genre Broadcast Media corpus have shown that the proposed generic features setup leads to achieve competitive performances, compared to state of the art systems in both tasks. Furthermore, we have shown that V-RNN trained on the proposed feature set appear to be an effective classifier for the ASR error detection with an Accuracy of 85.43%.

Keywords: Automatic Speech Recognition; ASR error detection; ASR error type classification; Recurrent Neural Network

1. Introduction

Over the last few decades, Automatic Speech Recognition (ASR) has gained increasing interest among researchers and industry. This is due to the wide variety of applications in which ASR has been used, such for example, speech-to-text technologies (e.g., broadcast news transcription, video/TV programs subtitling, meeting transcriptions), speech based human machine interaction (e.g., Voice search engines, voice controlled intelligent assistants) and natural language understanding applications (e.g., voice question answering, speech-to-speech translation). The reason of the wide adoption of ASR in our daily life is the added value that users perceive when using their voice as an input, because the voice is a more natural way for people to communicate and is often faster than typing and in some cases (hands are busy, over the phone, in the dark or we are moving around) is the only available mode of interaction (Kiseleva et al., 2016).

* Corresponding author.
E-mail address: elhannani.a@ucd.ac.ma (A. EL Hannani).

https://doi.org/10.1016/j.csl.2018.12.007
0885-2308/ 2018 Elsevier Ltd. All rights reserved.

However, despite the impressive progress made in the field of speech processing, ASR systems continue to make errors during transcription, especially when handling various phenomena, including acoustic conditions (e.g., noise, competing speakers, channel conditions), out of vocabulary words, and pronunciation variations. Error-prone ASR results usually impact performances of the post recognition applications, like information retrieval, speech to speech translation, spoken language understanding, etc. Thus, ASR systems need a suitable approach for localizing ASR errors that may affect the post recognition application performance. Confidence scores proposed by ASR literature, when available, might be helpful to indicate if a word is correct or not by setting up a confidence threshold. Nevertheless, it only indicates how confident the ASR system is concerning its own output and does not imply the word correctness.

ASR error detection, also known as confidence estimation, aims to improve the exploitation of ASR outputs by highlighting the erroneous words in the recognizer output. The ASR error detection can also be passed downstream to an error type classification component that can classify the erroneous word as: Substitution or Insertion. Where, Substitution (S) refers to the error where a word in the reference is transcribed as a different word. While, Insertion (I) refers to the error where a word, that has no correspondence in the reference, appears in the automatic transcription.

The most widely studied approach in ASR error detection field is features-based, requiring the extraction of features from the ASR system or its output. The identification of effective features for ASR error detection is an open problem and several approaches have been proposed in the literature (Zhang and Rudnicky, 2001; Gibson and Hain, 2012; Ogawa and Hori, 2017). The proposed features, in the literature, may be divided into two main categories based on their sources: decoder-based features and non-decoder-based features. The decoder-based features are heavily dependent on the recognizer's implementation since they are based on intermediate information generated by the recognizer's decoder such as word lattices, Word Confusion Networks (WCN), n-best lists, and the recognizer's acoustic models. This type of features has the disadvantage of not being always accessible, and particularly when the ASR system is used as a black-box and the user does not have access to the internal features of the decoder. The non-decoder based features are however derived from external sources such as the recognizer output (i.e., transcription text), language models, part of speech tags, and syntactic and semantic features.

When information about the inner workings of the decoder used for transcription is accessible, current ASR error detection methods can supply post ASR applications with reliable indicators about output word correctness. This condition, however, does not always hold in the above scenarios. A clear motivating example is provided by the exponential growth of black-box speech recognition services, as Google voice Search and automatic captions in Youtube videos, where no information is available about the system used to produce the transcriptions. In this paper, we extends our previous works (Errattahi et al., 2016; 2018) on ASR error detection to a new and different scenario where information about the inner workings of the ASR system is not accessible. Unlike most approaches reported in the literature, we propose to handle the speech recognition errors independently from the decoder's internal information using a set of features derived exclusively from the recognizer output and hence should be trainable for any ASR system.

To the best of our knowledge, this paper represents the first extensive investigation of an efficient and system-independent automatic error detection in ASR output. Along this direction, the main contributions of this paper can be summarized as follows: (i) build a generic and system independent feature set for ASR error detection and classification; (ii) apply a variant of Recurrent Neural Network as classifier in ASR error detection and classification tasks for the first time; (iii) perform feature analysis, isolating the contribution of each feature set; and (iv) perform experiments on a new domain, namely multi-genre broadcast data.

From an application perspective the proposed approach could be used to: (i) decide at run-time whether a given input signal has been properly recognized (e.g., dialogue application), (ii) assess if an automatic transcription is acceptable as it is (e.g., automatic video subtitling), (iii) select the best transcription among options from multiple ASR systems (e.g., ROVER-based hypothesis combination methods), or (iv) converge towards an automatic ASR error correction system.

The remainder of the paper is organized as follows. After an overview of related works in Section 2, we describe our proposed approach for ASR error detection and error type classification in Section 3. Our experimental settings is fully described in Section 4, including a detailed description of the training and testing data sets, and the ASR system used for the transcription. In Section 5 we present the evaluation of our system with a detailed discussion of the achieved results for error detection and error type classification, respectively. Finally, in Section 6 we give some concluding remarks and future directions of this work.

2. Related works

There is a plethora of research that addresses the issue of ASR errors. Various types of approaches have been proposed and can be broadly classified into three categories (Jiang, 2005). In the first set of approaches, the posterior probability of the word given the acoustic signal is regarded as the confidence measure using a critical decision threshold to distinguish between correct and erroneous words (Kemp and Schaaf, 1997; Wessel et al., 1998; 1999; 2001; Rueber, 1997). In the second set of approaches, the confidence measure problem of ASR is formulated as a statistical hypothesis testing problem where the task is to test the *null* hypothesis that a given word or a words sequence is correctly recognized and truly come from a segment of speech against the *alternative* hypothesis that such word or words sequence is wrongly recognized and is not from that speech segment (Sukkar and Lee, 1996; Rose et al., 1995; Rahim et al., 1997). In the third set of approaches, a classifier is built using features generated from different sources (i.e., decoder and non-decoder features) to distinguish correctly recognized words from incorrectly recognized words (Fayolle et al., 2010; Korenevsky et al., 2015; Huang et al., 2013; Tam et al., 2014; Gibson and Hain, 2012; Zhang and Rudnicky, 2001; Ogawa and Hori, 2015; 2017). Usually the features-based approaches outperform the two other approaches, thanks to the effect of combining various sources of information.

In this paper, we mainly focus on features-based ASR error detection approaches. Much research has been done to identify the effective features and the suitable classifier for word correctness prediction in the output transcription of ASR systems. In general, features-based works could be categorized into two sub-categories: error detection and error type classification. *Error detection*, also referred to as *confidence estimation*, is the most popular ASR error detection task, in which each recognized word in the automatic transcription is labeled either as correct if it is correctly recognized, or as error if not (Fayolle et al., 2010; Korenevsky et al., 2015; Seigel and Woodland, 2011; Ogawa and Hori, 2015; 2017; Huang et al., 2013; Tam et al., 2014; Gibson and Hain, 2012; Zhang and Rudnicky, 2001). *Error type classification* rely on specifying the type of the ASR error (Seigel and Woodland, 2014; Ogawa et al., 2012; Ogawa and Hori, 2015; 2017). In other words each recognized word is labeled as Correct, substitution, insertion or Deletion. The majority of earlier works in this field consider only substitution and insertion errors, giving the difficulty to identify deletion errors. However, recently some researchers start working on deletion error detection, e.g., (Seigel and Woodland, 2014; Ogawa and Hori, 2017).

Table 1 summarizes the related research in both areas. It can be seen clearly from this table that most of features used in the reported works are derived from the decoding process (e.g., acoustic features, lattice features, and confusion network based features) and that most effort is targeted towards the detection task. Therefore, the major contribution in ASR error detection and classification performance comes from recognizer dependent features which makes those approaches strictly related to the components of the ASR system used during the training process and hence can’t be generalized to other systems. Particularly, when the recognition systems are used as a black-box and the user does not have access to the internals of the decoder. In addition, most of these features are redundant with the information used in generating output by the speech recognition system in the first place and so contribute little new information. In the other hand, few non-decoder features have been investigated in the literature and most of the examined features are domain specific. For example, in Chen et al. (2013) the proposed method is based on the statistical machine translation features, which is in most cases built on a domain specific data. But, despite the use of such features, the achieved results may appear very modest with a classification accuracy of 69.1%. So, even by today, an effective recognizer-independent detection of ASR errors remains to be explored.

Therefore, we believe that a promising approach lies in combining the most commonly used and easily accessible decoder features that include acoustic information, such as posterior probability, with a non-decoder source of complementary information. Linguistic analysis which is represented by Language Models (LM) is one such source. For this purpose, we propose to use a set of generic features derived exclusively from the recognizer output. The feature space is based on two information sources, the recognizer confidence score and an out-of-domain LM. These features, which are totally independent from the decoding process, have wider applicability to the recognizer-independent ASR error detection and classification tasks that represent our target scenario.

3. Proposed approach

We approach both ASR error detection and ASR error type classification as a supervised learning problem. Given a training set of (transcription, labels) aligned utterances, the task is to predict the label of each word in a test set of

Table 1
Summary of the related research in the field of ASR error detection and classification.

Approach	Features	Classifier	Data	% Accuracy
Zhang and Rudnicky (2001)	**Decoder features:** 3 Acoustic features 3 Lattice features 2 N-Best features **Non-decoder features:** 1 LM feature 2 Parser-based Features	SVMs	CMU Communicator system (Rudnicky et al., 1999) Training: 1000 utterances Testing: 781 utterances	Detection: 81.8
Pellegrini and Trancoso (2010)	**Decoder features:** 1 Acoustic feature 1 Posterior feature 4 Lattice features **Non-decoder features:** 3 Parser-based Features	ANN-HMMs	ALERT European Portuguese BN corpus (Amaral et al., 2007) Training: 108k words Testing: 16.5k words	Detection: 87.84
Ogawa et al. (2012)	**Decoder features:** 3 WCN features 3 Acoustic features **Non-decoder features:** 3 Lexical features	CRF	MIT lecture speech (Chang and Glass, 2009) Training: 1.92M words Testing: 72K words	Detection: 84.2 Classification: 65.22
Chen et al. (2013)	**Decoder features:** 3 WCN features 2 Acoustic features **Non-decoder features:** 2 Statistical Machine Translation features 3 Parser-based Features	CRF	BBN Byblos ASR system (Nguyen and Schwartz, 1997) Training: 1.5M words Testing: 4.6K words	Detection: 69.1
Korenevsky et al. (2015)	**Decoder features:** 4 WCN features 2 Lattice features **Non-decoder features:** 1 LM feature 3 Lexical features	RNNs	Spontaneous Speech (Levin et al., 2014) Training: 85 hours Testing: 15 hours	Detection: Reporting results using Precision-Recall graph
Ogawa and Hori (2015); Ogawa and Hori (2017)	**Decoder features:** 8 WCN features 4 Acoustic features **Non-decoder features:** 2 LM features 3 Lexical features	DBRNNs	MIT lecture speech (Chang and Glass, 2009) Training: 2M words Testing: 72K words	Detection: 85.52 Classification: 83.33

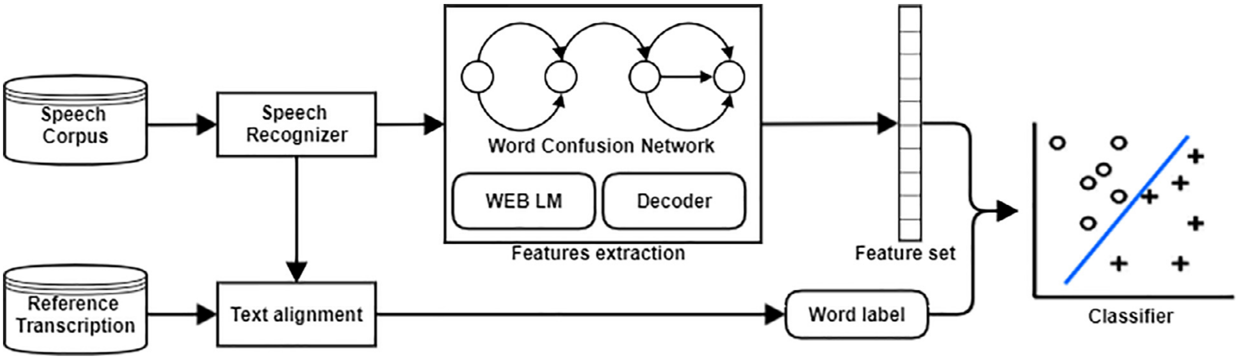


Fig. 1. General workflow of the proposed approach.

unseen (transcription) utterances using a variant Recurrent Neural Network. The workflow that we propose for both tasks is illustrated in Fig. 1.

3.1. Word alignment labels

To get the training labels in continuous speech recognition, we align the recognition output of the training set with its reference transcription using the NIST SCLITE[1] scoring package. Table 2 shows examples of such word alignment results. For the error detection task, a word label takes binary value that indicates if the recognized word is correct (C) or incorrect (E), while for the error type classification task, a recognized word is classified into one of three categories, i.e., correct (C), substitution error (S) or insertion error (I). In this work we don’t take into account the deletion errors.

3.2. Features

The identification of recognition errors in continuous speech recognition is accomplished by analysing each word within its context based on a set of features. As we aim to develop a generic model for ASR error detection, we did a

[1] NIST SCLITE Scoring Package Version 1.5, http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm.

Table 2
An example of word sequence alignment result between a recognized utterance and its corresponding reference transcription from the MGB corpus.

REFERENCE:	they	all	live	near	the	river	*****	GRANTA	in	Cambridge
HYPOTHESIS:	they	all	live	near	the	river	GRANT	ARE	in	Cambridge
Error detection label:	C	C	C	C	C	C	E	E	C	C
Error type classification label:	C	C	C	C	C	C	I	S	C	C

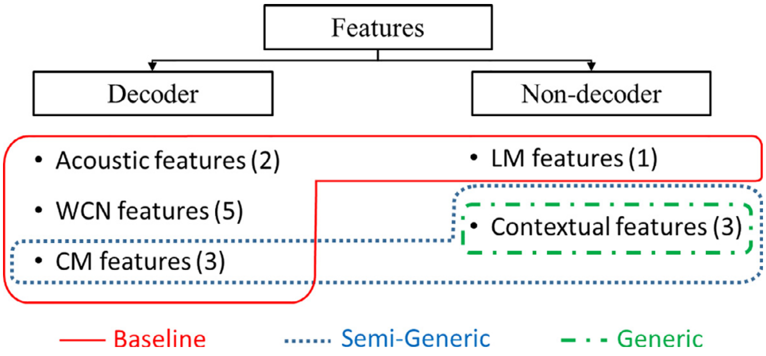


Fig. 2. Features categorisation.

categorization of all features that will be investigated in this work in three sets: **baseline, semi-generic** and **generic**. As illustrated in Fig. 2, the features categorization is performed depending on the nature and the source of the features. We split features into two main categories based on their sources: decoder based features and non-decoder based features. For the decoder features, they represent all features that are based on the ASR decoder or on the internal components of the decoder. The non-decoder features may include any features extracted from external information sources: such as LMs, semantic parsers, etc.

The baseline feature set consists of 11 features that have been consistently reported in the literature to be effective in ASR error detection (see Table 3). Where the first three features (i.e., CM features) are derived directly from the ASR output: posterior probabilities. Given that the majority of ASR systems generate, in addition to the output words, a score (probabilistic value between 0 and 1) called confidence score to indicate the trustworthiness of any word generated by the ASR decoder. In addition, two features are extracted: one from the ASR internal acoustic models and an extra feature that represent the word length in millisecond. The third type of baseline features are derived mainly from a Word Confusion Network (WCN). The WCN, example in Fig. 3, is build from a word lattices using the (Mangu et al., 2000) algorithm where each edge is labeled with a word hypothesis and its posterior probability and the ϵ arc corresponds to word deletion (null arc). We generated 5 features from the WCN using the SRILM toolkit[2]: The first feature represents the number of alternative arcs at each segment. The second represents the posterior probability of the ϵ arc. The third is calculated as the sum of the posterior probabilities of the alternative arcs in the current segment without the ϵ. And finally, the two last features are binary features that take value 1 if the posterior probability of ϵ arc is the highest in the adjacent segment, and 0 otherwise. The last type of baseline features is a non-decoder feature which represents an in-domain LM score.

In addition to the baseline features; we introduce three non-decoder features as in Errattahi et al. (2016), which are system-independent. These features, called contextual features are based on LM probability calculated from an out-of-domain N-gram dataset. In contrast to previous work where authors use a limited in-domain N-gram dataset, which is generally the same used in the ASR system, we propose to use a very generic N-gram dataset in order to make our solution flexible with any ASR System. Contrary to the baseline LM feature which is based on in-domain data. The N-gram LM probability is commonly used in ASR systems. However, in actual ASR systems, N-gram language models consider only the left side context, i.e., the probability of a given word with its preceding context. Including both sides of the context (i.e., left and right) could provide additional information about the correctness of

[2] http://www.speech.sri.com/projects/srilm.

Table 3
Baseline features used for ASR error detection and error type classification.

Feature	Description
	CM features
1	Log posterior probability of current word
2	Log posterior probability of previous word
3	Log posterior probability of next word
	Acoustic features
4	Word acoustic log-likelihood
5	Word duration
	WCN features
6	Number of alternatives
7	Insertion log probability (Total probability of ϵ arcs)
8	Substitution log probability (Total probability of alternatives)
9	Is the previous word equal to a null symbol corresponding to ϵ?
10	Is the next word equal to a null symbol corresponding to ϵ?
	LM Features
11	Word LM log-score

a given word in its context. Therefore, we propose two N-gram LM based features, where the first one is the probability of the word given its left context, and the second one is the probability of the word given its right context. In other words, giving a sequence S of N words $w_1, \ldots, w_i, \ldots, w_N$, left and right LM probability of w_i, are calculated using the Eqs. (1) and (2) respectively:

$$LeftLM = P(w_{i-n+1}, \ldots, w_{i-1}, w_i) \tag{1}$$

$$RightLM = P(w_i, \ldots, w_{i+n-2}, w_{i+n-1}) \tag{2}$$

Where in both equations n represents the context length ($n-1$ words in the left and $n-1$ words in the right), which correspond also to the N-gram order. When using a context window of two words ($n = 2$), the approximation of Eqs. 1 and 2 using the chain rule of probability and a bigram LM model, we get:

$$LeftLM = P(w_{i-1}, w_i) \approx P(w_i | w_{i-1}) \tag{3}$$

$$RightLM = P(w_i, w_{i+1}) \approx P(w_{i+1} | w_i) \tag{4}$$

where *LeftLM* is the standard forward bigram LM, and *RightLM* represents the backward bigram LM.

The third contextual feature is called the sentence oddity (Fong et al., 2006), which was first introduced for the problem of detecting substituted words in intercepted communication, where words that might raise attention are replaced by other innocent words that are in general not meaningful in the context of the sentence. In this work, we adopt the definition of sentence oddity for the problem of ASR error detection, starting from the following proposition: when a substitution or insertion has occurred, the joint probability of the entire remainder of the sentence, without the substituted or the inserted word, might be expected to be high. While, the joint probability of the sentence

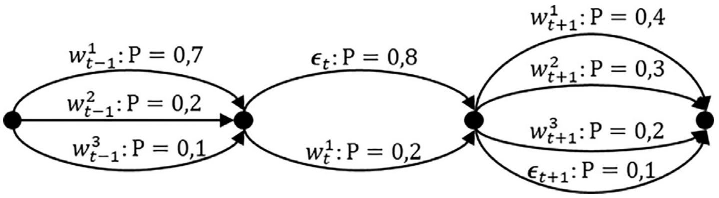


Fig. 3. Sample Word Confusion Network.

containing the substituted or the inserted word might be expected to be much lower, since the erroneous (i.e., substituted or inserted) word is unusual in the context of the sentence. Thus, instead of calculating the frequency of the bag-of-words as in Fong et al. (2006), we propose to use the joint probability of the words sequence. So, for a given sequence S of N words $w_1, \dots, w_i, \dots, w_N$, we redefine the Sentence Oddity (SO) as:

$$SO = \frac{P(w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_n)}{P(w_1, \dots, w_i, \dots, w_n)} \tag{5}$$

where, $P(w_1, \dots, w_{i-1}, w_{i+1}, \dots, w_n)$ denotes the joint probability of the words sequence without the word w_i, and $P(w_1, \dots, w_i, \dots, w_n)$ denotes the joint probability of the whole words sequence. The larger SO measure is, the more likely it is that the word is erroneous.

The joint probability of a given word sequence S, P(S), can be approximated with Eq. (4):

$$\begin{aligned} P(S) &= P(w_1, \dots, w_N) \\ &= \prod_{j=1}^{N} P(w_j | w_{j-n+1}, \dots, w_{j-1}) \end{aligned} \tag{6}$$

where, n denotes always the N-gram order.

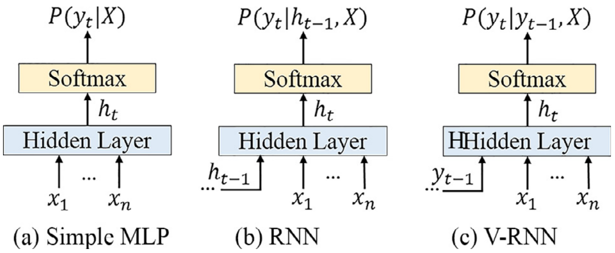
We denote by **semi-generic** features any features that could be easily extracted from the ASR outputs (e.g confidence measures) or from external sources. So, the semi-generic features set includes contextual features as well as the confidence scores features. The reasons behind considering CM features as semi-generic are: (i) most speech systems today provide the CS measure to inform users what can be trusted and what cannot; (ii) the value of the confidence score is thus one of the critical factors in determining success or failure of the speech decoder. While the **generic** features, include only features that are totally independent to the ASR decoder. The advantage of using such features is building an ASR error detection and classification system that could be easily trained for any ASR decoder that provides the hypothesis words as well as their confidence scores.

3.3. *Classifier*

ASR errors often are not single events (Errattahi et al., 2018). This is because a miss-recognized word generates often a sequence of ASR errors. Starting from this fact, we propose to use a Variant of Recurrent Neural Network (V-RNN) Dinarelli and Tellier as the classifier for ASR error detection for the first time. The proposed system is based on a recurrent learning strategy overs the outputs labels to train the network, as illustrated in Fig. 4c. This variant model perform recurrent connection between the output and the input layers, unlike simple RNN, see Fig. 4b, where the recurrent connection is only in the hidden layer. Intuitively, the network receives as input the previous word label prediction, $y_{t-1} \subset Y$ and the input vector $X = x_1, x_n$ and then computes an output vector h_t, which is dependent on the entire sequence of labels y_0, y_{t-1}, as follows:

$$h_t = relu(Az_t + b) \tag{7}$$

where z_t is the joint vector of model inputs $z_t = y_{t-1}, x_1, x_n$, A and b are parameters of the model. h_t are then passed to a softmax layer which defines a probability distribution over the set of output labels $P(y|y_{t-1}, X)$. Unlike in Multi-



(a) Simple MLP (b) RNN (c) V-RNN

Fig. 4. Neuronal based models for ASR error detection and classification.

Layer Perceptron (MLP) see Fig. 4a, which estimates the conditional probability of the labels at each time step using only the input vector $P(y|X)$.

The model can be optimized using a Stochastic Gradient-based technique named Adam Kingma and Ba. This method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. We note here that for the hidden layer we used Rectified Linear Units instead of sigmoids activation function. The experimental setup used to train and evaluate the classifiers as well as the configuration parameters are given in the following section.

4. Experimental settings

4.1. Multi-Genre Broadcast challenge data

The experiments in this paper make use of the data provided by the British Broadcasting Corporation (BBC) for the Multi-Genre Broadcast (MGB) challenge 2015 (Bell et al., 2015). Task 1 of the challenge involved participants having to perform the automatic transcription of a set of BBC shows. These shows were chosen to cover the multiple genres in broadcast TV, categorized in terms of 8 genres: advice, children's, comedy, competition, documentary, drama, events and news. The development data was used as the evaluation set in line with previous work [38, 39, 40]. This data set consisted of 47 shows that were broadcast by the BBC during a week in mid-May 2008.

The MGB development set were first transcribed using the ASR system described in Section 4.2, giving a word error rate of 30.1% as in Deena et al. (2016, 2017), and the resulting transcription was aligned with the reference transcription in order to get target labels for training our models. For ASR error detection and classification experiments, the MGB development set was split into 70% for training, 30% for test (after shuffling the utterances). The distribution of words in the training and test sets is summarized in Table 4.

4.2. ASR system

The setup for ASR experiments is the same as in Saz et al. (2015) and Deena et al. (2016, 2017) with 2-gram and 4-gram language models built on $LM1 + LM2$ text by first selecting a vocabulary of 200k words from all the words in the $LM2$ text (87k) and augmented with the most frequently occuring words in $LM1$. The acoustic models consisted of *Bottleneck* DNN-GMM-HMM. The *Bottleneck* system used a DNN for extracting 26 features. The DNN took as input 15 contiguous log-filterbank frames and consisted of 4 hidden layers of 1745 neurons plus the 26-neuron *Bottleneck* layer, and an output layer of 8000 triphone state targets. State-level Minimum Bayes Risk (sMBR) was used as the target function for training the DNN. Feature vectors for training the GMM-HMM systems were 65-dimensional, including the 26 dimensional *Bottleneck* features, as well as 13 dimensional PLP features together with their first and second derivatives. GMM-HMM models were trained using 16 Gaussian components per state, and around 8k distinct triphone states.

Decoding was performed using the 2-gram LM and rescored using the 4-gram LM, to produce lattices, which were then used to compute word confusion networks and subsequently WCN-derived features.

4.3. Out-of-domain N-gram dataset

Concerning the contextual features (i.e., LeftLM, RightLM, SO) extraction, we used the smoothed back-off Microsoft Web N-gram corpus (Wang et al., 2010). This corpus provides an open-vocabulary, smoothed back-off N-

Table 4
Words label distribution in the training and test sets.

Word label	Training set	Test set
Correct	86339	37158
Substitution	20406	8583
Insertion	2981	1260

gram Models and is dynamically updated as web documents are crawled. Since being composed of a huge volume of data crawled from web pages and documents of different domains, the Microsoft Web N-gram corpus provides a wide-ranging vocabulary (e.g., 1.2B 1-gram, 11.7B 2-gram) that can cover most of the English vocabulary in all domains, which justify our choice. In this work and for computational reason we used a context frame of two words (bigram) for contextual features (i.e. LeftLM, RightLM, SO).

4.4. ASR error detection and classification models

As a first set of experiments, we compared the proposed V-RNN with four other classifiers: MLP, Bayesian Network (BN), Support Vector Machines (SVM), and two Long Short-Term Memory (LSTM) based RNNs namely unidirectional LSTM (ULSTM) and bidirectional LSTM (BSLTM). Both, V-RNN and MLP models consist of a single layer of 2048 units with a *relu* (Nair and Hinton, 2010) activation function as described in 3.3. The SVMs were trained using non-linear radial basis function, and the parameters were optimized on the training data using a grid search procedure in the range between 10^{-9} and 10^{3}. In all our experiments the optimal value of γ was 10^{-7}. The BN classifier uses the simple estimator function to estimate the conditional probability and the k2 algorithm to heuristically search for the most probable beliefnetwork structure. The ULSTM consists of 1 hidden layer of 2048 staked LSTM units. While the BSLTM one has a bidirectional structure with two hidden layers, one forward and one backward, each of 2048 LSTM units.

The classifiers were trained for each task, error detection and error type classification, using the pairs of features and labels described above. In error detection, we have only two possible classes. A recognized word will take the label correct if it is well recognized and the label error if it is miss-recognized. In error type classification task, in addition to the correct label the classifier will be trained to distinguish between a substitution and insertion errors.

4.5. Performance metrics

To measure the performance of our models, we used two popular classification evaluation metrics: Accuracy and F-measure, which are calculated as follows:

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{8}$$

$$F-measure = \frac{2 * tp}{2 * tp + fp + fn} \tag{9}$$

where, *fn, fp, tn* and *tp* denote the false negative, false positive, true negative and true positive, respectively.

Another metric that is commonly used to measure binary classification is the DET curve, which plot the miss probability $P(miss)$ vs. false alarm rate FA. The miss probability measures the rate of missing the detection of an erroneous word. The false alarm rate measures the rate of incorrectly detecting a word error. $P(miss)$ and FA are given by:

$$P(miss) = \frac{fn}{tp + fn} \tag{10}$$

$$FA = \frac{fp}{N} \tag{11}$$

where, N is the total number of samples in the data.

5. Results

5.1. Basic experiments

As a first experiment we compared the performance of the proposed V-RNN model with that of the five other models; BN, SVM, MLP, ULSTM and BLSTM. The comparison was performed on error detection task using the

Table 5
Classification accuracies [%] and F-scores [%] of error detection obtained with BN, SVM, MLP, ULSTM, BLSTM and our proposed V-RNN for the test set using the baseline feature set.

Classifier	%Accuracy	%F-measure	
		correct	error
BN	81.47	88.74	47.60
MLP	79.70	86.67	57.51
SVM	83.34	90.11	47.05
ULSTM	83.46	90.14	48.71
BLSTM	83.47	91.66	47.31
V-RNN	85.06	90.93	57.69

baseline feature set (11 features). Table 5, shows the results (Classification accuracy and the F-score) of the error detection task using the six classifiers on the test set.

From the first three rows of the table, we can confirm that the SVM outperform both MLP and BN, especially for the correct words detection. Despite that the BN gives the best performance for the erroneous words detection with an F-measure around 47.6%; it remains less efficient than the SVM in term of classification accuracy with an accuracy of 81.47%, against 83.34% for the SVM.

Comparing the RNN based models (i.e., ULSTM, BLSTM and V-RNN) to the other stander machine learning models, namely BN, SVM and MLP, we can confirm the superiority of the RNN based models. This is because of the effect of training the RNN models on sequence data unlike other models, which are trained on the words level independently to their context. We can also observe that the BLSTM performs better than the unidirectional ULSTM. This improvement is due to the fact that Bidirectional LSTM-RNN can handle longer bidirectional contexts of input feature vectors and can model highly nonlinear relationships between the input feature vectors and output labels.

It is apparent from this table that the V-RNN show better performance than other models. Comparing the F-scores for the infrequent labels obtained with the V-RNN, we can clearly confirm the importance of considering previous outputs labels to predict the current word.

Fig. 5 shows the DET plot, for the 4 classifiers: BN, SVM, MLP and V-RNN. The plot shows significant gain when using V-RNN, and is consistent with the results in Table 5. At 10% false alarm rate, we achieved about 15% absolute reduction in Miss probability by using V-RNN based model compared to the BN based model, and about 6% compared to both SVM and MLP.

We can confirm that, as it was expected, the V-RNN outperform the other classifiers in ASR error detection task. Thus, in the following experiments, we will only report the results using the V-RNN model. The next set of experiments is aimed at investigating the features described in 3.2 in order to determine which ones perform best, both in isolation and in combination.

5.2. Comparison of Generic Versus Semi-Generic and Baseline features

Table 6 displays the V-RNN error detection and error type classification performance achieved on the test set using different features combination. Four settings are compared, corresponding to different types of features used in the V-RNN training. In addition to Classification Accuracy and F-score of each type of label, an averaged F-score across all types of labels is also reported. This is because the frequencies of each type of label are highly unbalanced and looking at the F-score of each class is not informative.

The results show that when using the baseline features; the V-RNN model achieves 85.06% as classification accuracy in ASR error detection. But, when using only the generic features, the model achieves slightly lower results with a classification accuracy of 81.64%. Nevertheless, it can be considered as a satisfying result since to the best of our knowledge non of the reported works in the literature has produced similar results using only non-decoder features. These later are often used as a boosting factor for the performance of the ASR error detection systems and not as isolated features. On the other hand, using the semi-generic feature set represents a good alternative to the baseline features since it provides an absolute improvement of 0.28% in the classification accuracy. Also, by checking the F-

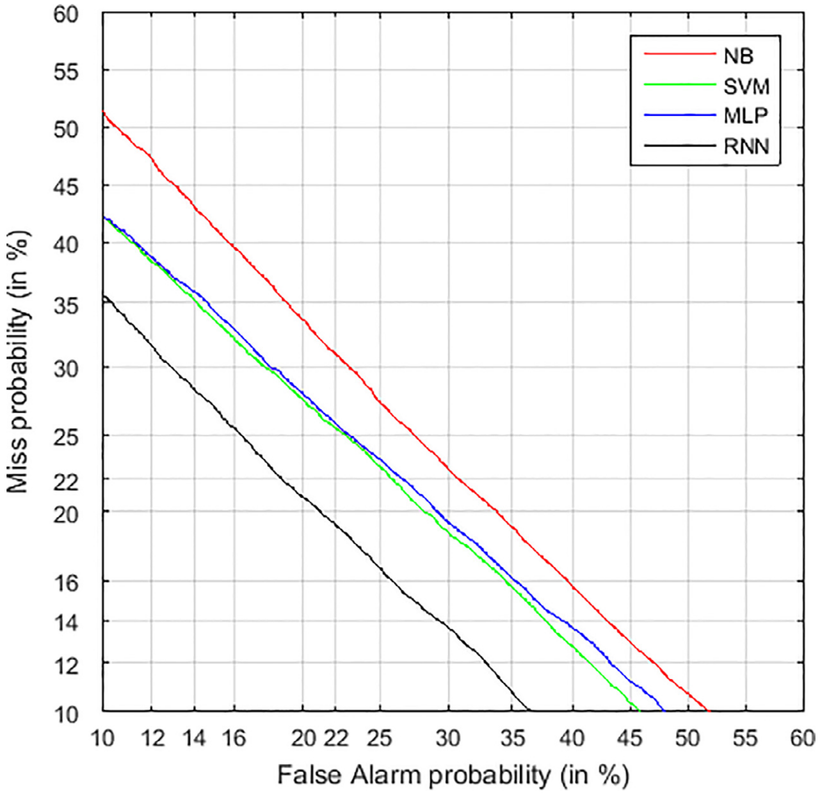


Fig. 5. DET Plot comparing classifiers detection performance: BN, SVM, MLP and V-RNN on the test set using baseline feature set.

scores, we can observe that a relatively significant improvement is obtained when using the semi-generic features as compared to the baseline features and the generic features alone. This improvement is especially relevant for the error labels where the F-score passed from 57.69% when using baseline features to 59.90% when using semi-generic features. For error type classification, we note that we perform an oversampling of the training samples with Insertion labels to adjust the class distribution. Taking a look at the second row of Table 6, we can observe that the F-scores change in correlation with the labels frequencies. Therefore, given that the insertion errors are less frequent than the substitution errors, the F-score of the substitution is higher than the F-score of the insertion. We observe also that there are small differences between the F-scores for the frequent labels (correct) obtained with each of the

Table 6
V-RNN classification accuracies [%] and F-scores [%] of error detection (Correct/Error) and and error type classification(Correct/Substitution/Insertion) obtained based on different combination of features.

Features		Baseline	Generic	Semi-generic	All
			Error detection		
Classification accuracy		85.06	81.64	**85.34**	85.30
F-score:	Correct	90.93	89.19	91.03	**91.07**
	Error	57.69	39.24	**59.90**	58.41
	Avg	83.84	78.54	**84.39**	84.10
			Error type classification		
Classification accuracy		83.55	80.18	83.27	**84.00**
F-score:	Correct	90.55	89.21	91.05	**91.06**
	Substitution	58.70	30.70	51.89	**55.95**
	Insertion	07.60	19.64	**22.90**	14.48
	Avg	82.37	76.43	81.92	**82.45**

feature set. On the other hand, we observe a large differences between the F-scores of the less frequent labels (Insertion and Substitution) obtained with different feature set. It is clear that training V-RNN on semi-generic features gives close results in comparison to the baseline feature set. In contrast, the best error classification results was achieved when using the total feature set. However, when comparing the F-scores for both type of errors e.g., Substitution and Insertion, we can confirm the superiority of the semi-generic features in error type classification task. One reason for this may be the effect of using contextual features, the F-score of insertion labels when using the semi-generic features is 19.64% compared to only 07.60% when using the baseline features.

Moreover, even when using only the generic features our results are still very positive, matching many and improving some previous state-of-the-art systems with an accuracy of 81.06% (see Table 1). It is encouraging to compare our findings with the state-of-the-art, particularly those reported in (Ogawa and Hori, 2017) by Ogawa et al., as in the later work authors make use of a lecture speech corpora with similar complexity to the broadcast corpora used in our experiments.

6. Conclusions

We have presented a generic approach for automatic speech recognition error detection and error type classification. Where we propose to handle the speech errors independently from the ASR decoder using a set of features derived exclusively from the ASR output and hence should be usable with any ASR system without any further tuning. Experimental results showed that the proposed generic features achieve competitive performance in error detection as compared to the state-of-the-art approaches. We have also shown that the same setup could be applied to ASR error type classification task. More interestingly, we have confirmed that ASR errors are influenced by their context and that by incorporating the label of the previous word via V-RNN leads to achieve higher results in both tasks (i.e. error detection and error type classification).

This study provides proof of concept that generic feature can provide confirmatory evidence of the correctness of word in the output transcription of ASR systems, as well as lead to reveal the effectiveness of our variant RNN (V-RNN) in sequence tagging and particularly in ASR error detection. Future works will include adding supplementary generic features, such as lexical and semantic features, with further refinements of our training model using advanced deep learning techniques while providing additional training data from different tasks and using different ASR decoders. We also intend to consider the deletion error based on contextual and lexical features.

References

Amaral, R., Meinedo, H., Caseiro, D., Trancoso, I., Neto, J., et al., 2007. A prototype system for selective dissemination of broadcast news in European Portuguese. EURASIP J. Adv. Signal Process 2007 (1), 037507.

Bell, P., Gales, M., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Webster, M., Woodland, P., 2015. The MGB challenge: evaluating multi−genre broadcast media transcription. In: Proceedings of the ASRU.

Chang, H.A., Glass, J.R., 2009. Discriminative training of hierarchical acoustic models for large vocabulary continuous speech recognition. IEEE. Proceedings of the ICASSP, 4481−4484.

Chen, W., Ananthakrishnan, S., Kumar, R., Prasad, R., Natarajan, P., 2013. ASR error detection in a conversational spoken language translation system. IEEE. Proceedings of the ICASSP, 7418−7422.

Deena, S., Hasan, M., Doulaty, M., Saz, O., Hain, T., 2016. Combining feature and model-based adaptation of RNNLMs for multi-genre broadcast speech recognition. In: Proceedings of the INTERSPEECH, pp. 2343–2347.

Deena, S., Ng, R.W.M., Madhyashta, P., Specia, L., Hain, T., 2017. Semi-supervised adaptation of RNNLMs by fine-tuning with domain-specific auxiliary features. ISCA. Proceedings of the INTERSPEECH.

Dinarelli, M., Tellier, I., Improving recurrent neural networks for sequence labelling. CoRRabs/1606.02555. http://arxiv.org/abs/1606.02555, 2016.

Errattahi, R., Hannani, A.E., Hain, T., Ouahmane, H., 2018. Towards a generic approach for automatic speech recognition error detection and classification. In: Proceedings of the 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), pp. 1–6.

Errattahi, R., Hannani, A.E., Ouahmane, H., Hain, T., 2016. Automatic speech recognition errors detection using supervised learning techniques. In: Proceedings of the 13th IEEE/ACS International Conference of Computer Systems and Applications AICCSA'16, pp. 1–6. Agadir, Morocco.

Fayolle, J., Moreau, F., Raymond, C., Gravier, G., Gros, P., et al., 2010. CRF-based combination of contextual features to improve a posteriori word-level confidence measures. In: Proceedings of the Interspeech, pp. 1942–1945.

Fong, S., Skillicorn, D., Roussinov, D., 2006. Detecting word substitution in adversarial communication. In: Proceedings of the 6th SIAM Conference on Data Mining. Bethesda, Maryland.

Gibson, M., Hain, T., 2012. Application of SVM-based correctness predictions to unsupervised discriminative speaker adaptation. In: Proceedings of the ICASSP, pp. 4341–4344.

Huang, P.S., Kumar, K., Liu, C., Gong, Y., Deng, L., 2013. Predicting speech recognition confidence using deep learning with word identity and score features. IEEE. Proceedings of the ICASSP, 7413−7417.

Jiang, H., 2005. Confidence measures for speech recognition: a survey. Speech Commun. 45 (4), 455–470.

Kemp, T., Schaaf, T., et al., 1997. Estimating confidence using word lattices. In: Proceedings of the EuroSpeech, pp. 827–830.

Kingma, D.P., Ba, J., Adam: a method for stochastic optimization. CoRRabs/1412.6980. http://arxiv.org/abs/1412.6980, 2014.

Kiseleva, J., Williams, K., Jiang, J., Awadallah, A.H., Crook, A.C., Zitouni, I., Anastasakos, T., 2016. Understanding user satisfaction with intelligent assistants. ACM. Proceedings of the ACM on Conference on Human Information Interaction and Retrieval, 121−130.

Korenevsky, M.L., Smirnov, A.B., Mendelev, V.S., 2015. Prediction of speech recognition accuracy for utterance classification. In: Proceedings of the Interspeech, pp. 1275–1279.

Levin, K., Ponomareva, I., Bulusheva, A., Chernykh, G., Medennikov, I., Merkin, N., Prudnikov, A., Tomashenko, N., 2014. Automated closed captioning for russian live broadcasting. In: Proceedings of the Interspeech.

Mangu, L., Brill, E., Stolcke, A., 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. Comput. Speech Lang. 14 (4), 373–400.

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 807–814.

Nguyen, L., Schwartz, R.M., 1997. Efficient 2-pass N-best decoder. In: Proceedings of the EuroSpeech.

Ogawa, A., Hori, T., 2015. ASR error detection and recognition rate estimation using deep bidirectional recurrent neural networks. IEEE. Proceedings of the ICASSP, 4370−4374.

Ogawa, A., Hori, T., 2017. Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks. Speech Commun. 89, 70–83.

Ogawa, A., Hori, T., Nakamura, A., 2012. Error type classification and word accuracy estimation using alignment features from word confusion network. IEEE. Proceedings of the ICASSP, 4925−4928.

Pellegrini, T., Trancoso, I., 2010. Improving ASR error detection with non-decoder based features. In: Proceedings of the Interspeech.

Rahim, M.G., Lee, C.H., Juang, B.H., 1997. Discriminative utterance verification for connected digits recognition. IEEE Trans. Speech Audio Process. 5 (3), 266–277.

Rose, R.C., Juang, B.H., Lee, C.H., 1995. A training procedure for verifying string hypotheses in continuous speech recognition. IEEE. Proceedings of the ICASSP, 1, 281−284.

Rudnicky, A.I., Thayer, E.H., Constantinides, P.C., Tchou, C., Shern, R., Lenzo, K.A., Xu, W., Oh, A., 1999. Creating natural dialogs in the carnegie mellon communicator system. In: Proceedings of the Eurospeech.

Rueber, B., 1997. Obtaining confidence measures from sentence probabilities. In: Proceedings of the Eurospeech.

Saz, O., Doulaty, M., Deena, S., Milner, R., Ng, R.W.M., Hasan, M., Liu, Y., Hain, T., 2015. The 2015 Sheffield system for transcription of Multi-Genre broadcast media. In: Proceedings of the ASRU.

Seigel, M.S., Woodland, P.C., et al., 2011. Combining information sources for confidence estimation with CRF models. In: Proceedings of the Interspeech, pp. 905–908.

Seigel, M.S., Woodland, P.C., 2014. Detecting deletions in ASR output. IEEE. Proceedings of the ICASSP, 2302−2306,

Sukkar, R.A., Lee, C.H., 1996. Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition. IEEE Trans. Speech Audio Process. 4 (6), 420–429.

Tam, Y.C., Lei, Y., Zheng, J., Wang, W., 2014. ASR error detection using recurrent neural network language model and complementary ASR. IEEE. Proceedings of the ICASSP, 2312−2316,

Wang, K., Thrasher, C., Viegas, E., Li, X.L., Hsu, B.J.P., 2010. An Overview of Microsoft Web N-gram Corpus and Applications.

Wessel, F., Macherey, K., Ney, H., 1999. A comparison of word graph and N-best list based confidence measures. In: Proceedings of the EuroSpeech.

Wessel, F., Macherey, K., Schluter, R., 1998. Using word probabilities as confidence measures. IEEE. Proceedings of the ICASSP, 1, 225−228.

Wessel, F., Schluter, R., Macherey, K., Ney, H., 2001. Confidence measures for large vocabulary continuous speech recognition. IEEE Trans. Speech Audio Process. 9 (3), 288–298.

Zhang, R., Rudnicky, A.I., 2001. Word level confidence annotation using combinations of features. In: Proceedings of the EuroSpeech, pp. 2105–2108.