Learning from Past Mistakes: Improving Automatic Speech Recognition Output via Noisy-Clean Phrase Context Modeling

Prashanth Gurunath Shivakumar[a,*], Haoqi Li[a], Kevin Knight[b], Panayiotis Georgiou[a,**]

[a] *Signal Processing for Communication Understanding & Behavior Analysis (SCUBA) Lab, University of Southern California, Los Angeles, USA*
[b] *Information Sciences Institute, University of Southern California, Los Angeles, USA*

Abstract

Automatic speech recognition (ASR) systems lack joint optimization during decoding over the acoustic, lexical and language models; for instance the ASR will often prune words due to acoustics using short-term context, prior to rescoring with long-term context. In this work we model the automated speech transcription process as a noisy transformation channel and propose an error correction system that can learn from the aggregate errors of all the independent modules constituting the ASR. The proposed system can exploit long-term context using a neural network language model and can better choose between existing ASR output possibilities as well as re-introduce previously pruned and unseen (out-of-vocabulary) phrases. The system provides significant corrections under poorly performing ASR conditions without degrading any accurate transcriptions. The proposed system can thus be independently optimized and post-process the output of even a highly optimized ASR. We show that the system consistently provides improvements over the baseline ASR. We also show that it performs better when used on out-of-domain and mismatched test data and under high-error ASR conditions. Finally, an extensive analysis of the type of errors corrected by our system is presented.

Keywords: Error Correction, Statistical Phrase-based Context Modeling, Noisy Channel Estimation, Error Analysis

*Corresponding author
**Principal corresponding author
Email addresses: pgurunat@usc.edu (Prashanth Gurunath Shivakumar), haoqili@usc.edu (Haoqi Li), knight@isi.usc.edu (Kevin Knight), georgiou@sipi.usc.edu (Panayiotis Georgiou)

1. Introduction

Due to the complexity of human language and quality of speech signals, improving performance of automatic speech recognition (ASR) is still a challenging task. The traditional ASR comprises of three conceptually distinct modules: acoustic modeling, dictionary and language modeling. Three modules are fairly independent of each other in research and operation.

In terms of acoustic modeling, Gaussian Mixture Model (GMM) based Hidden Markov Model (HMM) systems [1, 2] were a standard for ASR for a long time and are still used in some of the current ASR systems. Lately, advances in Deep Neural Network (DNN) led to the advent of Deep Belief Networks (DBN) and Hybrid DNN-HMM [3, 4], which basically replaced the GMM with a DNN and employed a HMM for alignments. Deep Recurrent Neural Networks (RNN), particularly Long Short Term Memory (LSTM) Networks replaced the traditional DNN and DBN systems [5]. Connectionist Temporal Classification (CTC) [6] proved to be effective with the ability to compute the alignments implicitly under the DNN architecture, thereby eliminating the need of GMM-HMM systems for computing alignments.

The research efforts for developing efficient dictionaries or lexicon have been mainly in terms of pronunciation modeling. Pronunciation modeling was introduced to handle the intra-speaker variations [7, 8], non-native accent variations [7, 8], speaking rate variations found in conversational speech [8] and increased pronunciation variations found in children's speech [9]. Various linguistic knowledge and data-derived phonological rules were incorporated to augment the lexicon.

Research efforts in language modeling share those of the Natural Language Processing (NLP) community. By estimating the distribution of words, statistical language modeling (SLM), such as n-gram, decision tree models [10], linguistically motivated models [11] amount to calculating the probability distribution of different linguistic units, such as words, sentences, and whole documents [12]. Recently, Deep Neural Network based language models [13, 14, 15] have also shown success in terms of both perplexity and word error rate.

Very recently, state-of-the-art ASR systems are employing end-to-end neural network models such as sequence-to-sequence [16] in an encoder-decoder architecture. The system is trained end-to-end from acoustic features as input to predict the phonemes or characters [17, 18]. The system can be viewed as an integration of acoustic and lexicon pronunciation models. The state-of-the-art performance can be attributed towards the joint training (optimization) between the acoustic model and the lexicon models (end-to-end) enabling them to overcome the short-comings of the former independently trained models.

Several research efforts were carried out for error correction using post-processing techniques. Much of the effort involves user input used as a feedback mechanism to learn the error patterns [19, 20]. Other work employs multi-modal signals to correct the ASR errors [21, 20]. There has been relatively less work dealing with ASR error correction. In [22], a word-based error correction technique was proposed. The technique demonstrated the ability to model the ASR

as a noisy channel. In [23], similar technique was applied to a syllable-to-syllable channel model along with maximum entropy based language modeling.

Our Contribution: The scope of this paper is to evaluate whether subsequent transcription corrections can take place, on top of a highly optimized ASR. We hypothesize that our system can correct the errors by (i) re-scoring lattices, (ii) recovering pruned lattices, (iii) recovering unseen phrases, (iv) providing better recovery during poor recognitions, (v) providing improvements under all acoustic conditions, (vi) handling mismatched train-test conditions, (vii) exploiting longer contextual information and (viii) text regularization. We target to satisfy the above hypotheses by proposing a Noisy-Clean Phrase Context Model (NCPCM). We introduce context of past errors of an ASR system, that consider all the automated system noisy transformations. These errors may come from any of the ASR modules or even from the noise characteristics of the signal. Using these errors we learn a noisy channel model, and apply it for error correction of ASR. Compared to the earlier efforts in ASR error correction [22, 23], our work differs in the following aspects: (1) we propose a phrase-based noisy channel modeling for error correction; (2) we utilize multiple hypotheses of ASR; (3) we employ the state-of-the-art neural network based language models which enables us to exploit long contexts; (4) we use minimum error rate training (MERT) to optimize and adapt to the domain data; (5) we present the effectiveness of the proposed method on an out-of-domain task; (6) we target specific types of error corrections to be handled by our system and provide evaluations.

Further, our work is different from discriminative training of language [24] and acoustic [25] models, which are trained directly to optimize the word error rate using the reference transcripts. Our method concentrates mostly on correcting the aggregated errors of independent modules of ASR using long term context.

Additionally, our proposed system comes with several advantages: (1) the system could potentially be trained without an ASR by creating a phonetic model of corruption and emulating an ASR decoder on generic text corpora, (2) the system rapidly adapts to new language, i.e., the unseen words during training, by creating contextual transformation for rapidly changing language.

The rest of the paper is organized as follows: Section 2 presents various hypotheses and discusses the different types of errors we expect to model. Section 3 elaborates on the proposed technique and Section 4 describes the experimental setup and the databases employed in this work. Results and discussion are presented in Section 5 and we finally conclude and present future research directions in Section 6.

2. Hypotheses

In this section we analytically present cases that we hypothesize the proposed system could help with. In all of these the errors of the ASR may stem from realistic constraints of the decoding system and pruning structure, while the proposed system could exploit very long context to improve the ASR output.

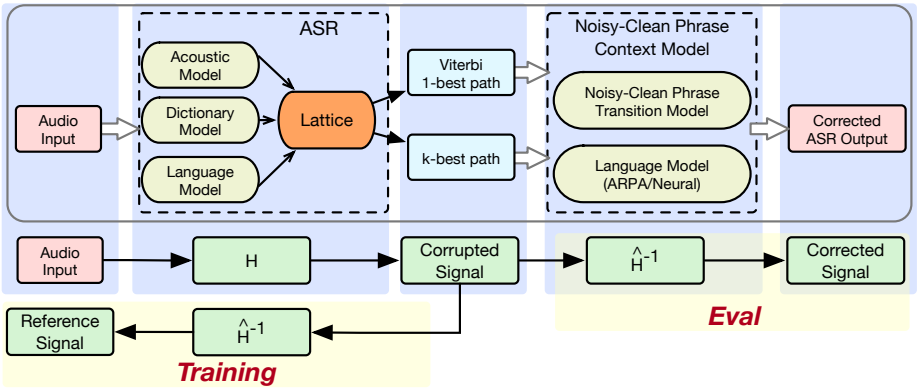


Figure 1: Overview of NCPCM

2.1. Re-scoring Lattices

1. *"I was born in nineteen ninety three in Iraq"*
2. *"I was born in nineteen ninety three in eye rack"*
3. *"I was born in nineteen ninety three in I rack"*

Example 1

In Example 1, all the three samples have the same phonetic transcription - "*ay . w aa z . b ao r n . ih n . n ay n t iy n . n ay n t iy . th r iy . ih n . ay . r ae k*". Let us assume sample 1 is the correct transcription. Since all the three examples have the same phonetic transcription, this makes them indistinguishable by the acoustic model. The language model is likely to down-score the sample 3. It is likely that sample 2 will score higher than sample 1 by a short context LM (e.g. bi-gram or 3-gram) i.e., "in" might be followed by "eye" more frequently than "Iraq" in the training corpora. This will likely result in an ASR error. Thus, although the oracle WER can be zero, the output WER is likely going to be higher due to LM choices.

> ***Hypothesis A:*** An ideal error correction system can select correct options from the existing lattice.

2.2. Recovering Pruned Lattices

A more severe case of Example 1 would be that the word "Iraq" was pruned out of the output lattice during decoding. This is often the case when there are memory and complexity constraints in decoding large acoustic and language models, where the decoding beam is a restricting parameter. In such cases, the word never ends up in the output lattice. Since the ASR is constrained to pick over the only existing possible paths through the decoding lattice, an error is enforced in the final output.

Hypothesis B: An ideal error correction system can generate words or phrases that were erroneously pruned during the decoding process.

2.3. Recovery of Unseen Phrases

On the other hand, an extreme case of Example 1 would be that the word "Iraq" was never seen in the training data (or is out-of-vocabulary), thereby not appearing in the ASR lattice. This would mean the ASR is forced to select among the other hypotheses even with a low confidence (or output an unknown, $< unk >$, symbol) resulting in a similar error as before. This is often the case due to the constant evolution of human language or in the case of a new domain. For example, names such as "Al Qaeda" or "ISIS" were non-existent in our vocabularies a few years ago.

Hypothesis C: An ideal error correction system can generate words or phrases that are out of vocabulary (OOV) and thus not in the ASR output.

2.4. Better Recovery during Poor Recognitions

An ideal error correction system would provide more improvements for poor recognitions from an ASR. Such a system could potentially offset for the ASR's low performance providing consistent performance over varying audio and recognition conditions. In real-life conditions, the ASR often has to deal with varying level of "mismatched train-test" conditions, where relatively poor recognition results are commonplace.

Hypothesis D: An ideal error correction system can provide more corrections when the ASR performs poorly, thereby offsetting ASR's performance drop (e.g. during mismatched train-test conditions).

2.5. Improvements under all Acoustic Conditions

An error correction system which performs well during tough recognition conditions as per *Hypothesis* 2.4 is no good if it degrades the good recognitions. Thus, in addition to our *Hypothesis* 2.4, an ideal system would cause no degradations on good recognitions simultaneously providing more improvements, handling poor recognitions from an ASR. Such a system can be hypothesized to always improve upon and provide benefits over any ASR system including state-of-the-art recognition systems. An ideal system would provide improvements over the entire spectrum of ASR performance (WER).

Hypothesis E: An ideal error correction system can not only provide improvements during poor recognitions, but also does not degrade output when the ASR performs well.

2.6. Adaptation

We hypothesize that the proposed system would help in adaptation over mismatched conditions. The mismatch could manifest in terms of acoustic conditions and lexical constructs. The acoustic adaptation can be seen as a consequence of *Hypothesis* 2.4 & 2.5. In addition, the proposed model is capable of capturing patterns of language use manifesting in specific speaker(s) and domain(s). Such a system could eliminate the need of retraining the ASR model for mismatched environments.

> ***Hypothesis F:*** An ideal error correction system can aid in mismatched train-test conditions.

2.7. Exploit Longer Context

- *"**Eyes** melted, when he placed his hand on her shoulders."*
- *"**Ice** melted, when he placed it on the table."*

Example 2

The complex construct of human language and understanding enables recovery of lost or corrupted information over different temporal resolutions. For instance, in the above Example 2, both the phrases, "Eyes melted, when he placed" and "Ice melted, when he placed" are valid when viewed within its shorter context and have identical phonetic transcriptions. The succeeding phrases, underlined, help in discerning whether the first word is "Eyes" or "Ice". We hypothesize that an error correction model capable of utilizing such longer contexts is beneficial. As new models for phrase based mapping, such as sequence to sequence models [16], become applicable this becomes even more possible and desirable.

> ***Hypothesis G:*** An ideal error correction system can exploit longer context than the ASR for better corrections.

2.8. Regularization

1.
   - *"I guess 'cause I went on a I went on a ..."*
   - *"I guess because I went on a I went on a ..."*
2.
   - *"i was born in nineteen ninety two"*
   - *"i was born in 1992"*
3.
   - *"i was born on nineteen twelve"*
   - *"i was born on 19/12"*

Example 3

As per the 3 cases shown in Example 3, although both the hypotheses for each of them are correct, there are some irregularities present in the language syntax. Normalization of such surface form representation can increase readability and usability of output. Unlike traditional ASR, where there is a need to explicitly program such regularizations, our system is expected to learn, given appropriate training data, and incorporate regularization into the model.

Hypothesis H: An ideal error correction system can be deployed as an automated text regularizer.

3. Methodology

The overview of the proposed model is shown in Figure 1. In our paper, the ASR is viewed as a noisy channel (with transfer function H), and we learn a model of this channel, $\widehat{H}^{-1}$ (estimate of inverse transfer function H^{-1}) by using the corrupted ASR outputs (equivalent to signal corrupted by H) and their reference transcripts. Later on, we use this model to correct the errors of the ASR.

The noisy channel modeling mainly can be divided into word-based and phrase-based channel modeling. We will first introduce previous related work, and then our proposed Noisy-Clean Phrase Context Model (NCPCM).

3.1. Previous related work

3.1.1. Word-based Noisy Channel Modeling

In [22], the authors adopt word-based noisy channel model borrowing ideas from a word-based statistical machine translation developed by IBM [26]. It is used as a post-processor module to correct the mistakes made by the ASR. The word-based noisy channel modeling can be presented as:

$$\begin{aligned}\hat{W} &= \arg\max_{W_{\text{clean}}} P(W_{\text{clean}}|W_{\text{noisy}}) \\ &= \arg\max_{W_{\text{clean}}} P(W_{\text{noisy}}|W_{\text{clean}})P_{\text{LM}}(W_{\text{clean}})\end{aligned}$$

where $\hat{W}$ is the corrected output word sequence, $P(W_{\text{clean}}|W_{\text{noisy}})$ is the posterior probability, $P(W_{\text{noisy}}|W_{\text{clean}})$ is the channel model and $P_{\text{LM}}(W_{\text{clean}})$ is the language model. In [22], authors hypothesized that introducing many-to-one and one-to-many word-based channel modeling (referred to as fertility model) could be more effective, but was not implemented in their work.

3.1.2. Phrase-based Noisy Channel Modeling

Phrase-based systems were introduced in application to phrase-based statistical translation system [27] and were shown to be superior to the word-based systems. Phrase based transformations are similar to word-based models with the exception that the fundamental unit of observation and transformation is a phrase (one or more words). It can be viewed as a super-set of the word-based [26] and the fertility [22] modeling systems.

3.2. Noisy-Clean Phrase Context Modeling

We extend the ideas by proposing a complete phrase-based channel modeling for error correction which incorporates the many-to-one and one-to-many as well as many-to-many words (phrase) channel modeling for error-correction. This also allows the model to better capture errors of varying resolutions made by the ASR. As an extension, it uses a distortion modeling to capture any re-ordering of phrases during error-correction. Even though we do not expect big benefits from the distortion model (i.e., the order of the ASR output is usually in accordance with the transcript), we include it in our study for examination. It also uses a word penalty to control the length of the output. The phrase-based noisy channel modeling can be represented as:

$$\begin{aligned}\hat{p} &= \arg\max_{p_{\text{clean}}} P(p_{\text{clean}}|p_{\text{noisy}}) \\ &= \arg\max_{p_{\text{clean}}} P(p_{\text{noisy}}|p_{\text{clean}}) P_{\text{LM}}(p_{\text{clean}}) w_{\text{length}}(p_{\text{clean}})\end{aligned}$$

where $\hat{p}$ is the corrected sentence, p_{clean} and p_{noisy} are the reference and noisy sentence respectively. $w_{\text{length}}(p_{\text{clean}})$ is the output word sequence length penalty, used to control the output sentence length, and $P(p_{\text{noisy}}|p_{\text{clean}})$ is decomposed into:

$$P(p^I_{\text{noisy}}|p^I_{\text{clean}}) = \prod_{i=1}^{I} \phi(p^i_{\text{noisy}}|p^i_{\text{clean}}) D(start_i - end_{i-1})$$

where $\phi(p^i_{\text{noisy}}|p^i_{\text{clean}})$ is the phrase channel model or phrase translation table, p^I_{noisy} and p^I_{clean} are the sequences of I phrases in noisy and reference sentences respectively and i refers to the i^{th} phrase in the sequence. $D(start_i - end_{i-1})$ is the distortion model. $start_i$ is the start position of the noisy phrase that was corrected to the i^{th} clean phrase, and end_{i-1} is the end position of the noisy phrase corrected to be the $i - 1^{th}$ clean phrase.

3.3. Our Other Enhancements

In order to effectively demonstrate our idea, we employ (i) neural language models, to introduce long term context and justify that the longer contextual information is beneficial for error corrections; (ii) minimum error rate training (MERT) to tune and optimize the model parameters using development data.

3.3.1. Neural Language Models

Neural network based language models have been shown to be able to model higher order n-grams more efficiently [13, 14, 15]. In [23], a more efficient language modeling using maximum entropy was shown to help in noisy-channel modeling of a syllable-based ASR error correction system.

Incorporating such language models would aid the error-correction by exploiting the longer context information. Hence, we adopt two types of neural network language models in this work. (i) Feed-forward neural network which is

trained using a sequence of one-hot word representation along with the specified context [28]. (ii) Neural network joint model (NNJM) language model [29]. This is trained in a similar way as in (i), but the context is augmented with noisy ASR observations with a specified context window. Both the models employed are feed-forward neural networks since they can be incorporated directly into the noisy channel modeling. Noise Contrastive Estimation was used to handle the large vocabulary size output.

3.3.2. Minimum Error Rate Training (MERT)

One of the downsides of the noisy channel modeling is that the model is trained to maximize the likelihood of the seen data and there is no direct optimization to the end criteria of WER. MERT optimizes the model parameters (in our case weights for language, phrase, length and distortion models) with respect to the desired end evaluation criterion. MERT was first introduced in application to statistical machine translation providing significantly better results [30]. We apply MERT to tune the model on a small set of development data.

4. Experimental Setup

4.1. Database

For training, development, and evaluation, we employ Fisher English Training Part 1, Speech (LDC2004S13) and Fisher English Training Part 2, Speech (LDC2005S13) corpora [31]. The Fisher English Training Part 1, is a collection of conversation telephone speech with 5850 speech samples of up to 10 minutes, approximately 900 hours of speech data. The Fisher English Training Part 2, contains an addition of 5849 speech samples, approximately 900 hours of telephone conversational speech. The corpora is split into training, development and test sets for experimental purposes as shown in Table 1. The splits of the data-sets are consistent over both the ASR and the subsequent noisy-clean phrase context model. The development dataset was used for tuning the phrase-based system using MERT.

We also test the system under mismatched training-usage conditions on TED-LIUM. TED-LIUM is a dedicated ASR corpus consisting of 207 hours of TED talks [32]. The data-set was chosen as it is significantly different to Fisher Corpus. Mismatch conditions include: (i) variations in channel characteristics, Fisher, being a telephone conversations corpus, is sampled at 8kHz where-as the TED-LIUM is originally 16kHz, (ii) noise conditions, the Fisher recordings are

	Train	Development	Test
Fisher English	1,833,088	4906	4914
TED-LIUM	-	507	1155

Table 1: Database split (Number of Utterances)

Method	Dev		Test	
	WER	BLEU	WER	BLEU
Baseline-1 (ASR output)	15.46%	75.71	17.41%	72.99
Baseline-2 (Word based + bigram LM)	16.23%	74.28	18.10%	71.76
Word based + bigram LM + MERT(B)	15.46%	75.70	**17.40%**	72.99
Word based + bigram + MERT(W)	**15.39%**	75.65	**17.40%**	72.77
Word based + trigram LM + MERT(B)	15.48%	75.59	17.47%	72.81
Word based + trigram LM + MERT(W)	15.46%	75.46	17.52%	72.46
Proposed NCPCM	20.33%	66.70	22.32%	63.81
NCPCM + MERT(B)	**15.11%**	**76.06**	**17.18%**	**73.00**
NCPCM + MERT(W)	**15.10%**	**76.08**	**17.15%**	**73.05**
NCPCM + MERT(B) w/o re-ordering	**15.27%**	**76.02**	**17.11%**	**73.33**
NCPCM + MERT(W) w/o re-ordering	**15.19%**	**75.90**	**17.18%**	**73.04**
NCPCM + 10best + MERT(B)	**15.19%**	**76.12**	**17.17%**	**73.22**
NCPCM + 10best + MERT(W)	**15.16%**	**75.91**	**17.21%**	**73.03**

Table 2: Noisy-Clean Phrase Context Model (NCPCM) results (uses exactly same LM as ASR)

significantly noisier, (iii) utterance lengths, TED-LIUM has longer conversations since they are extracts from TED talks, (iv) lexicon sizes, vocabulary size of TED-LIUM is much larger with 150,000 words where-as Fisher has 42,150 unique words, (v) speaking intonation, Fisher being telephone conversations is spontaneous speech, whereas the TED talks are more organized and well articulated. Factors (i) and (ii) mostly affect the performance of ASR due to acoustic differences while (iii) and (iv) affect the language aspects, (v) affects both the acoustic and linguistic aspects of the ASR.

4.2. System Setup

4.2.1. Automatic Speech Recognition System

We used the Kaldi Speech Recognition Toolkit [33] to train the ASR system. In this paper, the acoustic model was trained as a DNN-HMM hybrid system. A tri-gram language model was trained on the transcripts of the training dataset. The CMU pronunciation dictionary [34] was adopted as the lexicon. The resulting ASR is state-of-the-art both in architecture and performance and as such additional gains on top of this ASR are challenging.

4.2.2. Pre-processing

The reference outputs of ASR corpus contain non-verbal signs, such as [laughter], [noise] etc. These event signs might corrupt the phrase context model since there is little contextual information between them. Thus, in this paper, we cleaned our data by removing all these non-verbal signs from dataset. Also, to prevent data sparsity issues, we restricted all of the sample sequences to a maximum length of 100 tokens (given that the database consisted of only 3 sentences having more than the limit).

Method	Dev		Test	
	WER	BLEU	WER	BLEU
Baseline-1 (ASR output)	15.46%	75.71	17.41%	72.99
NCPCM + 3gram NNLM + MERT(B)	**15.46%**	**75.91**	**17.37%**	**73.24**
NCPCM + 3gram NNLM + MERT(W)	**15.28%**	**75.94**	**17.11%**	**73.31**
NCPCM + 5gram NNLM + MERT(B)	**15.35%**	**75.99**	**17.20%**	**73.34**
NCPCM + 5gram NNLM + MERT(W)	**15.20%**	**75.96**	**17.08%**	**73.25**
NCPCM + NNJM-LM (5,4) + MERT(B)	**15.29%**	**75.93**	**17.13%**	**73.26**
NCPCM + NNJM-LM (5,4) + MERT(W)	**15.28%**	**75.94**	**17.13%**	**73.29**

Table 3: Results for Noisy-Clean Phrase Context Models (NCPCM) with Neural Network Language Models (NNLM) and Neural Network Joint Models (NNJM)

4.3. Baseline System

We adopt two different baseline systems because of their relevance to this work:

Baseline-1: The raw performance of the ASR system, because of its relevance to the application of the proposed model.

Baseline-2: The word-based noisy channel model, to connect to a prior work as described in Section 3.1.1 based on [22] and for comparison purposes.

4.4. Evaluation Criteria

The final goal of our work is to show improvements in terms of the performance of ASR. Thus, we provide word error rate as it is a standard in the ASR community. Moreover, Bilingual Evaluation Understudy (BLEU) score [35] is used for evaluating our work, since our model can be also treated as a transfer-function ("translation") system from ASR output to NCPCM output.

5. Results and Discussion

In this section we evaluate the validity of our hypotheses from Section 2 along with the experimental results. The experimental results are presented in three

Method	Dev		Test	
	WER	BLEU	WER	BLEU
Baseline-1	26.92%	62.00	23.04%	65.71
Baseline-2	29.86%	57.55	25.51%	61.79
NCPCM + MERT(B)	**26.06%**	**63.30**	**22.51%**	**66.67**
NCPCM + MERT(W)	**26.15%**	**63.10**	**22.74%**	**66.36**
NCPCM + generic LM + MERT(B)	**25.57%**	**63.98**	**22.38%**	**66.97**
NCPCM + generic LM + MERT(W)	**25.56%**	**63.83**	**22.33%**	**66.96**

Table 4: Results for out-of-domain adaptation using Noisy-Clean Phrase Context Models (NCPCM)

different tasks: (i) overall WER experiments, highlighting the improvements of the proposed system, presented in Tables 2, 3 & 4, (ii) detailed analysis of WERs over subsets of data, presented in Figures 2 & 3, and (iii) analysis of the error corrections, presented in Table 5. The assessment and discussions of each task is structured similar to Section 2 to support their respective claims.

5.1. Re-scoring Lattices

Table 5 shows selected samples through the process of the proposed error correction system. In addition to the reference, ASR output and the proposed system output, we provide the ORACLE transcripts to assess the presence of the correct phrase in the lattice. Cases 4-6 from Table 5 have the correct phrase in the lattice, but gets down-scored in the ASR final output which is then recovered by our system as hypothesized in *Hypothesis* 2.1.

5.2. Recovering Pruned Lattices

In the cases 1 and 2 from Table 5, we see the correct phrase is not present in the ASR lattice, although they were seen in the training and are present in the vocabulary. However, the proposed system manages to recover the phrase as discussed in *Hypothesis* 2.3. Moreover, Case 2 also demonstrates an instance where the confusion occurs due to same phonetic transcriptions ("ridiculously" versus "ridiculous lee") again supporting *Hypothesis* 2.1.

5.3. Recovery of Unseen Phrases

Case 3 of Table 5, demonstrates an instance where the word "qaeda" is absent from the ASR lexicon (vocabulary) and hence absent in the decoding lattice. This forces the ASR to output an unknown-word token ($< unk >$). We see that the system recovers an out-of-vocabulary word "qaeda" successfully as claimed in *Hypothesis* 2.3.

5.4. Better Recovery during Poor Recognitions

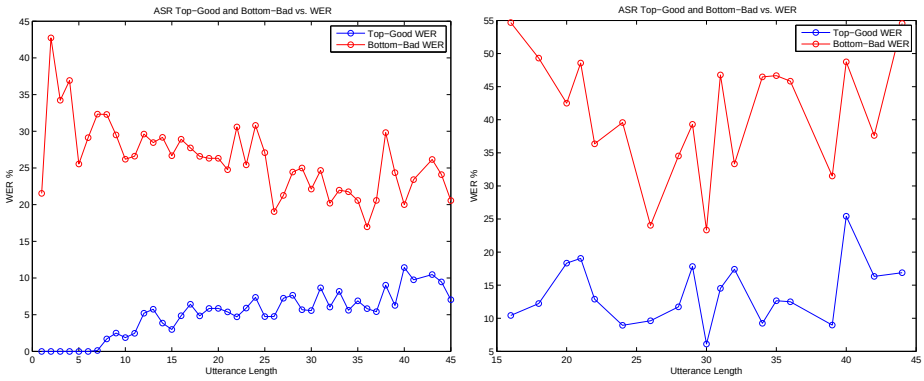
To justify the claim that our system can offset for the performance deficit of the ASR at tougher conditions (as per *Hypothesis* 2.4, we formulate a sub-problem as follows:

Problem Formulation: We divide equally, per sentence length, our development and test datasets into top-good (good recognition results) and bottom-bad (poor recognition results) subsets based on the WER of the ASR and analyze the improvements and any degradation caused by our system.

Figure 3 shows the plots of the above mentioned analysis for different systems as captioned. The blue lines are representative of the improvements provided by our system for top-good subset over different utterance lengths, i.e., it indicates the difference between our system and the original WER of the ASR (negative values indicate improvement and positive values indicate degradation resulting from our system). The green lines indicate the same for bottom-bad subset of the database. The red indicates the difference between the bottom-bad WERs and the top-good WERs, i.e., negative values of red indicate that the system

1.	**REF:**	oysters clams and mushrooms i think
	ASR:	wasters clams and mushrooms they think
	ORACLE:	wasters clams and mushrooms i think
	NCPCM:	oysters clams and mushrooms they think
2.	**REF:**	yeah we had this awful month this winter where it was like a good day if it got up to thirty it was ridiculously cold
	ASR:	yeah we had this awful month uh this winter where it was like a good day if i got up to thirty was ridiculous lee cold
	ORACLE:	yeah we had this awful month this winter where it was like a good day if it got up to thirty it was ridiculous the cold
	NCPCM:	yeah we had this awful month uh this winter where it was like a good day if i got up to thirty it was ridiculously cold
3.	**REF:**	oh well it depends on whether you agree that al qaeda came right out of afghanistan
	ASR:	oh well it depends on whether you agree that al <unk> to came right out of afghanistan
	ORACLE:	oh well it depends on whether you agree that al <unk> to came right out of afghanistan
	NCPCM:	oh well it depends on whether you agree that al qaeda to came right out of afghanistan
4.	**REF:**	they laugh because everybody else is laughing and not because it's really funny
	ASR:	they laughed because everybody else is laughing and not because it's really funny
	ORACLE:	they laugh because everybody else is laughing and not because it's really funny
	NCPCM:	they laugh because everybody else is laughing and not because it's really funny
5.	**REF:**	yeah especially like if you go out for ice cream or something
	ASR:	yeah it specially like if you go out for ice cream or something
	ORACLE:	yeah it's especially like if you go out for ice cream or something
	NCPCM:	yeah especially like if you go out for ice cream or something
6.	**REF:**	we don't have a lot of that around we kind of live in a nicer area
	ASR:	we don't have a lot of that around we kinda live in a nicer area
	ORACLE:	we don't have a lot of that around we kind of live in a nicer area
	NCPCM:	we don't have a lot of that around we kind of live in a nicer area

Table 5: Analysis of selected sentences



(a) In Domain (Fisher) (b) Out of Domain (TEDLIUM)

Figure 2: Top-Good, Bottom-Bad WER Splits. As we can see the WER for top-good is often 0%, which leaves no margin for improvement. We will see the impact of this later, as in Fig. 3

provides more improvements to the bottom-bad subset relative to the top-good subset. The dotted lines represent their respective trends which is obtained by a simple linear regression (line-fitting).

For poor recognitions, we are concerned about the bottom-bad subset, i.e., the green lines in Figure 3. Firstly, we see that the dotted green line is always below zero, which indicates there is always improvements for bottom-bad i.e., poor recognition results. Second, we observe that the dotted red line usually stays below zero, indicating that the performance gains made by the system add more for the bottom-bad poor recognition results compared to the top-good subset (good recognitions). Further, more justifications are provided later in the context of out-of-domain task (Section 5 5.6) where high mismatch results in tougher recognition task are discussed.

5.5. Improvements under all Acoustic Conditions

To justify the claim that our system can always provide benefits over any ASR system (*Hypothesis* 2.5), we need to show that the proposed system: (i) does not degrade the performance of the good recognition, (ii) provides improvements to poor recognition instances, of the ASR. The latter has been discussed and confirmed in the previous Section 5 5.4. For the former, we provide evaluations from two point of views: (1) assessment of WER trends of top-good and bottom-bad subsets (as in the previous Section 5 5.4), and (2) overall absolute WER of the proposed systems.

Firstly, examining Figure 3, we are mainly concerned about the top-good subset pertaining to degradation/improvement of good recognition instances. We observe that the dotted blue line is close to zero in all the cases, which implies that the degradation of good recognition is extremely minimal. Moreover, we observe that the slope of the line is almost zero in all the cases, which indicates that the degradation is minimal and consistent over all the utterance

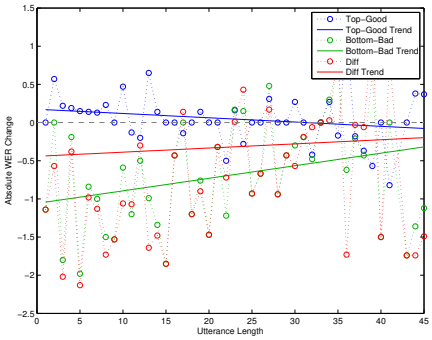
lengths. Moreover, assessing the degradation from the absolute WER perspective, Figure 2a shows the WER over utterance lengths for the top-good and bottom-bad subsets for the in-domain case. The top-good WER is small, at times even 0% (perfect recognition) thereby allowing very small margin for improvement. In such a case, we see minimal degradation. Although we lose a bit on very good recognitions which is extremely minimal, we gain significantly in the case of 'bad' recognitions. Thus to summarize, the damage that this system can make, under the best ASR conditions, is minimal and offset by the potential significant gains present when the ASR hits some tough recognition conditions.

Secondly, examining the overall WER, Table 2 gives the results of the proposed technique. Note that we use the same language model as the ASR. This helps us evaluate a system that does not include additional information. We provide the performance measures on both the development and held out test data. The development data is used for MERT tuning. The performance is evaluated both in terms of WER and BLEU scores. The output of the ASR (i.e., Baseline-1) suggests that the development data is less complex compared to the held out test set. We note that none of the baseline, word-based systems provide any improvements, even when we increase context and use MERT optimization. We also tried different optimization criteria with MERT, i.e., using BLEU(B) and WER(W).

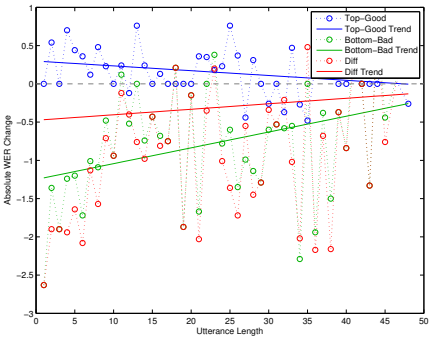
The proposed phrase-based system significantly outperforms the baseline when employing MERT (statistically significant with $p < 0.001$ for both word error and sentence error [36]). For all our subsequent experiments, MERT always helps and hence we only include results with MERT. We find that MERT optimized for WER consistently outperforms that with optimization criteria of BLEU score. We also perform trials by disabling the distortion modeling and see that results remain relatively unchanged. This is as expected since the ASR preserves the sequence of words with respect to the transcripts and there is no reordering effect over the errors. The phrase based context modeling provides a relative improvement of 1.72% over the baseline and the ASR output. Using multiple hypotheses (10-best) from the ASR, we hope to capture more relevant error patterns of the ASR model, thereby enriching the noisy channel modeling capabilities. However, we find that the 10-best gives about the same performance as the 1-best. In this case we considered 10 best as 10 separate training pairs for training the system. In the future we want to exploit the interdependency of this ambiguity (the fact that all the 10-best hypotheses represent a single utterance) for training and error correction at test time.

5.6. Adaptation

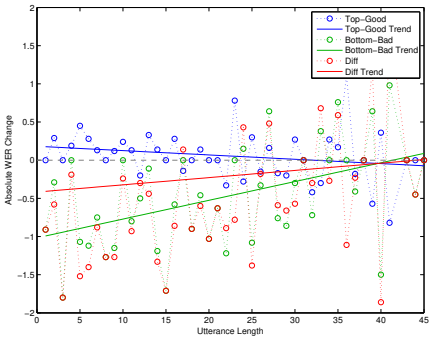
To assess the adaptation capabilities, we evaluate the performance of the proposed noisy-clean phrase context model on an out-of-domain task, TED-LIUM data-base, shown in Table 4. The baseline-1 (ASR performance) confirms of the heightened mismatched conditions between the training Fisher Corpus and the TED-LIUM data-base. However, we see that the phrase context modeling provides modest improvements over the baseline of approximately 2.3% relative on the held-out test set. We note that the improvements are consistent compared



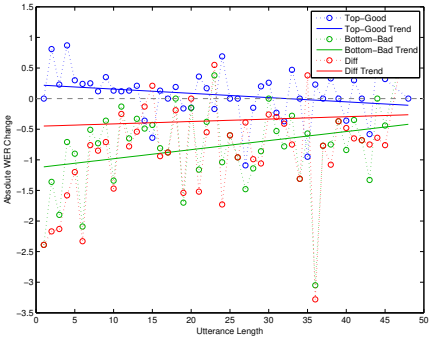
(a) Dev: NCPCM + MERT(W)



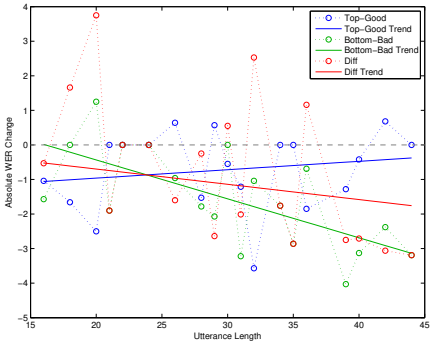
(b) Test: NCPCM + MERT(W)



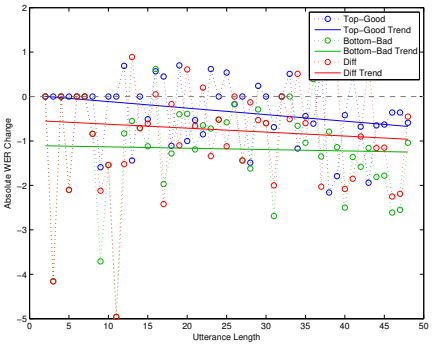
(c) Dev: NCPCM + 5gram NNLM + MERT(W)



(d) Test: NCPCM + 5gram NNLM + MERT(W)



(e) Out-of-Domain Dev: NCPCM + generic LM + MERT(W)



(f) Out-of-Domain Test: NCPCM + generic LM + MERT(W)

Figure 3: Length of hypotheses through our NCPCM models versus absolute WER change. Blue & Green lines represent difference between WER of our system and the baseline ASR, for top-good and bottom-bad hypotheses, respectively. In an ideal scenario, all these lines would be below 0, thus all providing a change in WER towards improving the system. However we see in some cases that the WER increases, especially when the hypotheses length is short and when the performance is good. This is as expected since from Fig. 2 some cases are at 0% WER due to the already highly-optimized nature of our ASR. The red line represents the aggregate error over all data for each word length and as we can see in **all** cases the trend is one of **improving** the WER.

to the earlier in-domain experiments in Table 2. Moreover, since the previous LM was trained on Fisher Corpus, we adopt a more generic English LM which provides further improvements of up to 3.1%. This confirms the robustness of the proposed approach and its application to the out-of-domain data. More importantly, the result confirms *Hypothesis* 2.6, i.e., our claim of rapid adaptability of the system to varying mismatched acoustic and linguistic conditions and also supports the possibility of training our system without the need of an ASR.

Further, comparing the WER trends from the in-domain task (Figure 3b) to the out-of-domain task (Figure 3f), we firstly find that the improvements in the out-of-domain task are obtained for both top-good (good recognition) and bottom-bad (bad recognition), i.e., both the dotted blue line and the dotted green line are always below zero. Secondly, we observe that the improvements are more consistent throughout all the utterance lengths, i.e., all the lines have near zero slopes compared to the in-domain task results. The two findings are fairly meaningful considering the high mismatch of the out-of-domain data.

5.7. Exploit Longer Context

Firstly, inspecting the error correction results from Table 5, cases 2 and 4 hint at the ability of the system to select appropriate word-suffixes using long term context information.

Second, from detailed WER analysis in Figure 3, we see that the bottom-bad (dotted green line) improvements decrease with increase in length in most cases, hinting at potential improvements to be found by using higher contextual information for error correction system as future research directions. Moreover, closer inspection across different models, comparing the trigram ARPA model (Figure 3b) with the 5gram NNLM (Figure 3d), we find that the NNLM provides minimal degradation and better improvements especially for longer utterances by exploiting more context (the blue dotted line for NNLM has smaller intercept value as well as higher negative slope). We also find that for the bottom-bad poor recognition results (green dotted-line), the NNLM gives consistent (smaller positive slope) and better improvements especially for the higher length utterances (smaller intercept value). Thus emphasizing the gains provided by higher context NNLM.

Third, Table 3 shows the results obtained using a neural network language model of higher orders (also trained only on the in-domain data). Comparing results from Table 2 with Table 3, we note the benefits of higher order LMs, with the 5-gram neural network language model giving the best results (a relative improvement of 1.9% over the baseline), outperforming the earlier ARPA models as per *Hypothesis* 2.7. However, the neural network joint model LM with target context of 5 and source context of 4 did not show significant improvements over the traditional neural LMs. We expect the neural network models to provide further improvements with more training data.

5.8. Regularization

Finally, the last case in Table 5 is of text regularization as described in Section 2, *Hypothesis* 2.8. Overall, in our experiments, we found that approximately 20% were a case of text regularization and the rest were a case of the former hypotheses.

6. Conclusions and Future Work

In this work, we proposed a noisy channel model for error correction based on phrases. The system post-processes the output of an automated speech recognition system and as such any contributions in improving ASR are orthogonal. Firstly, we hypothesized what type of errors we target to correct. Later on, we supported our claims with apt problem formulation and their respective results. We showed that our system can improve the performance of the ASR by (i) re-scoring the lattices (*Hypothesis* 2.1), (ii recovering words pruned from the lattices (*Hypothesis* 2.2), (iii) recovering words never seen in the vocabulary and training data (*Hypothesis* 2.3), (iv) exploiting longer context information (*Hypothesis* 2.7), and (v) by regularization of language syntax (*Hypothesis* 2.8). Moreover, we also claimed and justified that our system can provide more improvement in low-performing ASR cases (*Hypothesis* 2.4), while keeping the degradation to minimum in cases when the ASR performs well (*Hypothesis* 2.5). In doing so, our system could effectively adapt (*Hypothesis* 2.6) to changing recognition environments and provide improvements over any ASR systems.

In our future work, the output of the noisy-clean phrase context model could be fused with the ASR belief to obtain a new hypothesis. We also intend to introduce ASR confidence scores and signal SNR estimates, to improve the channel model. We are investigating introducing the probabilistic ambiguity of the ASR in the form of lattice or confusion networks as inputs to the channel-inversion model.

Further, we will utilize sequence-to-sequence (Seq2seq) translation modeling [16] to map ASR outputs to reference transcripts. The Seq2seq model has been shown to have benefits especially in cases where training sequences are of variable length [37]. We intend to employ Seq2seq model to encode ASR output to a fixed-size embedding and decode this embedding to generate the corrected transcripts.

Financial Support

The U.S. Army Medical Research Acquisition Activity, 820 Chandler Street, Fort Detrick MD 21702- 5014 is the awarding and administering acquisition office. This work was supported by the Office of the Assistant Secretary of Defense for Health Affairs through the Psychological Health and Traumatic Brain Injury Research Program under Award No. W81XWH-15-1-0632. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the Department of Defense.

References

[1] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[2] L. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Processing Magazine 29 (6) (2012) 82–97.

[4] G. E. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 20 (1) (2012) 30–42.

[5] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Proceedings of Acoustics, speech and signal processing (ICASSP), 2013 IEEE international conference on, IEEE, 2013, pp. 6645–6649.

[6] A. Graves, S. Fernández, F. Gomez, J. Schmidhuber, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 369–376.

[7] H. Strik, C. Cucchiarini, Modeling pronunciation variation for ASR: A survey of the literature, Speech Communication 29 (2) (1999) 225–246.

[8] M. Wester, Pronunciation modeling for ASR–knowledge-based and data-derived methods, Computer Speech & Language 17 (1) (2003) 69–85.

[9] P. G. Shivakumar, A. Potamianos, S. Lee, S. Narayanan, Improving speech recognition for children using acoustic adaptation and pronunciation modeling., in: WOCCI, 2014, pp. 15–19.

[10] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, A tree-based statistical language model for natural language speech recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (7) (1989) 1001–1008.

[11] R. Moore, D. Appelt, J. Dowding, J. M. Gawron, D. Moran, Combining linguistic and statistical knowledge sources in natural-language processing for atis, in: Proc. ARPA Spoken Language Systems Technology Workshop, 1995.

[12] R. Rosenfeld, Two decades of statistical language modeling: Where do we go from here?, Proceedings of the IEEE 88 (8) (2000) 1270–1278.

[13] E. Arisoy, T. N. Sainath, B. Kingsbury, B. Ramabhadran, Deep neural network language models, in: Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT, Association for Computational Linguistics, 2012, pp. 20–28.

[14] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, S. Khudanpur, Recurrent neural network based language model., in: Proceedings of Interspeech, Vol. 2, 2010, p. 3.

[15] M. Sundermeyer, R. Schlüter, H. Ney, LSTM neural networks for language modeling., in: Proceedings of Interspeech, 2012, pp. 194–197.

[16] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[17] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: Proceedings of Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 4945–4949.

[18] W. Chan, N. Jaitly, Q. Le, O. Vinyals, Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, in: Proceedings of Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 4960–4964.

[19] W. A. Ainsworth, S. Pratt, Feedback strategies for error correction in speech recognition systems, International Journal of Man-Machine Studies 36 (6) (1992) 833–842.

[20] J. Noyes, C. Frankish, Errors and error correction in automatic speech recognition systems, Ergonomics 37 (11) (1994) 1943–1957.

[21] B. Suhm, B. Myers, A. Waibel, Multimodal error correction for speech user interfaces, ACM transactions on computer-human interaction (TOCHI) 8 (1) (2001) 60–98.

[22] E. K. Ringger, J. F. Allen, Error correction via a post-processor for continuous speech recognition, in: Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on, Vol. 1, IEEE, 1996, pp. 427–430.

[23] M. Jeong, S. Jung, G. G. Lee, Speech recognition error correction using maximum entropy language model, in: Proc. of INTERSPEECH, 2004, pp. 2137–2140.

[24] B. Roark, M. Saraclar, M. Collins, Discriminative n-gram language modeling, Computer Speech & Language 21 (2) (2007) 373–392.

[25] P. Woodland, D. Povey, Large scale discriminative training of hidden markov models for speech recognition, Computer Speech & Language 16 (1) (2002) 25 – 47. doi:http://dx.doi.org/10.1006/csla.2001.0182.
URL http://www.sciencedirect.com/science/article/pii/S0885230801901822

[26] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, P. S. Roossin, A statistical approach to machine translation, Computational linguistics 16 (2) (1990) 79–85.

[27] P. Koehn, F. J. Och, D. Marcu, Statistical phrase-based translation, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, ACL, 2003, pp. 48–54.

[28] A. Vaswani, Y. Zhao, V. Fossum, D. Chiang, Decoding with large-scale neural language models improves translation., in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2013, pp. 1387–1392.

[29] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. M. Schwartz, J. Makhoul, Fast and robust neural network joint models for statistical machine translation., in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics„ ACL, 2014, pp. 1370–1380.

[30] F. J. Och, Minimum error rate training in statistical machine translation, in: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, ACL, 2003, pp. 160–167.

[31] C. Cieri, D. Miller, K. Walker, The Fisher Corpus: a resource for the next generations of speech-to-text., in: International Conference on Language Resources and Evaluation, Vol. 4, LREC, 2004, pp. 69–71.

[32] A. Rousseau, P. Deléglise, Y. Estève, Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks., in: International Conference on Language Resources and Evaluation, LREC, 2014, pp. 3935–3939.

[33] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, et al., The Kaldi speech recognition toolkit, in: IEEE 2011 workshop on automatic speech recognition and understanding, no. EPFL-CONF-192584, IEEE Signal Processing Society, 2011.

[34] R. L. Weide, The CMU pronouncing dictionary, URL: http://www. speech. cs. cmu. edu/cgibin/cmudict.

[35] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual

meeting on association for computational linguistics, ACL, 2002, pp. 311–318.

[36] L. Gillick, S. J. Cox, Some statistical issues in the comparison of speech recognition algorithms, in: Proceedings of Acoustics, Speech, and Signal Processing,. ICASSP-89., 1989 International Conference on, IEEE, 1989, pp. 532–535.

[37] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, arXiv preprint arXiv:1409.1259.