INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

ASR Confidence Estimation with Speaker-Adapted Recurrent Neural Networks

Miguel Ángel del-Agua, Santiago Piqueras, Adrià Giménez, Alberto Sanchis, Jorge Civera, Alfons Juan

MLLP, DSIC, Universitat Politècnica de València (UPV), Spain.
{mdelagua,spiqueras,agimenez,josanna,jcivera,ajuan}@dsic.upv.es

Abstract

Confidence estimation for automatic speech recognition has been very recently improved by using Recurrent Neural Networks (RNNs), and also by speaker adaptation (on the basis of Conditional Random Fields). In this work, we explore how to obtain further improvements by combining RNNs and speaker adaptation. In particular, we explore different speaker-dependent and -independent data representations for Bidirectional Long Short Term Memory RNNs of various topologies. Empirical tests are reported on the LibriSpeech dataset showing that the best results are achieved by the proposed combination of RNNs and speaker adaptation.

Index Terms: speech recognition, speaker adaptation, confidence measures, recurrent neural networks, blstm

1. Introduction

Confidence estimation (CE) has been broadly investigated in automatic speech recognition (ASR) with the aim of assessing the reliability of the ASR output [1]. Over the years, an approach that has demonstrated to be very effective is to consider CE as a classical two-category (correct or incorrect) pattern recognition problem. Following this approach, CE has been gradually improved by exploring novel input features and by designing more and more accurate classifiers [1, 2, 3, 4].

Recent improvements to CE include the use of Recurrent Neural Networks (RNNs) [4] and speaker adaptation [3]. On the one hand, the use of RNNs has yielded better performance due to its ability to model context [4]. On the other hand, experimental results have shown that speaker-adapted classifiers such as naïve Bayes, logistic regression and conditional random fields outperform their non-adapted counterparts [3]. It is worth noting, however, that RNNs and speaker-adaptation have been studied separately, and thus it is still unclear whether using them in conjunction would lead to further improvements in accuracy.

In this work, we explore possible ways to use RNNs and speaker-adaptation techniques in conjunction. In particular, we propose to use the long short-term memory (LSTM) version of RNNs [5]. In this way, the vanishing gradient problem will be conveniently addressed in the case of long-span relations [6], while both history and future contexts will be modelled at the same time through its bidirectional version (BLSTMs). Furthermore, we propose to apply speaker adaptation techniques to LSTM models through the use of speaker-dependent input features based on their specific vocabulary, as well as training speaker-dependent models.

The content of the paper is organized as follows. The proposed speaker-adapted LSTM architecture is presented in Section 2. Empirical results on the LibriSpeech dataset are reported in Section 3, showing that the best results are achieved by the proposed combination of RNNs and speaker adaptation. Finally, the conclusions of this work are summarized in Section 4.

2. Speaker-Adapted LSTM Networks for Confidence Estimation

Recent work on CE [4] suggests that using temporal context by means of RNNs outperform other approximations where the sequential dependence cannot be exploited. For that reason, we propose to use LSTM networks as a further step towards context dependency. Aside from circumventing the vanishing gradient problem, LSTM networks introduce a temporal dependence over the entire segment by means of its bidirectional version. In this work, we use LSTM networks with both unidirectional and bidirectional layers, and thus we will refer to them simply as LSTMs.

What makes LSTM [5] networks different from RNNs is the use of purpose built-in memory cells which perform element-wise multiplications to control the information flow in the network. This memory cells are able to store information for a long period of time because of a gating structure that determines when the input is relevant enough to remember, when it should continue to remember or forget, and when it should yield an output. Specifically, the LSTM cells replace the activation function of a classical RNN with the following set of equations:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (2)$$
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$
$$h_t = o_t \tanh(c_t) \quad (5)$$

where σ is the logistic sigmoid function and i, f, c, o, h represent five different vectors at time t from each gate: input, forget, cell memory activation, output and hidden layer, respectively. As depicted in Fig. 1, the LSTM Network proposed in this work follows a classical LSTM architecture. To use it in CE, input vectors at word-level are composed of two parts: a compact representation of the word identity and a set of word-level features extracted from ASR word-lattices.

Word identities have been included in the input vectors as they have been shown to be very useful in CE [2, 3, 4, 7]. To this end, we have not used a conventional one-hot encoding since this would entail a number of parameters growing linearly with the vocabulary size. Instead, we have used a more compact global word vector representation based on a

http://dx.doi.org/10.21437/Interspeech.2016-1142

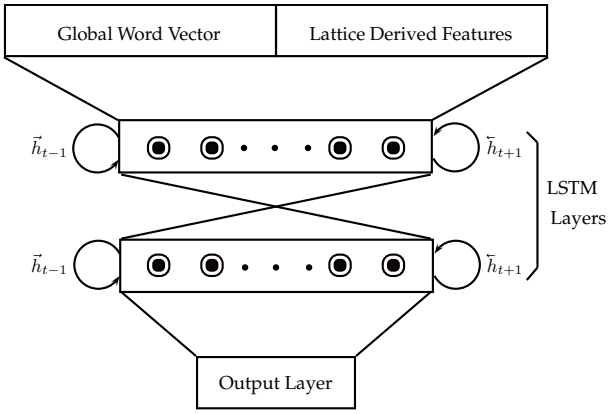

Figure 1: LSTM architecture for CE.

"GloVe" model [8]. This is an embedding model, which tries to maintain the semantic similarities between words in their vector representation. Two very similar words will result in two very similar vectors. It is trained on the non-zero entries of a global word-word co-occurrence matrix which tabulates how frequently words co-occur with one another in a given corpus, in this case, the same training as the one used for CE.

Given a sequence of input vectors $\mathcal{X} = (\boldsymbol{x_1}, ..., \boldsymbol{x_T})$ which represents a sequence of T recognized words, the network produces a sequence of output vectors $\mathcal{Y} = (\boldsymbol{y_1}, ..., \boldsymbol{y_T})$ defining a probability distribution over each class c ($c = \{\text{correct}, \text{incorrect}\}$). These probabilities correspond to the network's estimate of observing each class c at time t given $\mathcal{X}$.

The LSTM network is trained to minimize the cross-entropy error of the targets using a softmax output layer with 2 output units representing the two-category class using the standard back-propagation through time algorithm (BPTT) [9]. Given a target sequence $\mathcal{Z} = (\boldsymbol{z_1}, ..., \boldsymbol{z_T})$, the network minimizes the negative log-probability of the target sequence given the input sequence:

$$-\log P(\mathcal{Z}|\mathcal{X}) = -\sum_{t=1}^{T} \log y_t^{z_t} \tag{6}$$

After an LSTM network has been estimated based on Eq. (6) using a set of N training pairs $\{\mathcal{X}, \mathcal{Z}\}_1^N$, we propose to adapt the LSTM to a new speaker by performing a few more iterations of the BPTT algorithm using a small subset of training pairs belonging to that speaker. It is worth mentioning that this adaptation also implies adapting the system to the vocabulary of the speaker, so it becomes necessary to re-estimate the global word vector model taking into account the new vocabulary of the speaker concerned. This is needed to ensure that the same word representation is used before and after adaptation.

3. Experiments

3.1. Experimental Setup

The proposed approach has been evaluated in the LibriSpeech ASR corpus [10]. The ASR system has been built using the transLectures-UPV toolkit [11], which is an open source set of tools for designing an ASR system from scratch. Acoustic models have been trained using the train-clean-100 LibriSpeech subset (100 hours). They consist of an hybrid HMM-DNN built on top of MFCC-CMLLR features. The DNN has been trained with a context window of 11 frames, 7 hidden layers with ReLu activation functions and 2048 units each. The number of target tied-states accounts for a total of 8132. As language model, we have used the pre-built 4-gram provided by the authors in the release of the corpus.

The official dev-other and test-other subsets of the LibriSpeech corpus have been used to adjust and evaluate CE models in a speaker-independent (SI) fashion. Also, 50h from the train-other-500 LibriSpeech subset were randomly selected for the training of the SI CE models. The main statistics of this experimental setting can be found in Table 1.

Table 1: Statistics of the speaker-independent setting.

Set	Duration (h)	Words	Vocab	WER
Train	49	475K	27K	15.6
Dev	5.3	51K	7K	21.2
Test	5.1	52K	8K	23.1

Additionally, 20 speakers not used in the SI experiments were randomly selected from the train-other-500 subset in order to evaluate speaker adaptation of the SI CE models. Specific training, development and test subsets were built for each speaker using their own speech data. Global statistics of this speaker-dependent (SD) setting are shown in Table 2.

Table 2: Statistics of the speaker-dependent setting.

Set	Duration (h)	Words	Vocab	WER
Train	5.9	54.9K	8.6K	26.2
Dev	2	19.1K	4.6K	26.3
Test	2	19.3K	4.6K	25.5

It is worth mentioning that all the speakers in LibriSpeech have almost the same amount of speech so as not to suffer from unbalanced speaker data. Therefore, in our SD partition, there is almost the same amount of data for each speaker in order to adapt, adjust parameters and evaluate.

3.2. Evaluation metrics

We have used three metrics to evaluate the performance of the CE classifiers: the area under a ROC curve (AUC), the classification error rate (CER) and the normalized cross entropy (NCE).

Let us assume that ASR output results in C correctly recognized words and I mis-recognized words. Let *False Rejection* be the number of correctly recognized words with confidence lower than a decision threshold τ ($FR(\tau)$) and, equivalently, let *True Rejection* be the number of mis-recognized words with confidence lower than τ ($TR(\tau)$). The *False Rejection Rate* (FRR(τ)) and the *True Rejection Rate* (TRR(τ)) for a decision threshold τ are computed as:

$$FRR(\tau) = \frac{FR(\tau)}{C} \qquad TRR(\tau) = \frac{TR(\tau)}{I} \tag{7}$$

A *Receiver Operating Characteristic* (ROC) curve represents TRR(τ) against FRR(τ) for different values of τ. The AUC provides an adequate overall estimation of the classification accuracy, being 100 a perfect classification and 50 a random classification (diagonal ROC curve).

The *Classification Error Rate* (CER) for a decision threshold τ is computed as:

$$CER(\tau) = \frac{FR(\tau) + (I - TR(\tau))}{C + I} \cdot 100 \quad (8)$$

A *baseline* CER can be computed by classifying all the words as correct (i.e. $\tau = 1$):

$$CER(1) = \frac{I}{C + I} \cdot 100 \quad (9)$$

Clearly, $\tau = 1$ is not necessarily optimal in the sense of minimizing Eq. (8). Therefore, it is convenient to consider the classification threshold $\tau = \tau^*$, which minimizes the CER criterion (usually that which provided the minimum CER in a *development set*):

$$\tau^* = \arg\min_{\tau} CER(\tau) \quad (10)$$

The *Normalized Cross Entropy* (NCE) is defined as the average log distance of the score to the real class. It attains its maximum of 1 when the system provides perfect confidence measures, that is, $0/1$ values allowing us to perfectly discriminate between correctly and incorrectly recognized words.

3.3. Results

As was mentioned in Section 2, a part of the input features of the LSTM Network are extracted from an ASR word-lattice. In the experiments, we used 5 word-lattice based features commonly used in CE [3]:

- SP: Word Acoustic log-score per time frame (10ms).
- D: Duration (in ms) of the word.
- NL: Length of the N-gram in which the word was decoded.
- PAvg: Word posterior probability computed as the average of frame-based posteriors [12].
- PMax: Like PAvg but using the maximum instead of the average [12].

On the other hand, a global word vector was obtained for SI and SD experiments, respectively, using the training data of each experimental setting. The optimal size of the word vectors was evaluated on the SI development set. Particularly, different vector sizes were explored, establishing the number of training epochs and window size during the global word vector model training. The best result was reached training during 30 epochs with a window size of 15 and a vector dimension of 30.

Regarding the network topology, different models were built using several types of layers and dimensions with the open source toolkit "currennt" [13]. All of them were tested on the development set and, finally, the best topology corresponded with a network with 2 hidden layers (BLSTM and LSTM) of 64 units each. This network architecture corresponds to that of Fig. 1.

Table 3 summarizes the results obtained using the SI experimental setting in terms of the different metrics presented in Section 3.2. The performance of the LSTM network is evaluated comparatively with respect to conditional random fields (CRF) and naïve Bayes (NB), which have shown to achieve very competitive results in CE [2, 14]. The experiments with CRF have been carried out using the Wapiti toolkit [15]. The best CRF models were obtained using the training algorithm *rprop-* and modelling dependencies between consecutive words.

Table 3: *Results on the speaker-independent test-set.*

	AUC	CER	NCE
Baseline	$--$	20.66	-
NB	84.4	16.54	-0.03
CRF	86.8	15.30	0.31
LSTM	88.3	14.58	0.35

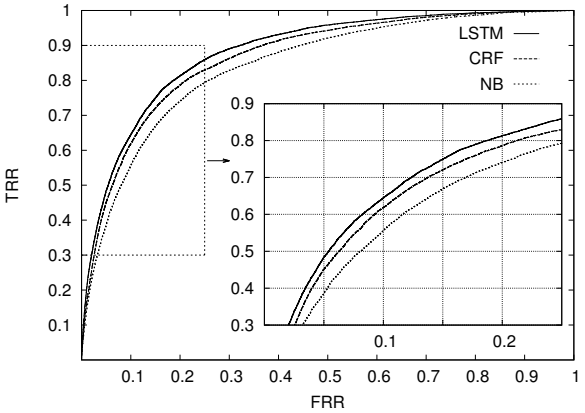

Figure 2: *ROC curves on the speaker-independent test-set.*

As can be seen, LSTM models significantly achieve the best performance in terms of AUC, CER and NCE. LSTM networks stated a relative improvement of 4.7% in terms of CER with respect CRF. This statement is confirmed in Fig. 2, where the LSTM network outperforms consistently (for all decision thresholds τ) the rest of the classifiers. For instance, given an FRR of 20%, the LSTM classifier is the only one which can provide a TRR above 80%.

The evaluation of the speaker-adaptation technique proposed in Section 2 is shown in Table 4. This table summarizes the results obtained by different experiments using the SD experimental setting. First, the non-adapted LSTM network used in the SI experiments was evaluated in order to establish a baseline performance. Second, starting from this non-adapted LSTM network, we trained a speaker-adapted LSTM network per speaker applying a few more training iterations using the BPTT algorithm with the data of each speaker. It is worth mentioning that the global word vector model was re-estimated so as to take into account the new speaker vocabulary along with the vocabulary from the SI experimental setting. Finally, a linear interpolation between both models (non-adapted and speaker-adapted) was evaluated. The optimal weights of interpolation were estimated using the development set.

Table 4: *Results on the speaker-dependent test-set.*

	AUC	CER	NCE
Baseline	$--$	21.83	-
CRF	87.4	15.82	0.33
CRF+spkadapt	87.6	15.56	0.34
LSTM	89.3	14.48	0.38
LSTM+spkadapt	89.6	14.42	0.39
LSTM+spkadapt (interpolated)	90.0	13.81	0.41

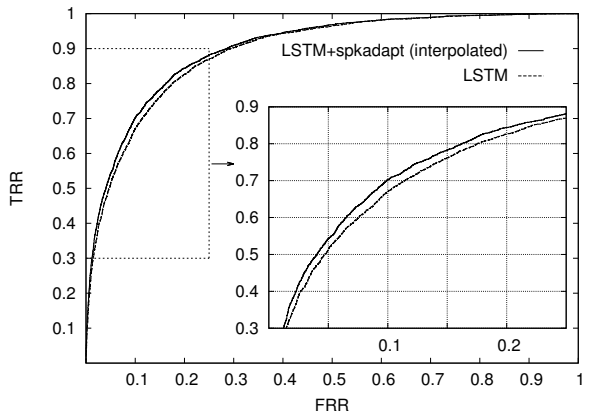

Figure 3: *ROC curves on the speaker-dependent test-set.*

As shown, the model interpolation results in the best model giving a relative improvement of 4.6% in CER with respect to the non-adapted model. This result is confirmed in Fig. 3, where the speaker-adapted model outperforms for any threshold τ their non-adapted counterpart. From our point of view, this final approximation has performed better because it has effectively prevented overfitting. This overfitting effect is usual when huge models such as LSTM networks are trained with scarce data, which is the case of adaptation to a single speaker.

For further analysis, Table 5 summarizes the performance of the interpolated model per speaker. In general, it can be stated that speaker-adapted models outperform their non-adapted counterparts in all cases in AUC, CER or both, except for speakers 4487 and 5248. For these two speakers, the adapted model achieves slightly worse CER. This could be produced by a particular vocabulary setting, quality of the adaptation data or a speaker-adapted system overfitting that could not be avoided with the interpolation.

Table 5: *Results on the speaker-dependent test-set per speaker.*

	AUC			CER		
SPK	¬Adapt	Adapt	R. I.	¬Adapt	Adapt	R. I.
644	88.7	90.1	1.6	17.8	16.6	6.7
778	88.9	89.7	0.9	12.7	11.3	11.4
1065	88.0	88.3	0.3	14.0	13.1	6.3
1085	87.3	87.3	0.0	13.8	13.7	0.7
1544	89.9	89.8	0.0	11.9	11.2	5.5
3318	91.0	92.4	1.5	13.1	12.3	6.5
3793	92.0	92.7	0.8	12.8	11.9	7.0
3798	92.3	92.9	0.7	11.1	9.2	16.3
3992	90.8	90.8	0.1	11.3	10.9	4.1
4034	88.2	89.1	1.0	13.2	13.1	0.7
4487	87.9	88.6	0.8	13.8	14.3	-3.7
4546	86.7	87.5	0.9	12.3	11.7	4.9
5136	91.5	92.5	1.2	13.8	11.7	15.3
5248	86.9	87.2	0.3	16.1	16.2	-0.6
5993	89.4	90.1	0.7	10.4	10.4	0.0
6353	88.7	90.3	1.9	17.1	15.2	11.3
7389	91.7	92.6	1.0	12.5	11.7	6.5
7597	90.0	90.4	0.5	13.3	13.1	1.6
8042	84.8	85.3	0.6	20.2	19.7	2.5
8356	86.7	87.2	0.6	15.5	15.0	3.1

4. Conclusions and Future Work

In this work, we have presented speaker-adapted confidence estimation using LSTM Networks. The use of LSTM Networks along with speaker-adaptation techniques constitutes a novelty in word confidence estimation. The results obtained over a publicly available dataset such as LibriSpeech confirm that LSTM networks improve state-of-the-art word confidence estimation models such as conditional random fields. Particularly, LSTM networks are able to produce relative reductions in CER of 4.7%. Moreover, the best speaker-adaptation technique presented is able to further reduce CER in 4.6%.

As future work, we plan to explore different word-embedding approaches. Also, we plan to study adaptation techniques for the (nearly) unsupervised case.

5. Acknowledgments

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 287755 (transLectures) and ICT Policy Support Programme (ICT PSP/2007-2013) as part of the Competitiveness and Innovation Framework Programme (CIP) under grant agreement no. 621030 (EMMA), the Spanish MINECO Active2Trans (TIN2012-31723) and EC FEDER Spanish MINECO MORE (TIN2015-68326-R) research projects.

6. References

[1] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, no. 4, pp. 455–470, 2005.

[2] A. Sanchis, A. Juan, and E. Vidal, "A Word-Based Naïve Bayes Classifier for Confidence Estimation in Speech Recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 565–574, 2012.

[3] I. Sanchez-Cortina, J. Andrés-Ferrer, A. Sanchis, and A. Juan, "Speaker-adapted confidence measures for speech recognition of video lectures," *Computer Speech & Language*, vol. 37, pp. 11–23, 2016.

[4] K. Kalgaonkar, C. Liu, Y. Gong, and K. Yao, "Estimating confidence scores on asr results using recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4999–5003.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

[7] P.-S. Huang, K. Kumar, C. Liu, Y. Gong, and L. Deng, "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, 2013, pp. 7413–7417.

[8] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[10] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5206–5210.

[11] M. del Agua, A. Giménez, N. Serrano, J. Andrés-Ferrer, J. Civera, A. Sanchis, and A. Juan, "The translectures-upv toolkit," in *Advances in Speech and Language Technologies for Iberian Languages*. Springer, 2014, pp. 269–278.

[12] F. Wessel, R. Schlüter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 3, pp. 288–298, 2001.

[13] F. Weninger, J. Bergmann, and B. Schuller, "Introducing currennt: The munich open-source cuda recurrent neural network toolkit," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 547–551, 2015.

[14] M. S. Seigel, "Confidence estimation for automatic speech recognition hypotheses," Ph.D. dissertation, Department of Engineering, University of Cambridge, 2013.

[15] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, July 2010, pp. 504–513. [Online]. Available: http://www.aclweb.org/anthology/P10-1052