Word Error Rate Estimation for Speech Recognition: e-WER

Ahmed Ali
Qatar Computing Research Institute
QCRI
Doha, Qatar
amali@qf.org.qa

Steve Renals
Centre for Speech Technology Research
University of Edinburgh
UK
s.renals@ed.ac.uk

Abstract

Measuring the performance of automatic speech recognition (ASR) systems requires manually transcribed data in order to compute the word error rate (WER), which is often time-consuming and expensive. In this paper, we propose a novel approach to estimate WER, or e-WER, which does not require a gold-standard transcription of the test set. Our e-WER framework uses a comprehensive set of features: ASR recognised text, character recognition results to complement recognition output, and internal decoder features. We report results for the two features; black-box and glass-box using unseen 24 Arabic broadcast programs. Our system achieves 16.9% WER root mean squared error (RMSE) across 1,400 sentences. The estimated overall WER e-WER was 25.3% for the three hours test set, while the actual WER was 28.5%.

1 Introduction

Automatic Speech Recognition (ASR) has made rapid progress in recent years, primarily due to advances in deep learning and powerful computing platforms. As a result, the quality of ASR has improved dramatically, leading to various applications, such as speech-to-speech translation, personal assistants, and broadcast media monitoring. Despite this progress, ASR performance is still closely tied to how well the acoustic model (AM) and language model (LM) training data matches the test conditions. Thus, it is important to be able to estimate the accuracy of an ASR system in a particular target environment.

Word Error Rate (WER) is the standard approach to evaluate the performance of a large vocabulary continuous speech recognition (LVCSR) system. The word sequence hypothesised by the ASR system is aligned with a reference transcription, and the number of errors is computed as the sum of substitutions (S), insertions (I), and deletions (D). If there are N total words in the reference transcription, then the word error rate WER is computed as follows:

$$\text{WER} = \frac{I + D + S}{N} \times 100. \quad (1)$$

To obtain a reliable estimate of the WER, at least two hours of test data are required for a typical LVCSR system. In order to perform the alignment, the test data needs to be manually transcribed at the word level – a time-consuming and expensive process. It is, thus, of interest to develop techniques which can estimate the quality of an automatically generated transcription without requiring a gold-standard reference.

Such quality estimation techniques have been extensively investigated for machine translation (Specia et al., 2013), with extensions to spoken language translation (Ng et al., 2015, 2016). Although there is a long history of exploring word-level confidence measures for speech recognition (Evermann and Woodland, 2000; Cox and Dasmahapatra, 2002; Jiang, 2005; Seigel et al., 2011; Huang et al., 2013), there has been less work on the direct estimation of speech recognition errors.

Seigel and Woodland (2014) studied the detection of deletions in ASR output using a conditional random field (CRF) sequence model to detect one or more deleted word regions in ASR output. Ghannay et al. (2015) used word embeddings to build a confidence classifier which labeled each word in the recognised word sequence with an error or a correct label. Tam et al. (2014) investigated the use of a recurrent neural network (RNN) language model (LM) with complementary deep neural network (DNN) and Gaussian Mix-

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 20–24
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

ture Model (GMM) acoustic models in order to identify ASR errors, based on the assumption that when two ASR systems disagree on an utterance region, then it is most likely an error.

Ogawa and Hori (2015) investigated using deep bidirectional recurrent neural networks (DBRNNs) to detect errors in ASR results. They explored four tasks for ASR error detection and recognition rate estimation: confidence estimation, out-of-vocabulary (OOV) word detection, error type classification, and recognition rate estimation. In an extension to this work, Ogawa et al. (2016); Ogawa and Hori (2017) investigated the estimation of speech recognition accuracy based on the classification of error types, in which sequence classification was performed by a CRF. Each word in a hypothesised word sequence was classified into one of three categories: correct, substitution error, or insertion error. Their study did not estimate the presence of deletions, and consequently cannot estimate the WER.

Jalalvand et al. (2016) developed a tool for ASR quality estimation, TranscRater, which is capable of predicting WER per utterance. This approach is based on a large set of extracted features (which do not require internal access to the ASR system) used to train a regression model (e.g., extremely randomised trees), and can also rank different transcriptions from multiple sources (Negri et al., 2014; de Souza et al., 2015; Jalalvand and Falavigna, 2015; Jalalvand et al., 2015a,b). TranscRater provides a WER per utterance, reporting the results as the MAE with respect to a reference transcription. This work did not report WER estimates for complete recordings or test sets, although it is possible that this could be done using utterance length estimates.

In this paper, we build on these contributions to develop a system to directly estimate the WER of an ASR output hypothesis. Our contributions are: (*i*) a novel approach to estimate WER per sentence and to aggregate them to provide WER estimation per recording or for a whole test set; (*ii*) an evaluation of our approach which compares the use of "black-box" features (without ASR decoder information) and "glass-box" features which use internal information from the decoder; and (*iii*) a release of the code and the data used for this paper for further research[1].

[1] https://github.com/qcri/e-wer

2 e-WER Framework

Estimating the probability of error of each word in a recognised word sequence has been successfully used to detect insertions, substitutions, and interword deletions (Ogawa et al., 2016; Ogawa and Hori, 2015; Ghannay et al., 2015; Jalalvand and Falavigna, 2015; Seigel and Woodland, 2014). However, these local estimates do not provide an estimate of the overall pattern of error, such as the total number of deletions in an utterance.

In our framework, we use two speech recognition systems; a word-based LVCSR system and a grapheme-sequence based system. Following Tam et al. (2014), we assume that when two corresponding ASR systems disagree on a sentence or part of a sentence, there is a pattern of error to be learned. Our architecture also benefits from utterance-based LVCSR decoder features including the total number of frames, the average log likelihood and the duration. Intuitively, we correlate short sentences with less context and assume that LM scoring will not be able to capture long context. Therefore, e-WER is defined as follows:

$$\text{e-WER} = \frac{\text{ERR}}{\hat{N}} \times 100\% \quad (2)$$

Our model is required to predict two values for each utterance: ERR and $\hat{N}$. Given that each is integer-valued, we decided to frame their estimation as a classification task rather than a regression problem as shown in equations 3 and 4. Each class represents a specific word count. We limit the total number of classes to a maximum of C in ERR, with range from 0 to C. However, the total number of classes for $\hat{N}$ is $C - K$ to avoid estimating an utterance length of zero, with a range from K to C. If an utterance has more than C words or less than K words, it will thus be penalised by the loss function,

$$\text{ERR} = \arg\max_{c_j \in C} P(c_j | x_1, x_2, ..., x_n) \quad (3)$$

$$\hat{N} = \arg\max_{k_j \in C-K} P(k_j | x_1, x_2, ..., x_n) \quad (4)$$

Table 1 shows that fewer than 5% of the sentences have more than 20 words, and it is very unlikely to have an utterance with fewer than 2 words. We trained our system with $C = 20$ and $K = 2$. Since our approach predicts ERR and $\hat{N}$ for each sentence, it is possible to aggregate each of the two

values across the entire test set in order to estimate the overall WER, as shown in section 3.

2.1 e-WER features

To estimate e-WER, we combine features from the word-based LVCSR system with features from the grapheme-based system. By running both word-based and character-based ASR systems, we are able to align their outputs against each other.

We split the studied features into four groups

- *L*: lexical features – the word sequence extracted from the LVCSR.
- *G*: grapheme features – character sequence extracted from the grapheme recognition.
- *N*: numerical features – basic features about the speech signal, as well as grapheme alignment error details.
- *D*: decoder features – total frame count, average log-likelihood, total acoustic model likelihood and total language model likelihood.

Similar to previous research in ASR quality estimation, we refer to $\{L,G,N\}$ as the black-box features, and $\{L,G,N,D\}$ as the glass-box features, which are used to estimate the total number of words $\hat{N}$, and the total number of errors ERR in a given sentence.

2.2 Classification Back-end

We deployed a feed-forward neural network as a backend classifier for e-WER. The deployed network in this work has two fully-connected hidden layers (ReLU activation function), with 128 neurons in the first layer and 64 neurons in the second layer followed by a softmax layer. A minibatch size of 32 was used, and the number of epochs was up to 50 with an early stopping criterion.

2.3 Data

The e-WER training and development data sets are the same as the Arabic MGB-2 development and evaluation sets (Ali et al., 2016; Khurana and Ali, 2016), which is comprised of audio extracted from Al-Jazeera Arabic TV programs recorded in the last months of 2015. To test whether our approach generalises to test sets from a different source, and not tuned to the MGB-2 data set, we validated our results on three hours test set collected by BBC Monitoring during November 2016, as part of the SUMMA project[2].

[2] http://summa-project.eu

	Train	Dev	Test
Number of programs in corpus	17	17	24
Utterances	58K	56K	1.4K
Duration (in hours)	9.9	10.2	3.2
2-20 words sentences	96%	95%	96%
Word count (N)	**75K**	**69K**	**20K**
ASR word count (hyp)	58K	60K	18K
WER	42.6%	33.1%	28.5%
Total INS	1.9K	1.8K	130
Total DEL	19.1K	10.2K	2.6K
Total SUB	11.1K	10.8K	2.9K
ERR count (ERR)	**32.1K**	**22.8K**	**5.7K**

Table 1: Analysis of the train, dev and test data.

	MAE/Dev			**MAE/Test**		
	ERR	$\hat{N}$	e-WER	ERR	$\hat{N}$	e-WER
glass-box	1.6	1.8	13.8	1.7	1.7	12.3
black-box	1.8	2.2	28.4	1.9	2.3	24.7

Table 2: MAE per sentence reported for the glass-box and black-box features.

3 Experiments and discussions

We trained two DNN systems to estimate $\hat{N}$ and ERR separately. We explored training both a black-box based DNN system (without the decoder features) and a glass-box system using the decoder features. Overall, four systems were trained: two glass-box systems and two black-box systems. We used the same hyper-parameters across the four systems. Tables 2 and 3 present the e-WER performance in terms of the mean absolute error (MAE) and root mean squared error (RMSE) per sentence for ERR, $\hat{N}$ and the estimated WER for the dev and test sets with reference to the errors computed using a gold-standard reference. As expected, the glass-box features help to reduce MAE and RMSE for both ERR and $\hat{N}$. Although the difference between the black-box estimation and the glass-box results is not big for ERR and N, we can see that the impact becomes substantial on the estimated WER per sentence, which is almost double the error in both MAE and RMSE per sentence.

Table 4 reports the overall performance on the dev and on the test set. Across the 17 programs in the MGB-2 dev data, the actual WER is 33.1%, and the glass-box e-WER is 29.3%, while the black-box e-WER is 30.9%. Evaluating the same models on the 24 programs in the test data set results in an actual WER of 28.5%, while the glass-box e-WER is 25.3%, and the black-box e-WER is 30.3%.

Tables 2 and 3 show the glass-box features outperformed the black-box features in predicting both ERR and $\hat{N}$. Furthermore, the performance

	RMSE/Dev			RMSE/Test		
	ERR	$\hat{N}$	e-WER	ERR	$\hat{N}$	e-WER
glass-box	2.2	2.1	18.3	2.3	2.2	16.9
black-box	2.4	2.7	36.1	2.6	2.9	35.0

Table 3: RMSE per sentence reported for the glass-box and the black-box features.

	Actual/estimated WER		
Data	Reference	glass-box	black-box
Dev	33.1%	29.3%	30.9%
Test	28.5%	25.3%	30.3%

Table 4: Overall WER across the dev and the test data set.

of the estimated WER per sentence in the glass-box is substantially better than the black-box for both development and test sets. Table 4 indicates that the glass-box estimate is systematically lower than the black-box estimate. To further visualise these results, figure 1 plots the cumulative WER and e-WER across the three hours test set. This plot indicates that the glass-box estimate is continually lower than the black-box estimate. The large difference during the first 30 minutes arises owing the glass-box system is capable of better estimation with less data compared to the black-box system.

We estimate $\hat{N}$ and ERR separately. Therefore, our system is capable of estimating the WER at different levels of granularity. We visualise the prediction per program. In scenarios such as media-monitoring, where the main objective is to have a robust monitoring system for specific programs, we plot the WER across the 24 programs in the test set, and we can see in figure 2 that both the glass-box and black-box estimation are following the gold-standard WER per program. However, unlike predicting word count $\hat{N}$ or error count ERR, we can see that the black-box, in general, over-estimates the WER, while the glass-box system under-estimates WER similar to figure 1. One can argue from figure 2 that the decoder features are not helping in programs with high WER. We found both systems to be useful for reporting WER per program.

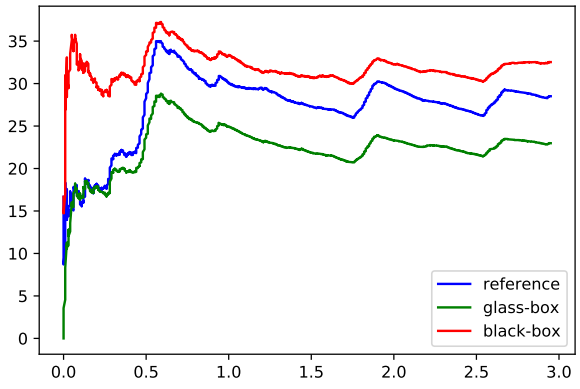

Figure 1: Test set cumulative WER over all sentences (X-axis is duration in hours and Y-axis is WER in %).

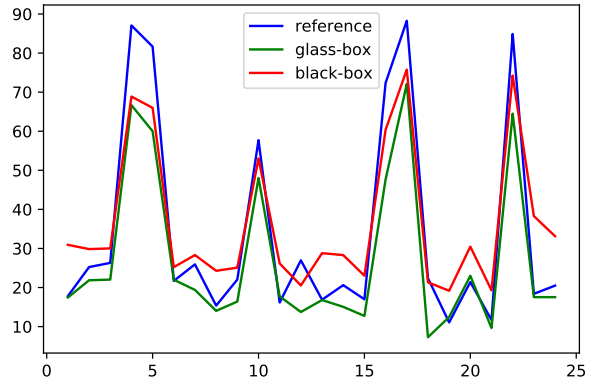

Figure 2: WER estimated over 24 programs on the test data.

4 Conclusions

This paper presents our efforts in predicting speech recognition word error rate without requiring a gold-standard reference transcription. We presented a DNN based classifier to predict the total number of errors per utterance and the total word count separately. Our approach benefits from combining word-based and grapheme-based ASR results for the same sentence, along with extracted decoder features. We evaluated our approach per sentences and per program. Our experiments have shown that this approach is highly promising to estimate WER per sentence and we have aggregated the estimated results to predict WER for complete recordings, programs or test sets without the need for a reference transcription. For our future work, we shall continue our investigation into approaches that can estimate the word error rate using convolutional neural networks. In particular, we would like to explore combining the DNN numerical features with the CNN word embedding features.

References

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The MGB-2 challenge: Arabic multi-dialect broadcast media recognition. In *Proc IEEE SLT*.

Stephen Cox and Srinandan Dasmahapatra. 2002. High-level approaches to confidence estimation in speech recognition. *IEEE Transactions on Speech and Audio processing* 10(7):460–471.

José GC de Souza, Hamed Zamani, Matteo Negri, Marco Turchi, and Daniele Falavigna. 2015. Multitask learning for adaptive quality estimation of automatically transcribed utterances. In *HLT-NAACL*. pages 714–724.

Gunnar Evermann and PC Woodland. 2000. Posterior probability decoding, confidence estimation and system combination. In *Proc. Speech Transcription Workshop*. Baltimore, volume 27, page 78.

Sahar Ghannay, Yannick Esteve, and Nathalie Camelin. 2015. Word embeddings combination and neural networks for robustness in asr error detection. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, pages 1671–1675.

Po-Sen Huang, Kshitiz Kumar, Chaojun Liu, Yifan Gong, and Li Deng. 2013. Predicting speech recognition confidence using deep learning with word identity and score features. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pages 7413–7417.

Shahab Jalalvand and Daniele Falavigna. 2015. Stacked auto-encoder for ASR error detection and word error rate prediction. In *Interspeech*.

Shahab Jalalvand, Daniele Falavigna, Marco Matassoni, Piergiorgio Svaizer, and Maurizio Omologo. 2015a. Boosted acoustic model learning and hypotheses rescoring on the chime-3 task. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, pages 409–415.

Shahab Jalalvand, Matteo Negri, Falavigna Daniele, and Marco Turchi. 2015b. Driving rover with segment-based asr quality estimation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. volume 1, pages 1095–1105.

Shahab Jalalvand, Matteo Negri, Marco Turchi, José GC de Souza, Daniele Falavigna, and Mohammed RH Qwaider. 2016. Transcrater: a tool for automatic speech recognition quality estimation. *ACL 2016* page 43.

Hui Jiang. 2005. Confidence measures for speech recognition: A survey. *Speech Communication* 45(4):455–470.

Sameer Khurana and Ahmed Ali. 2016. QCRI advanced transcription system (QATS) for the Arabic Multi-Dialect Broadcast Media Recognition: MGB-2 Challenge. In *SLT*.

Matteo Negri, Marco Turchi, José GC de Souza, and Daniele Falavigna. 2014. Quality estimation for automatic speech recognition. In *COLING*. pages 1813–1823.

Raymond WM Ng, Kashif Shah, Lucia Specia, and Thomas Hain. 2015. A study on the stability and effectiveness of features in quality estimation for spoken language translation. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Raymond WM Ng, Kashif Shah, Lucia Specia, and Thomas Hain. 2016. Groupwise learning for asr k-best list reranking in spoken language translation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pages 6120–6124.

Atsunori Ogawa and Takaaki Hori. 2015. Asr error detection and recognition rate estimation using deep bidirectional recurrent neural networks. In *ICASSP*.

Atsunori Ogawa and Takaaki Hori. 2017. Error detection and accuracy estimation in automatic speech recognition using deep bidirectional recurrent neural networks. *Speech Communication* 89:70–83.

Atsunori Ogawa, Takaaki Hori, and Atsushi Nakamura. 2016. Estimating speech recognition accuracy based on error type classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24(12):2400–2413.

Matthew Stephen Seigel and Philip C Woodland. 2014. Detecting deletions in asr output. In *ICASSP*. IEEE, pages 2302–2306.

Matthew Stephen Seigel, Philip C Woodland, et al. 2011. Combining information sources for confidence estimation with crf models. In *Interspeech*. pages 905–908.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst – a translation quality estimation framework. In *ACL: System Demonstrations*. pages 79–84.

Yik-Cheung Tam, Yun Lei, Jing Zheng, and Wen Wang. 2014. Asr error detection using recurrent neural network language model and complementary asr. In *ICASSP*. IEEE, pages 2312–2316.