ASR N-BEST FUSION NETS

Xinyue Liu, Mingda Li, Luoxin Chen, Prashan Wanigasekara,
Weitong Ruan, Haidar Khan, Wael Hamza, Chengwei Su

Amazon Alexa AI

ABSTRACT

Current spoken language understanding systems heavily rely on the best hypothesis (ASR 1-best) generated by automatic speech recognition, which is used as the input for downstream models such as natural language understanding (NLU) modules. However, the potential errors and misrecognition in ASR 1-best raise challenges to NLU. It is usually difficult for NLU models to recover from ASR errors without additional signals, which leads to suboptimal SLU performance. This paper proposes a fusion network to jointly consider ASR n-best hypotheses for enhanced robustness to ASR errors. Our experiments on Alexa data show that our model achieved 21.71% error reduction compared to baseline trained on transcription for domain classification.

1. INTRODUCTION

Current spoken language understanding (SLU) systems [1] usually rely on the best hypothesis from the automatic speech recognition (ASR) component. Any error in ASR 1-best hypothesis is propagated to downstream SLU components such as natural language understanding (NLU), leading to potential performance degradation. Without additional information, it is usually difficult for NLU models to recover from such ASR errors. Under this context, it is of great interest to investigate whether one could improve the SLU performance by exploiting additional ASR signals. In this paper, we explore approaches to utilize ASR n-best hypotheses for improved SLU tasks. In Alexa, we have observed some traffic with an incorrect ASR 1-best, but one can find a better hypothesis in the remaining ASR n-best hypotheses that are more similar to or identical to the transcription. Thus, there are considerable potentials of using ASR n-best hypotheses to boost the SLU performance once one could extract useful information from them. Unlike other contextual signals we usually exploit in the NLU world (*e.g.,* device type, device state, and customer embedding), ASR n-best is an ordered list of token sequences originating from the same spoken utterance by an ASR model. The n-best hypotheses are usually highly similar to each other, with only several different tokens. Moreover, the signals that reside in ASR n-best list are highly noisy and not always useful. In most cases, the ASR 1-best is the best interpretation, and the remaining n-best only adds more noise to the data. Given its high similarity and noisy nature, it is challenging to extract the informative patterns and signals that lie in ASR n-best for corresponding downstream tasks.

To better capture the correlation among n-best, we propose to use a pipelined framework. It first employs a shared hypothesis encoder to embed each hypothesis (Section 2.2). Then we design a fusion network called n-Best Fuser to exploit the embeddings as a coherent feature map and convert them into a single fused feature representation (Section 2.3).

To evaluate the effectiveness of the proposed methods, we focus on the task of domain classification in this paper, where each utterance in the SLU data set is associated with a predefined *domain*. The n-best fusion network exhibits significant improvement over baseline domain classifier.

2. ASR N-BEST FUSION NETS

Given an utterance, its ASR n-best hypotheses are the output of the ASR component in the format of $\mathcal{A} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$, where each hypothesis sentence can be denoted as $\boldsymbol{a}_i = (w_1^{(i)}, \ldots, w_T^{(i)})$, and each w is a word or Byte Pair Encoding (BPE) token. Our framework estimates $\mathcal{P}_{\boldsymbol{\theta}}(d_i|\mathcal{A}^{(x)})$ that maps each utterance $\mathcal{A}^{(x)}$ to a probability distribution over domains $\boldsymbol{D} = (d_1, \ldots, d_C)$, where $\boldsymbol{\theta}$ is the model parameters to be learned. And the framework learning can be formalized as minimizing the following cross entropy loss: $\underset{\boldsymbol{\theta}}{\text{Minimize}} - \sum_{x=1}^{N} \sum_{i=1}^{C} y_i^{(x)} \log\left([\mathcal{P}_{\boldsymbol{\theta}}(d_i|\mathcal{A}^{(x)})]\right)$ where $y_i^{(x)}$ is an indicator for i-th class for utterance $\mathcal{A}^{(x)}$, N is the number of data, and C is the number of classes (domains).

2.1. BPE and Word Embedding

We apply byte pair encoding (BPE) [2] on the corpus of ASR n-best hypotheses during tokenization. BPE works by identifying high-frequency sub-token in the corpus of words and replace them with tokens that are not within the original data. We vectorize each BPE converted token with a token embedding layer of embedding size e. After the embedding layer, the i-best ASR hypothesis is embedded as $\mathcal{E}_i = (\boldsymbol{e}_1^{(i)}, \ldots, \boldsymbol{e}_T^{(i)})$, where $\boldsymbol{e}_j^{(i)} \in \mathbb{R}^e$ is the word embedding vector for $w_j^{(i)}$.

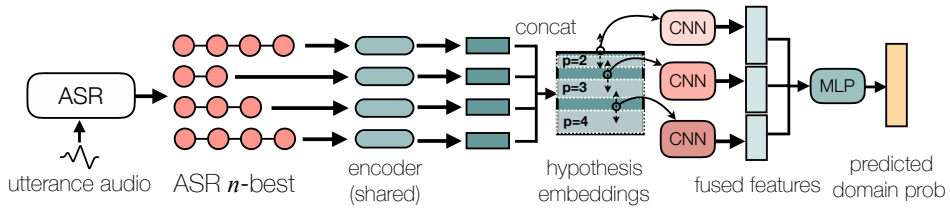


Fig. 1: The illustration of the n-best fusion network. The ASR n-best is fed to a shared Bi-LSTM encoder to obtain the hypothesis embeddings. Then we apply multiple 1D CNNs with various kernel size ($p \in \mathbb{N}^+$) on the concatenated embeddings to acquire the fused features. A final task-specific MLP is employed to generate the prediction from the concatenated fused features.

2.2. Hypothesis Encoder

We choose bi-directional RNN [3, 4] encoder to obtain the hypothesis embedding from its token embedding. The encoder takes $\mathcal{E}$ as the input, at each time step t, it follows

$$\overrightarrow{\boldsymbol{h}}_t = f_h(W_{\overrightarrow{eh}}\boldsymbol{e}_t + W_{\overrightarrow{hh}}\overrightarrow{\boldsymbol{h}}_{t-1} + \overrightarrow{\boldsymbol{b}}) \quad (1)$$

$$\overleftarrow{\boldsymbol{h}}_t = f_h(W_{\overleftarrow{eh}}\boldsymbol{e}_t + W_{\overleftarrow{hh}}\overleftarrow{\boldsymbol{h}}_{t+1} + \overleftarrow{\boldsymbol{b}}) \quad (2)$$

where $\leftarrow$ and $\rightarrow$ denotes the direction of the parameter, $\boldsymbol{h}$s are the hidden states, and Ws are the weight matrices in the network. As to the choice of hidden layer function $f_h(\cdot)$, we adopt the widely used LSTM cell introduced in [5] due to its superior performance and its capability of capturing long term dependencies. We follow [6] to implement the $f_h(\cdot)$ of LSTM cell, detailed formulations are omitted here due to the space limitation. For each hypothesis $\mathcal{E} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T)$, we regard the concatenation of the two hidden states at the last step as the final encoding of it, denoted as $\boldsymbol{c} = \begin{pmatrix} \overrightarrow{\boldsymbol{h}}_T \\ \overleftarrow{\boldsymbol{h}}_T \end{pmatrix}$.

2.3. n-Best Fuser

We feed the n-best hypotheses into the same encoder as demonstrated above to obtain the n-best encoding $\mathcal{C} = (\boldsymbol{c}_1, \ldots, \boldsymbol{c}_n)$. Capturing the semantic pattern within $\mathcal{C}$ is a crucial step for our downstream NLU tasks. We could represent the n-best encoding as a matrix $C \in \mathbb{R}^{2m \times n}$, where m is the size of hidden states of our encoder. Although C looks like a 2-D feature map and a conventional CNN [7] seems promising on extracting patterns from it, it is better to be regarded as a 1-D feature of length n and $2m$ channels as inspired by previous work [8]. It is worth noting that each column of C corresponds to an ASR hypothesis's feature encoding. Although all n hypotheses are from the same spoken utterance, they are considered separate signal origins. CNNs with a 2-D convolutional layer would be inappropriate since the first axis of C concerns no spatial relationship. Thus, we propose to use CNNs with a 1-D convolutional layer and a 1-D pooling layer on C for feature extraction.

Let $\boldsymbol{c}^{(i)} \in \mathbb{R}^n$ be the i-th channel (row) of C a 1-D convolutional layer works by applying a filter $W_f^{(i)} \in \mathbb{R}^{k \times p}$ for each channel of C as follow, where p is the kernel size and k is the number of output channels in the feature maps.

$$\tilde{\boldsymbol{c}}_q = \sum_{i=1}^{2m} \sigma(W_f^{(i)} \boldsymbol{c}_{q:q+p}^{(i)} + \boldsymbol{b}_f) \quad (3)$$

where $\boldsymbol{c}_{q:q+p}^{(k)}$ denotes the slicing of $\boldsymbol{c}^{(k)}$ from its q-th element to $(q+p-1)$-th element, and $\sigma(\cdot)$ is the activation function. The convoluted feature maps could be denoted as $\tilde{C} = (\tilde{\boldsymbol{c}}_1, \tilde{\boldsymbol{c}}_{1+s}, \tilde{\boldsymbol{c}}_{1+2s}, \ldots, \tilde{\boldsymbol{c}}_{1+zs}) \in \mathbb{R}^{k \times (\lfloor (n-p)/s \rfloor + 1)}$, where $s \in \mathbb{N}^+$ is the size of strides. A pooling layer on $\tilde{C}$ can additionally aggregate the convoluted feature maps. In the end, we flatten the aggregated feature maps of all k channels and connect it with an MLP [9] to yield a probability distribution over all domains.

To best capture the complex pattern lies in ASR n-best, we do employs multiple CNNs with different filter (kernel) size $p \in \{2, 3, \ldots, n-1\}$ as shown in Figure 1, where each CNN captures the p-th order correlation among the n-best. The flatten output of all CNNs are concatenated before it goes to the MLP for the final output.

3. EXPERIMENTS

3.1. Experiment Setup

We use 8.7 million utterances from 22 domains for training and validation. We use other 900k utterances for testing. Each utterance goes through the same ASR model to obtain its n-best hypotheses. The following methods are implemented and tested for domain classification performance comparison.

Oracle: A BiLSTM model trained and tested on hand-picked ASR hypothesis with lowest edit distance to the transcription.

Baseline: BiLSTM trained on manual transcription, and tested on ASR 1-best hypothesis.

1-best BiLSTM [10]: BiLSTM trained and tested on ASR 1-best hypothesis.

n-best Random: This is a lower bound for n-best models. It randomly selects a hypothesis out of the n-best list for each utterance and passes it to the baseline BiLSTM DC model we trained on transcription.

Table 1: Evaluation on full test set and ASR error set (all numbers are in %), all numbers are normalized to reflect the relative improvement (degradation) w.r.t to the baseline.

Model	Full Test Set		ASR error set		ASR-NLU Error Set
	DCER (↓)	Macro F1 (↑)	DCER (↓)	Macro F1 (↑)	Relative DCER Reduction (↑)
oracle	-1.13	+1.27	-2.76	+3.06	20.17
baseline	0	0	0	0	0
1-best [10]	-0.23	**+0.47**	-1.18	+1.49	17.14
n-best random	+2.66	-2.96	+0.24	-0.09	15.25
n-best mean pooling [11, 12]	+0.48	-0.69	-1.44	+1.75	21.08
n-best weighted pooling [11, 12]	-0.08	+0.08	-1.48	+1.88	19.96
n-best local attention [13]	+0.20	-0.40	-1.58	+1.76	20.51
n-best global attention [13]	+0.44	-0.47	-1.63	+1.86	21.59
n-best fuser [this paper]	**-0.24**	+0.14	**-1.94**	**+2.14**	**21.71**

n-best Pooling [11, 12]: This method employs the same encoder to obtain n-best hypotheses embedding. Then it applies a pooling layer to merge the embedding into a dense vector representation. We compare two pooling strategies: mean pooling and weighted pooling.

n-best Attention [13]: It is the same architecture as n-best Pooling, except we use more generalized attention methods for pooling purpose. We compare two attention mechanisms: local attention and global attention. In local attention, each hypothesis embedding is fed to a shared attention layer separately to obtain an attention score; while in global attention, all hypotheses' embeddings are concatenated and fed to an attention layer to obtain n attention scores at once.

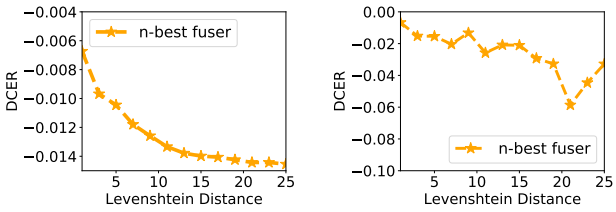
n-best Fuser: Our proposed model as in Sec. 2.3.

3.2. Training

We randomly hold out 0.9 million training data for the experiments as the validation (development) set. The hypothesis encoder in all compared models is a 2-layer BiLSTM with 256 hidden units and the dropout rate 0.2. The output MLP also has 256 hidden units and ReLU activation. We employed Adam optimization algorithm [14] with a triangle learning rate scheduler [15]. The learning rate ranges from 1^{-4} to 0.02 with cycle length fixed as 71000 and an incremental fraction of 0.05. The mini-batch size is 128. All models are trained for at least 20 epochs, and validation is performed every 3000 updates. The best model on the validation set is saved for testing..

3.3. Result

We employ domain classification error rate (DCER) and macro F1 as the metrics in this study, where DCER is defined as $(1 - \text{accracy})$ and macro F1 follows the *F1 of averages* definition in [16]. Table 1 shows the evaluation results on all compared models listed in Sec. 3.1, where we use ↑ to denote "larger is better" and use ↓ to denote "smaller is better". The n-best fusion nets achieved 0.24% absolute improvement in terms of DCER, and its absolute improvement in terms of Macro F1 is 0.14% Given the test set's size, the improvement indicates our n-best fusion nets can recover a significant number of ASR errors by exploiting the ASR n-best.



(a) Accumulative (b) Non-accumulative

Fig. 2: DCER improvement on ASR error set by only including utterances with the corresponding Levenshtein edit distance (X-axis) between the ASR 1-best and the transcription.

3.4. ASR Error Reduction

It is more important to evaluate the cases where ASR 1-best makes mistakes to understand the contribution made by our proposed fusion network. We collected 173k utterances in the test set where the ASR 1-best hypothesis is different from the manual transcription, and we refer to this test set as ASR Error Set. We report the evaluation results on ASR error set in Table 1. On this set, the n-best fusion network achieved 1.94% absolute improvement over the baseline model in terms of DCER, and its relative performance gain of macro F1 is 2.14%, indicating its superb capability of alleviating the negative impact of ASR errors on DC task.

To study the ASR error effects in finer granularity, we il-

Table 2: DCER (%) comparison for utterances with different number of ASR hypotheses (k).

k	baseline	n-best fuser
1	0	-0.01
2	0	-0,11
3	0	-0.15
4	0	-0.19
5	0	-0.29

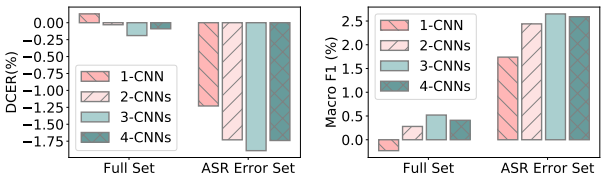
lustrate the DCER improvement made by n-best fuser by selecting utterances with various Levenshtein edit distance [17] between their ASR 1-best hypothesis and transcription in Figure 2. Figure 2a illustrates the DCER (the Y-axis) on all utterances with edit distance up to ℓ (the X-axis) between the ASR 1-best and transcription. Figure 2b shows the case where only utterances with distance within $[\ell, \ell + 1)$ are considered. One can tell that the DCER improvement is small when the edit distance is relatively low, indicating the impact of such minor ASR error is small, and the n-best fuser cannot help much. As the edit distance increases, we observe more significant DCER improvement made by n-best fuser against the baseline. It demonstrates that the baseline model cannot recover from severe ASR errors as the ASR 1-best is very different from the ground-truth. For such cases, n-best fuser effectively alleviates the impact of ASR errors by collectively considering ASR n-best hypotheses.

3.5. NLU-ASR Error Reduction

As shown in Figure 2, NLU models can recover from minor ASR errors. It is more of our interest to study the effects of n-best on the utterances where the ASR misrecognition does cause NLU errors (*i.e.*, wrong domain classifications made by the baseline model in our experiments). We selected all such cases from the ASR error set we tested above, which end up with a test set of 37.1k utterances, and we regard it as the NLU-ASR Error set. We test all compared models on NLU-ASR Error test set and report their DCER reduction *w.r.t.* the baseline model in Table 1. Among these 37.1k utterances, we observed 21.71% DCER reduction achieved by n-best fuser, outperforming all other compared methods, including the oracle.

3.6. ASR Uncertainty (Ambiguity)

The ASR generally produces fewer hypotheses when it is confident. Table 2 shows the DCER of baseline and n-best fuser for utterances with different number of output hypotheses ($k = 1, 2, 3, 4, 5$). The error reduction increases while k gets bigger, indicating that the n-best fuser degenerates slower than the baseline model while the ASR uncertainty becomes high.



(a) DCER (↓) (b) Macro F1 (↑)

Fig. 3: DCER and Macro F1 comparison of different number of CNNs used in n-best fuser.

3.7. Effect of Multiple CNNs

To demonstrate the effect of employing multiple CNNs, we illustrate the performance comparison of n-best fuser with a different number of CNNs in Figure 3. In our experiments, the fuser with 3 CNNs (with kernel sizes $2, 3, 4$ respectively) achieved the best results. Further, incorporating an additional CNN with kernel size 5 does not improve the performance but degenerates the DCER and macro F1.

4. RELATED WORK

[12, 18] proposed a variation of RNNs work on the topological sort of lattice graph to get better encoding of a spoken utterance. Although lattice potentially contains more structural information than ASR n-best, it requires significant modeling change to utilize lattice. [19] shows a two-step approach to generate the best path from confusion network to improve slot filling task. A more recent work [20] studies the approach of using the confusion network in a neural dialogue state tracker (DST), where the authors propose an attentional confusion network encoder that can be used in any DST. On the other hand, there has been work on re-scoring ASR n-best by exploring the morphological, lexical, and syntactic features [21, 22]. [23] shows a joint model for ASR error correction and language understanding tasks such as dialog act prediction and slot filling. Our work differs in that we are not attempting to correct the ASR error by re-aligning the hypotheses but considering the n-best directly for downstream NLU tasks jointly..

5. CONCLUSION

We have proposed a fusion network to take advantage of ASR n-best hypotheses for improving the NLU task performance. The proposed n-best fuser provides a robust representation of the n-best hypotheses accordingly based upon the downstream NLU tasks. The empirical study demonstrates that our method can alleviate the impact of ASR errors on NLU tasks by collectively exploiting the ASR n-best hypotheses.

6. REFERENCES

[1] Gokhan Tur and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.

[2] Philip Gage, "A new algorithm for data compression," *C Users Journal*, vol. 12, no. 2, pp. 23–38, 1994.

[3] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[4] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

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

[6] Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[8] Yoon Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[9] Matt W Gardner and SR Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[10] Zhiheng Huang, Wei Xu, and Kai Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[11] Mingda Li, Weitong Ruan, Xinyue Liu, Luca Soldaini, Wael Hamza, and Chengwei Su, "Improving spoken language understanding by exploiting asr n-best hypotheses," *arXiv preprint arXiv:2001.05284*, 2020.

[12] Faisal Ladhak, Ankur Gandhe, Markus Dreyer, Lambert Mathias, Ariya Rastrow, and Björn Hoffmeister, "Latticernn: Recurrent neural networks over lattices.," in *Interspeech*, 2016, pp. 695–699.

[13] Minh-Thang Luong, Hieu Pham, and Christopher D Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[14] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] Leslie N Smith, "Cyclical learning rates for training neural networks," in *WACV*. IEEE, 2017, pp. 464–472.

[16] Juri Opitz and Sebastian Burst, "Macro f1 and macro f1," *arXiv preprint arXiv:1911.03347*, 2019.

[17] Vladimir I Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, 1966, vol. 10, pp. 707–710.

[18] Jinsong Su, Zhixing Tan, Deyi Xiong, Rongrong Ji, Xiaodong Shi, and Yang Liu, "Lattice-based recurrent neural network encoders for neural machine translation," in *AAAI*, 2017.

[19] Dilek Hakkani-Tür, Frédéric Béchet, Giuseppe Riccardi, and Gokhan Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.

[20] Vaishali Pal, Fabien Guillot, Jean-Michel Renders, and Laurent Besacier, "Modeling asr ambiguity for dialogue state tracking using word confusion networks," *arXiv preprint arXiv:2002.00768*, 2020.

[21] Haşim Sak, Murat Saraçlar, and Tunga Güngör, "Discriminative reranking of asr hypotheses with morpholexical and n-best-list features," in *ASRU*. IEEE, 2011, pp. 202–207.

[22] Fuchun Peng, Scott Roy, Ben Shahshahani, and Françoise Beaufays, "Search results based n-best hypothesis rescoring with maximum entropy classification," in *ASRU*. IEEE, 2013, pp. 422–427.

[23] Yue Weng, Sai Sumanth Miryala, Chandra Khatri, Runze Wang, Huaixiu Zheng, Piero Molino, Mahdi Namazifar, Alexandros Papangelis, Hugh Williams, Franziska Bell, et al., "Joint contextual modeling for asr correction and language understanding," *arXiv preprint arXiv:2002.00750*, 2020.