White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/170079/

Version: Published Version

Article:
El Hannani, A., Errattahi, R., Salmam, F.Z. et al. (2021) Evaluation of the effectiveness and efficiency of state-of-the-art features and models for automatic speech recognition error detection. Journal of Big Data, 8. 5. ISSN: 2196-1115

https://doi.org/10.1186/s40537-020-00391-w

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

Journal of Big Data

RESEARCH Open Access

Evaluation of the effectiveness and efficiency of state-of-the-art features and models for automatic speech recognition error detection

Asmaa El Hannani[1*], Rahhal Errattahi[1], Fatima Zahra Salmam[2], Thomas Hain[3] and Hassan Ouahmane[1]

*Correspondence:
elhannani.a@ucd.ac.ma
[1] Laboratory of Information Technologies, National School of Applied Sciences, University of Chouaib Doukkali, El Jadida, Morocco
Full list of author information is available at the end of the article

Abstract

Speech based human-machine interaction and natural language understanding applications have seen a rapid development and wide adoption over the last few decades. This has led to a proliferation of studies that investigate Error detection and classification in Automatic Speech Recognition (ASR) systems. However, different data sets and evaluation protocols are used, making direct comparisons of the proposed approaches (e.g. features and models) difficult. In this paper we perform an extensive evaluation of the effectiveness and efficiency of state-of-the-art approaches in a unified framework for both errors detection and errors type classification. We make three primary contributions throughout this paper: (1) we have compared our Variant Recurrent Neural Network (V-RNN) model with three other state-of-the-art neural based models, and have shown that the V-RNN model is the most effective classifier for ASR error detection in term of accuracy and speed, (2) we have compared four features' settings, corresponding to different categories of predictor features and have shown that the generic features are particularly suitable for real-time ASR error detection applications, and (3) we have looked at the post generalization ability of our error detection framework and performed a detailed post detection analysis in order to perceive the recognition errors that are difficult to detect.

Keywords: Automatic Speech Recognition, Confidence estimation, ASR error detection, ASR error type classification, Recurrent Neural Network, Multi-Genre Broadcast

Introduction

Large Vocabulary Continuous Speech Recognition (LVCSR), which is characterized by a high variability of the speech, is still a challenging task for Automatic Speech Recognition (ASR) technology developers. And even though many algorithms and technologies have been proposed by the scientific community for all steps of LVCSR over the last decade, including pre-processing, feature extraction, acoustic modeling, language modeling, and decoding, the problem of LVCSR is far from being solved as described in [1]. In some domains, like read continuous speech where generally the speech is recorded

Springer Open

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

under clean conditions, results are satisfying with an error rate under 5%. While in other domains that contain more speech variations, such as video speech or distant conversational speech (meeting), results are still not acceptable presenting an error rate near 50%. To deal with this key problem and to enhance the performance of imperfect ASR systems, the automatic detection and correction of the transcription errors can, in some cases, be the only choice. Particularly, when tuning the ASR system itself is not possible (e.g. the system is purchased as a black-box) or when the manual correction is not convenient or even impossible as in the case where the transcription is not the final goal of the system (e.g. machine translation, information retrieval and question answering systems).

In this context, ASR error detection and classification, also known as confidence estimation, has been largely addressed in the literature, and we refer the reader to [2, 3] for a detailed overview. The most widely studied approach is features-based, in which a classifier is built using features generated from different sources (i.e. decoder and non-decoder features) to distinguish the correctly from the incorrectly recognized words [4–6]. Neverthless most of features used in the reported works are derived from the decoding process ASR systems (e.g. acoustic features, lattice features, and confusion network based features). Therefore, the major contribution in ASR error detection and classification performance comes from recogniser 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. A clear motivating example is provided by the exponential growth of black-box speech recognition services, such as Google voice Search and automatic captions in Youtube videos, where no information is available about the system used to produce the transcriptions. In addition, the extraction of most of these features is time-consuming which makes them not suitable for real-time systems.

To tackle these problems, we have been developing a new approach for ASR error detection and error type classification [3, 7, 8]. We have targeted a new and different scenario where information about the inner workings of the ASR system is not accessible. Unlike the majority of research in this field, our work focuses on handling the recognition errors independently from the ASR decoder using a generic Framework. In [3] we proposed an effective set of features acquired exclusively from the recogniser output to compensate for the absence of ASR decoder information. The proposed features are derived from two types of Language Models (LM)s. The first set, called contextual features, are extracted using an out-of-domain n-gram LM. While the second set of features is derived using a standard and a reverse-word Recurrent Neural Network LM. Furthermore, we proposed a V-RNN model in order to incorporate additional information to the recognised word classification using label dependency. As a result, experiments on Multi-Genre Broadcast (MGB) Media corpus have shown that: (i) both contextual and Recurrent Neural Network Language Model (RNNLM) features have a positive impact on the model performance, whether isolated or combined, (ii) the RNNLM adaptation techniques boost the Framework performance through the introduction of auxiliary features about the domain, (iii) the proposed generic and semi-generic setups lead to achieve competitive performances compared to state-of-the-art systems in both tasks, and (iv) the new V-RNN appears to be an

effective classifier in sequence tagging and particularly in ASR error detection and ASR error type classification. Nevertheless, the computational complexity of the proposed approach have not yet been investigated.

As stated before, ASR error detection can also serve as input to downstream systems like machine translation, information retrieval, and question answering. Therefore at least real-time performance must be a requirement. Thus, the purpose of this paper is to evaluate and compare not only the effectiveness but also the efficiency of state-of-the-art approaches, including our V-RNN based and generic approach, in a unified framework for both errors detection and errors type classification. More precisely, we are interested in the Real Time Factor (RTF), which expresses the ratio of the ASR error detection system time to the speech duration. RTF is commonly used to decide how suitable an approach is for real-time applications, in which case the RTF needs to be smaller than one ($RTF < 1.0$). In other words, the error detection of an utterance should take less time than a user needed for pronouncing the utterance. A second objective of this work was to perform a detailed post detection analysis in order to perceive the recognition errors that are difficult to detect. The results of the evaluation and analysis may serve as a benchmark for future studies.

The rest of this paper is organized as follows. In "ASR error detection and classification system" section the different components of the ASR error detection and classification system are presented. "Experiments and data description" section describes the experimental setup and data set. Section "Results and discussion" contains the experimental results along with a discussion of the effectiveness and efficiency of the compared features and models. "Detailed analysis" section looks at the post generalization ability of the best ASR error detection system as well as the post detection analysis. Finally, "Conclusion" section summarises the paper and gives directions of future works.

ASR error detection and classification system

In this work, ASR error detection is considered as a pattern recognition problem, in both sub-tasks; error detection and error type classification. For the error detection task, a word label takes a binary value that indicates if the recognized word is correct or erroneous, while for the error type classification task, a recognized word is assigned into one of three classes, i.e. correct, substitution error or insertion error. In this work we do not take into account the deletion errors. To this effect, a classifier is trained to map input features, called predictor features, to class posterior probabilities. During training, audio recordings are processed by an ASR system that generates the most likely word sequence, called hypothesis. Then, each hypothesis is aligned with its reference transcription to get the correct (ground truth) label sequence. In parallel with text alignment, a set of predictor features are computed from the speech decoding, confusion network and Language Models for each word in the hypothesis. The predictor features and the label sequence are then passed to the classification component, which is trained to predict the label of each word in the hypothesis based on information encoded by the so-called predictor features. The test phase is similar to the training phase, with the exception that the trained classifier in the end has to predict labels of unseen words given only their predictor features.

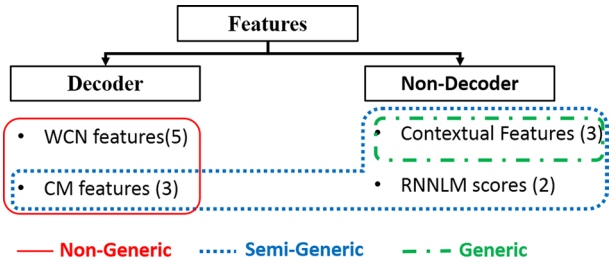

Fig. 1 Features categorisation

Predictor 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. In [3, 9], we have focused on collecting several features and analysing their effect on the ASR Error Detection performance. In this paper, we will consider only the features that we found to be the most effective:

- Three features based on confidence score (CS): (1) Log posterior probability of the recognized word,(2) Log posterior probability of the preceding (Left) word (w_{i-1}) and (3) Log posterior probability of the following (Right) word (w_{i+1});
- Five features derived from Word Confusion Network (WCN): (1) Number of alternatives, (2) Insertion log probability, (3) Substitution log probability, (4) A binary feature: is the previous word equal to a null symbol? and (5) is the next word equal to a null symbol?;
- Three contextual features: (1) Left LM probability, (2) Right LM probability and (3) Sentence oddity (SO);
- Two features based on RNNLM: (1) Forward RNNLM score and (2) Reverse-word RNNLM score.

As we aim to develop a generic model for ASR error detection, we propose furthermore to categorise the features on the basis of their dependency to the ASR system in three categories: non-generic, semi-generic and generic.

As illustrated in Fig. 1, 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, referred also in this work as non-generic features. The non-decoder features may include any features extracted from external information sources: such as LMs, semantic parsers, etc.

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 in addition to the RNNLM scores features. The reasons behind considering CS 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

Reference transcription	...	the	most		dissimilar	about	them	is	that	...
Recognition result	...	the	more	this	similar	about		is	that	...
Alignment result	...	C	S	I	S	C	D	C	C	...

Fig. 2 Words sequence alignment result between a recognized utterance and its corresponding reference transcription. The dotted rectangle indicates the segment that is influenced by the utterance of an Out-of-Vocabulary (OOV) word "dissimilar"

score is thus one of the critical factors in determining the success or failure of the speech decoder. While the generic features, include only features that are totally independent to the ASR decoder and from the ASR domain. What are the motivations behind the idea of building such a generic error detection Framework? The answers to this question are mainly based on the actual shortcomings of ASR error detection methods. First, such Framework could be easily trained on any ASR and any domain. The generic features are based on web crawled n-gram LMs and hence they could be used in the assessment of the output of any ASR system. Second, this is suitable when using black-box system. Most of the ASR technologies used in our daily life are provided as a black-box, thereby the user does not have access to the internals of the decoder. So, when using generic features we could train our assessment system directly on the text output.

Classifiers

As stated before, a specific classifier will be needed to learn the probability distribution functions for each word label in the training set. These functions play the role of a knowledge base to perform classification on the test set. Recently, Recurrent Neural Network (RNN) have been successfully used in several natural language processing tasks and achieved state-of-the-art performances for many sequence-tagging-tasks including handwriting recognition [10], speech recognition [11], machine translation [12], and also error detection in automatic speech recognition [6].

Recurrent networks (including LSTM and GRU based RNNs) are built to model the conditional distribution of label sequence, given the input sequence. Thereby RNN are only able to represent distributions in which the label values are conditionally independent from each other given the input values. ASR errors often are not single events [7]. This is because a miss-recognized word generates often a sequence of ASR errors, as illustrated in Fig. 2, where the Out-of-Vocabulary word "dissimilar" causes a sequence of recognition errors. So, the conditional independence assumption is not satisfied in this application. This fact motivated us to propose a new Variant of Recurrent Neural Network (V-RNN) as the classifier for ASR error detection for the first time in [8].

The proposed V-RNN is based on a recurrent learning strategy over the outputs labels to train the network, meaning that the previous word label is considered as input to the network in the next time step as illustrated in Fig. 3c. This variant model performs recurrent connection between the output and the input layers, unlike the standard RNN (Fig. 3b) where the recurrent connection is only in the hidden layer or the Multi Layer

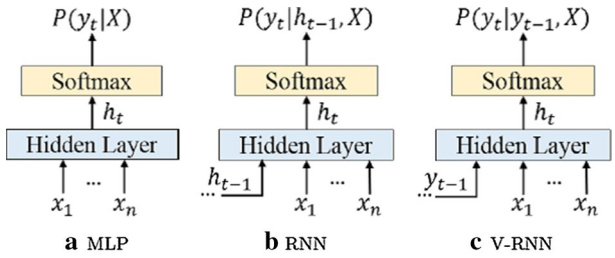

Fig. 3 V-RNN model versus the MLP and the standard RNN [3]

Table 1 The distribution of the data in terms of shows and broadcast time

Genre	Shows	Time (H)
Advice	4	3.0
Children's	8	3.0
Comedy	6	3.2
Competition	6	3.3
Documentary	9	6.8
Drama	4	2.7
Events	5	4.3
News	5	2.0
Total	47	28.3

Perceptron (MLP) (Fig. 3a) which estimates the conditional probability of the labels at each time step using only the input vector at the same time step.

In this paper we will extend our previous works and investigate more the effectiveness of the V-RNN classifier. In addition, we will compare it with three other classifiers: a MLP and two LSTM based RNNs namely Unidirectional Long Short-Term Memory (ULSTM) and Bidirectional Long Short-Term Memory (BLSTM). The experimental setup used to train and evaluate the four classifiers as well as the configuration parameters are given in the next section.

Experiments and data description

ASR system and data

The experiments in this paper are performed using the development data set provided by the British Broadcasting Corporation (BBC) for the MGB challenge 2015 [13]. This data consists of a set of BBC shows covering the multiple genres in broadcast TV, categorised in terms of 8 genres: advice, children's, comedy, competition, documentary, drama, events and news.

Table 1 shows the numbers of shows and the associated broadcast time for the data set we used to train and evaluate the proposed ASR error detection and classification systems across the 8 genres. The transcription of this data set is obtained using the ASR system described in [14, 15], giving a Word Error Rate (WER) of 30.1%. The resulting

Table 2 Words label distribution in the training and test sets

Word label	# in the training set	# in the test set
Correct	86339	37158
Substitution	20406	8583
Insertion	2981	1260

transcription was aligned with the reference transcription using the NIST SCLITE[1] scoring package in order to get target labels for training the prediction models. The data set was split into 70% for the training and 30% for the test (after shuffling the utterances). The distribution of words labels in the training and test sets is summarized in Table 2.

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.

Experimental setup

Regarding the contextual features (i.e. LeftLM, RightLM, SO) extraction, we used the smoothed back-off Microsoft Web n-gram corpus [16]. This corpus provides an open-vocabulary, smoothed back-off n-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 only used a context frame of two words (bigram) for the contextual features.

The experimented RNNLMs have 512 nodes in the hidden layer and where applicable, 512 nodes for the adaptation layer. The RNNLMs were trained using the CUED RNNLM toolkit [17] with a 60 K vocabulary for the input word list and a 50 K vocabulary for the output word list, both obtained by shortlisting the 200K vocabulary based on most frequent words.

It was shown in [14] that the LDA features, extracted from the ASR output, can be used as an auxiliary feature and input to the RNNLM hidden layer. Latent Dirichlet Allocation (LDA) is a generative probabilistic model that describes collections of text or other types of discrete data [18]. The idea is to consider texts as random mixtures on unobserved topics, where each topic is characterized by a distribution on words. In order to train LDA models, the term frequency-inverse document frequency (TF-IDF) vectors are extracted on the text. Then LDA features are derived by computing Dirichlet posteriors over the topics.

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

Table 3 Classification accuracies, F-measures and RTF of error detection obtained with the four models trained using the best feature set

Feature set	Accuracy (%)	F-measure (%)			RTF
		Correct	Error	Average	
MLP	85.17	90.82	61.42	84.66	0.29×10^{-4}
ULSTM	84.67	90.61	58.13	83.80	2.97×10^{-4}
BLSTM	85.19	90.80	61.93	84.75	7.94×10^{-4}
V-RNN	86.58	91.67	65.42	86.17	0.28×10^{-4}

In order to choose the LDA feature dimensionality, previous work has investigated different number of LDA dimensions [14], where authors extracted LDA features from the reference text for each show and varied the number of topics from 10 to 150 and then computed the RNNLM perplexity on the MGB development set. They found that 100 topics gives the best result and based on their founding the number of LDA topics in this work was fixed to 100.

Both, V-RNN and MLP models consist of a single layer of 2048 units with a *relu* [19] activation function as described in [8]. 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.

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

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{1}$$

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

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

Results and discussion

Assessment of the performance of the models

In this section we report the experimental results on the MGB data using the predictor features and models presented in previous sections. The performance of the proposed V-RNN model is compared with ULSTM and BLSTM based RNNs using the same experiment setup (same features set and same training data). To measure the performance of our models, we report the classification accuracy, the F-measure and the RTF. In addition, an averaged F-measure 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-measure of each class is not informative.

Table 4 Classification accuracies and F-measures of error type classification obtained with the four models trained using the best feature set

Feature set	Accuracy (%)	F-measure (%)			
		Correct	Substitution	Insertion	Average
MLP	83.97	90.84	57.84	01.70	82.41
ULSTM	83.39	90.59	49.90	01.69	80.76
BLSTM	83.41	90.59	52.13	03.14	81.21
V-RNN	85.10	91.58	57.81	04.10	83.05

When looking at the ASR error detection results in Table 3, we can first observe that the BLSTM performs better than both ULSTM and MLP when comparing their average f-measure and accuracy. This improvement is due to the fact that BLSTM can handle longer bidirectional contexts of input feature vectors and can model highly nonlinear relationships between the input feature vectors and output labels.

We can also observe clearly from these results that the V-RNN shows better performance than other models. The classification accuracy increases when using the V-RNN, with an absolute improvement of about 2% over the ULSTM, and 1.4% absolute improvement over the BLSTM. Also, by checking the F-measure, we can observe that a relatively significant improvement is obtained when using the V-RNN as compared to both LSTM (unidirectional and bidirectional) models. This improvement is especially relevant for the error labels where the F-measure passed from 58.13% when using ULSTM to 65.42% when using the proposed V-RNN. The superiority of the V-RNN to the ULSTM and BLSTM indicates that adding labels history to the input feature vector of the RNN is very effective in ASR error detection task and could be generalized to other tagging problem in natural language processing. When looking at the Real Time Factor of the 4 models we can observe that all models are suitable for real time ASR error detection with $RTF < 1.0$. Particularly the MLP and the V-RNN models had the best performance on RTF in the order of 10^{-5} as compared to ULSTM and BLSTM based models. In other words, the proposed V-RNN model is not only efficient for ASR error detection but also it is 10 times faster than the best stat-of-the-art based RNNs.

The same findings can be extended to the ASR error classification task as shown in Table 4. Where the V-RNN model achieves the best accuracy against other models with an accuracy of 85.10%. It is also important to note that for the first time the F-measure of the insertion class reaches 4% which is a good indicator of the model performance. However, this result is still below the expectations and as it was shown in Table 2 is due to the fact that insertion errors are infrequent in the training set which could be resolved by using data augmentation technique.

Therefore, we can confirm that, as it was expected, the V-RNN outperforms the other classifiers. This proves the utility of label dependency learning strategy in ASR error detection task. We believe also that combining advanced RNNs units such as LSTM or GRU, which are claimed the most effective RNNs, and label dependency strategy would give better results. We leave this as future work.

Table 5 V-RNN classification accuracies, F-measures and RTF of error detection obtained based on generic versus non-generic and semi-generic features

Feature set	Accuracy (%)	F-measure (%)			RTF
		Correct	Error	Average	
Non-generic	85.85	91.22	63.57	85.42	35.86815
Semi-generic	85.36	90.99	61.02	84.70	0.22128
Generic	81.09	88.80	39.53	78.47	0.00025
All	86.58	91.67	65.42	86.17	36.08944

Table 6 V-RNN classification accuracies and F-measures of error type classification obtained based on generic versus semi-generic and non-generic features

Feature set	Accuracy (%)	F-measure (%)			
		Correct	Substitution	Insertion	Average
Non-generic	84.77	91.28	61.51	00.32	83.40
Semi-generic	84.20	91.02	58.82	04.21	82.80
Generic	81.34	89.20	43.84	00.00	78.51
All	85.10	91.58	57.81	04.10	83.05

Assessment of the performance of the predictor features

Tables 5 and 6 display the V-RNN error detection and error type classification performance achieved on the test set using different features categories. Four settings are compared, corresponding to different categories of features used in the V-RNN training.

The results in Table 5, show that when using the non-generic features, the V-RNN model achieves 85.85% as classification accuracy in ASR error detection, but has the worst RTF. When using only the generic features, the model achieves slightly lower results with a classification accuracy of 81.09%. 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. Generic features are often used as a boosting factor for the performance of the ASR error detection systems and not as isolated features. Moreover, the very low RTF of the generic features makes them particularly suitable for real-time ASR error detection applications since, unlike non-generic features, we can potentially make a decision about the transcription errors in less time than a user needs for pronouncing the utterance. On the other hand, using the semi-generic feature set represents a good alternative to the non-generic features since it provides an absolute improvement of 4.27% in the classification accuracy and also shows a significant improvement on RTF as compared to non-generic features with an RTF value below 0.3. Also, by checking the F-measures, we can observe that a relatively remarkable improvement is obtained when using the semi-generic features as compared to the generic features alone. This improvement is especially relevant for the error labels where the F-measure passed from 39.53% when using generic features to 61.02% when using semi-generic features.

For error type classification, and taking a look at Table 6, we can observe that the F-measures change in correlation with the labels frequencies. Therefore, given that the insertion errors are less frequent than the substitution errors, the F-measure of

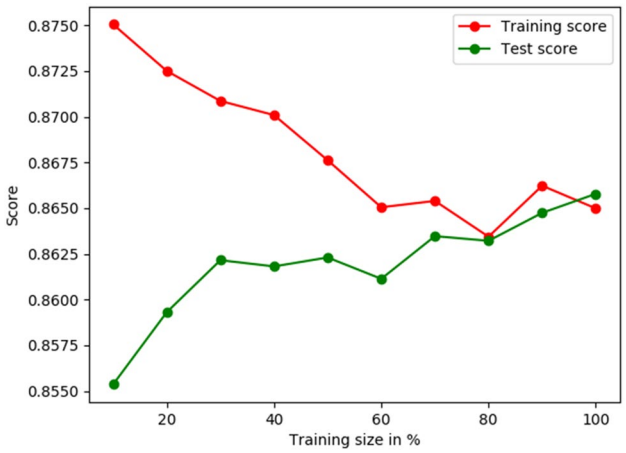

Fig. 4 Learning curve of the V-RNN with reduced training size on the ASR error detection task

the substitution is higher than the F-measure of the insertion. We observe also that there are small differences between the F-measures for the frequent labels (correct) obtained with each of the feature set. On the other hand, we observe a large differences between the F-measures 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 non-generic feature set. In contrast, the best error classification results was achieved when using the total feature set. However, when comparing the F-measures 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-measure of insertion labels when using the semi-generic features is 04.21% compared to only 00.32% when using the non-generic 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%.

Detailed analysis

Effect of the training set size

We are aimed at investigating the generalization ability of the proposed V-RNN. For this reason we varied the size of the training set from 10 to 100% in 10% steps and observed the effect on ASR error detection accuracy as shown in Fig. 4. The first observation from this figure, is that as the size of the training data was reduced, the gap between the training and evaluation data performances became larger. It is also shown that when the training data size was reduced from 30 to 20% and gradually to 10%, the performance of the V-RNN model on the test set quickly degraded. This indicates that 20% of training samples is insufficient to train the classifier accurately.

Furthermore, it is clearly shown that the training accuracies still in progress even when using 100% of the training set. Thus, as claimed before, the size of the

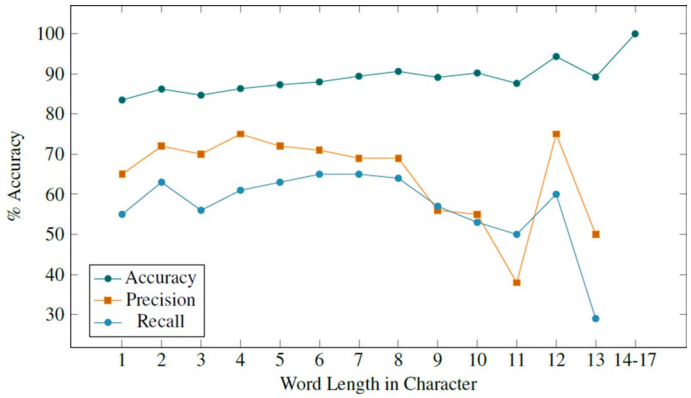

Fig. 5 General classification accuracy, recall and precision for the erroneous word prediction by word length on the test set

training set used in our experiments is not sufficient. Given that we are using only 28.3 h of annotated speech while in other work reported in the literature authors use hundreds of hours of annotated speech to train their models. For example in [6], the training size is composed of 215 h of lecture English speech, also in [20] they make of use a larger training set of 488 h of French Broadcast News with manual transcriptions.

Post detection analysis

In this section, we are interested in analyzing the outputs of our error detection Framework based on the V-RNN model. For this reason we preformed several comparisons between the Framework outputs and the ground truth labels in order to perceive recognition errors that are difficult to detect.

Word length analysis

This analysis seeks to study the impact of word length on speech recognition errors and the detection of these errors. The result of this analysis is summarized in Fig. 5. This Figure shows general classification accuracy as well as precision and recall for detecting error words in the test set. The first observation from this figure, is that there is a correlation between words length and the Framework performance, meaning that shorts erroneous word are very hard to be detected. We observe also that the words of length between 3 and 8 characters have the highest precision and recall, given that the average word length in English is 5.1 characters according to [21].

The present demonstrates that our proposed Framework is still suffering when dealing with too short words and very long words. A possible explanation for this might be that most of short words are function words (see next section), so they have ambiguous meaning which make their assessment very hard. Another possible explanation for this is that very long words are generally infrequent so they may not appear on the training set.

Table 7 Function and non function words analysis on the test set

	Accuracy (%)	F-measure (%)		
		Correct	Error	Avg
Function words	85.85	91.22	63.57	85.42
Non function words	87.52	92.18	69.18	87.23

Table 8 Word position analysis on the test set

	Accuracy (%)	F-measure (%)		
		Correct	Error	Avg
Start words	79.65	85.38	66.57	79.42
Middle words	88.12	92.92	63.26	87.64
End words	85.13	89.88	72.01	84.89

Function words analysis

Function words, called also stop words, are a category of words whose syntactic role is more important than the semantic role. These are generally the most frequent words in the corpus and are used to express grammatical relationships among other words within a sentence. On the other hand, non function words are generally semantic support words.

For this reason we collected a list of 179 English function words. Usually they are non-content words like conjunctions (i.e. for, and, nor, but), determiners (i.e. the, my, some, this), prepositions (i.e. of, on , out), etc. It is clearly shown in Table 7 that the ASR error detection Framework performs better on non function words than on function ones. Furthermore, we noticed that about 38% of function words have a length of 3 characters or less. Thus this confirms the results presented in the previous subsection.

Word position analysis

Considering that our proposed V-RNN model predicts the label of the current word given its context, we believe that the position of the word within the utterance may affect the Framework performance.

In this direction, we look at the effect of word position on the system performance. We consider three position, Start Words, Middle Words and End Words. It is clearly shown in Table 8 that words in the middle of utterances present the best performance, followed by words situated at the end of utterances. This means that words in the middle benefit from both context, *left* and *right*, given that several features are base on the word context. While words at the beginning of the utterance remain difficult in assessment, meaning that it is hard to decide whether these words are correct or not. This is because the V-RNN works in forward, so it considers only the past to predict the correctness of the current word.

Word context analysis

Considering that the V-RNN model takes as input the previous word label in addition to the current input, we analyse the effect of the number of errors in each utterance on the

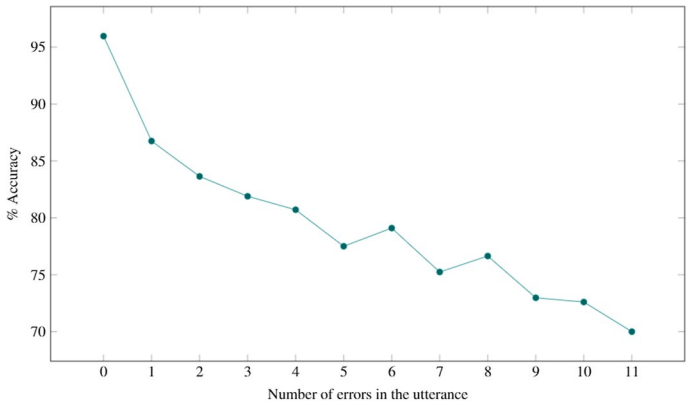

Fig. 6 General classification accuracy by number of error in the utterance on the test set

Framework performance. For this reason we experiment each group of utterances that have the same number of errors separately. Results are reported in Fig. 6.

It is clearly shown that the system performance decreases dramatically while the number of effective errors increases. We observe that, when there are 1, 2 or 3 errors in the utterance, the Framework performance archives respectively 86.74%, 83.64% and 81.89% of general accuracy. When the utterance has more errors the Framework is less accurate and could decrease to less than 70% in a context containing 11 errors or more. One explanation to this is that utterances that contain height number of errors are generally caused by height acoustic noise or a speech overlap, given that 77.3% of the utterances in the test set have 3 errors or less.

Conclusion

In this paper we have evaluated and compared the effectiveness and efficiency of state-of-the-art features and models for automatic speech recognition error detection in a unified framework. We put special emphasis on handling the ASR errors independently from the decoder's internal information using a generic Framework based on a set of predictor features derived exclusively from the recognizer output. The experimental results have shown that the generic features can provide confirmatory evidence of the correctness of word in the output transcription of ASR systems, nevertheless the best ASR error detection accuracy (86.58%) is achieved when combining all predictor features. The main source of computational complexity appears to originate from the non-generic features (around 35.9 RTF). On the other hand, using the semi-generic feature set represents a good trade-off between accuracy (85.36%) and RTF (below 0.3), which makes them suitable for time-constrained ASR error detection applications. The same results could be generalized for the ASR error classification. The results have also shown that the V-RNN model significantly outperforms the other RNN models on the ASR error detection and classification. The V-RNN model is not only efficient for ASR error detection but also it is 10 times faster than the best stat-of-the-art based RNN models. However, despite these promising results, the best performing system is still suffering when dealing with words at the beginning of the

utterance, function words, very short and very long words. Therefore there is clearly substantial room for improvement. We believe that using more data for training and combining advanced RNNs units such as LSTM or GRU, which are claimed the most effective RNNs, and label dependency strategy would give better results. In addition, more fundamental issues on deletion errors detection and cross evaluation of the error detection system on different transcriptions from different ASR systems need to be addressed for significant progress.

Abbreviations

ASR: Automatic Speech Recognition; BBC: British Broadcasting Corporation; BLSTM: Bidirectional Long Short-Term Memory; LDA: Latent Dirichlet Allocation; LM: Language Models; LVCSR: Large Vocabulary Continuous Speech Recognition; MGB: Multi-Genre Broadcast; MLP: Multi Layer Perceptron; OOV: Out-of-Vocabulary; RNN: Recurrent Neural Network; RNNLM: Recurrent Neural Network Language Model; RTF: Real Time Factor; ULSTM: Unidirectional Long Short-Term Memory; V-RNN: Variant Recurrent Neural Network; WCN: Word Confusion Network; WER: Word Error Rate.

Acknowledgements

Not applicable.

Authors' contributions

Not applicable.

Funding

Not applicable. This research received no specifc grant from any funding agency in the public, commercial, or not-for-proft sectors.

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Laboratory of Information Technologies, National School of Applied Sciences, University of Chouaib Doukkali, El Jadida, Morocco. [2] LAROSERI Laboratory, University of Chouaib Doukkali, El Jadida, Morocco. [3] Speech and Hearing Group, Department of Computer Science, University of Sheffield, Sheffield, UK.

Received: 5 September 2020 Accepted: 4 December 2020
Published online: 06 January 2021

References

1. Errattahi R, El Hannani A. Recent advances in LVCSR: a benchmark comparison of performances. Int J Electr Comput Eng. 2017;7(6):3358–68.
2. Errattahi R, El Hannani A, Ouahmane H. Automatic speech recognition errors detection and correction: a review. Procedia Comput Sci. 2018;128:32–7.
3. Errattahi R, El Hannani A, Hain T, Ouahmane H. System-independent asr error detection and classification using recurrent neural network. Comput Speech Language. 2019;55:187–99.
4. Zhang R, Rudnicky AI. Word level confidence annotation using combinations of features. In: The proceedings of the European conference on speech communication and technology (EuroSpeech); 2001. p. 2105–8.
5. Gibson M, Hain T. Application of SVM-based correctness predictions to unsupervised discriminative speaker adaptation. In: The proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP); 2012. p. 4341–4.
6. Ogawa A, Hori T. Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks. Speech Commun. 2017;89:70–83.
7. Errattahi R, El Hannani A, Hain T, Ouahmane H. Towards a generic approach for automatic speech recognition error detection and classification. In: The proceedings of the IEEE 4th international conference on advanced technologies for signal and image processing (ATSIP); 2018. p. 1–6.
8. Errattahi R, El Hannani A, Salmam FZ, Ouahmane H. Incorporating label dependency for asr error detection via rnn. Procedia Comput Sci. 2019;148:266–72.
9. Errattahi R, Deena S, El Hannani A, Ouahmane H, Hain T. Improving asr error detection with rnnlm adaptation. In: The proceedings of the IEEE spoken language technology workshop (SLT); 2018. p. 190–6.
10. Graves A, Schmidhuber J. Offline handwriting recognition with multidimensional recurrent neural networks. In: The proceedings of the advances in neural information processing systems (NIPS); 2009. p. 545–52.
11. Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. In: The proceedings of the international conference on machine learning (ICML); 2014. p. 1764–72.

12. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. In: The proceedings of the advances in neural information processing systems (NIPS); 2014. p. 3104–12.
13. Bell P, Gales MJ, Hain T, Kilgour J, Lanchantin P, Liu X, McParland A, Renals S, Saz O, Wester M, et al. The MGB challenge: evaluating multi–genre broadcast media transcription. In: The proceedings of IEEE workshop on automatic speech recognition and understanding (ASRU); 2015.
14. Deena S, Hasan M, Doulaty M, Saz O, Hain T. Combining feature and model-based adaptation of RNNLMs for multi-genre broadcast speech recognition. In: The proceedings of the annual conference of the international speech communication association (INTERSPEECH); 2016. p. 2343–7.
15. Deena S, Ng RWM, Madhyashta P, Specia L, Hain T. Semi-supervised adaptation of RNNLMs by fine-tuning with domain-specific auxiliary features. In: The Proceedings of the annual conference of the international speech communication association (INTERSPEECH); 2017.
16. Wang K, Thrasher C, Viegas E, Li X, Hsu B-jP. An overview of microsoft web n-gram corpus and applications. In: The proceedings of the NAACL HLT 2010 demonstration session; 2010. p. 45–8.
17. Chen X, Liu X, Qian Y, Gales MJ, Woodland PC. CUED-RNNLM—an open-source toolkit for efficient training and evaluation of recurrent neural network language models. In: The proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP); 2016. p. 6000–4.
18. Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. J Mach Learn Res. 2003;3:993–1022.
19. Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: The proceedings of the international conference on machine learning (ICML); 2010. p. 807–14.
20. Ghannay S, Esteve Y, Camelin N, Deléglise P. Acoustic word embeddings for ASR error detection. In: The Proceedings of the annual conference of the international speech communication association (INTERSPEECH); 2016. p. 1330–4.
21. Bochkarev V, Shevlyakova A, Solovyev V. Average word length dynamics as indicator of cultural changes in society; 2012. arXiv preprint arXiv:1208.6109.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com