Toward Robust Neural Machine Translation for Noisy Input Sequences

Matthias Sperber, Jan Niehues, Alex Waibel

Interactive Systems Labs
Karlsruhe Institute of Technology, Germany

`matthias.sperber@kit.edu`

Abstract

Translating noisy inputs, such as the output of a speech recognizer, is a difficult but important challenge for neural machine translation. One way to increase robustness of neural models is by introducing artificial noise to the training data. In this paper, we experiment with appropriate forms of such noise, exploring a middle ground between general-purpose regularizers and highly task-specific forms of noise induction. We show that with a simple generative noise model, moderate gains can be achieved in translating erroneous speech transcripts, provided that type and amount of noise are properly calibrated. The optimal amount of noise at training time is much smaller than the amount of noise in our test data, indicating limitations due to trainability issues. We note that unlike our baseline model, models trained on noisy data are able to generate outputs of proper length even for noisy inputs, while gradually reducing output length for higher amount of noise, as might also be expected from a human translator. We discuss these findings in details and give suggestions for future work.

1. Introduction

Many natural language processing tasks require applying sequence models on corrupted or noisy input sequences. A typical example is machine translation of erroneous outputs from an automatic speech recognizer (ASR). Ideally, we would like the translation process to ignore or even correct the corrupted inputs. Translation models are usually trained on wellformed parallel sentences that do not exhibit such noise. This results in a harmful mismatch between training and test data, and further aggravates the difficulty of having to transform malformed inputs in the first place. The now prevalent neural sequence-to-sequence models [1, 2, 3] have been identified to be especially sensitive to noisy data [4, 5, 6], and more specifically to corrupted inputs due to erroneous ASR [7].

Robustness at test-time may be improved by inducing suitable forms of noise during the training process. The spectrum of suitable approaches ranges from general-purpose regularizers[1] such as dropout [9] to task-specific approaches that alter the training data to resemble the corrupted inputs at test-time. Task-specific approaches can make stronger assumptions about the data distribution and are potentially more effective or provide additive gains when combined with general-purpose methods. As a disadvantage, they are also more complex and may require task-specific knowledge or resources. Another tradeoff to consider concerns trainability. Neural sequence-to-sequence models are known to suffer from explaining-away effects, where models may learn to generate outputs by relying on the target-side context while ignoring the source-side context [10, 11], especially when the source side provides only a weak or noisy signal. As a result, careful calibration of type and amount of induced noise may be necessary.

Prior work on speech translation attempted inducing task-specific noise by training on actual ASR outputs paired with their correct translations. Unfortunately, such data is scarce, and exploiting it may not be straightforward (see [12] and §4.1; but [13]). Alternatively, it has been proposed to synthesize realistic ASR error patterns and suitable translations thereof, and augment the training data accordingly [14, 15]. However, this approach has not yet been shown to transfer to neural machine translation, and is relatively complex, requiring availability of resources such as pronunciation dictionaries and suitable language models.

In this paper, we seek to improve robustness of a neural machine translation model applied to speech recognition input by exploring tradeoffs between general-purpose and task-specific methods. For this purpose, we introduce a simple noise model that is inspired by the word error rate (WER), which categorizes the common ASR error types into substitutions, insertions, and deletions. Accordingly, our noise model artificially corrupts the source side of a parallel training corpus by randomly introducing substitutions, insertions, or deletions. Our noise model is simpler than the prior approaches [14, 15], but nonetheless effective, and provides a flexible test bed that allows exploring the middle ground between task specificity and generality in the context of neural sequence-to-sequence models. In addition, we discuss preliminary efforts toward refining the noise model to capture more task-specific intuitions similar to these prior ap-

[1]In this paper, we use the notions of good generalization (avoiding overfitting, e.g. via regularization) and robustness (stability w.r.t. noisy data) loosely interchangeably. In fact, both are strongly linked in the sense that in general, good generalization implies robustness [8].

proaches.

We conduct experiments on the Fisher and Callhome Spanish–English speech translation corpus [12] and observe minor improvements in robustness when applying our noise model. We find that increasing the amount of noise during training up to a certain point helps translation of noisy inputs but hurts translation of clean inputs. Strikingly, the optimal amount of noise is much smaller than the amount of noise in our test data, indicating trainability issues. Increasing the amount of noise further leads to a drop in recall but slight increase in precision, leading to the question of to what extent it is desirable from a usability perspective to drop uncertain source-side content as opposed to guessing a translation for it. We conclude with discussing shortcomings of our approach and give suggestions for future work.

2. Related Work

Inducing noise in the training inputs can be seen as a form of data augmentation, which has been used in several applications such as acoustic modeling [16], computer vision [17], language modeling [18], and statistical machine translation where data can be augmented by paraphrases [19]. It has been described as more powerful than general-purpose regularization in the context of deep learning [17]. Note that these approaches aim at inducing *label-preserving* noise, in contrast to our noise model which may alter or destroy the meaning of an input despite keeping targets unchanged. Data augmentation has also been used specifically to improve robustness to noisy inputs as in our work, such as research on speech recognition under noisy conditions [20] and translating spelling mistakes [5, 6]. The latter work demonstrates the importance of using natural (as opposed to synthetic) noise to make models robust to realistic noisy test-time conditions.

Several works have identified noisy or mismatched text inputs as a challenge for neural models: [21] mention domain mismatch as a challenge for neural machine translation, [4] show that NMT suffers from noisy training data, [22] show that recurrent neural networks can be sensitive to corrupted input sequences.

Our approach is methodologically inspired by reward-augmented maximum likelihood (RAML) [23]. We use a similar sampling procedure on the source side, instead of the target side as in RAML. However, RAML is very differently motivated, aiming at fixing exposure bias whereas we are concerned with noise from upstream components. In addition, sampling according to [23]'s approach is biased toward producing less deletions than substitutions and insertions, which our noise model purposefully avoids.

Finally, prior work has dealt with uncertain inputs from upstream components through explicit representation of the uncertainty, for example by directly translating word lattices produced by the speech recognizer [24, 25, 26, 27].

3. Noise Model

This section introduces a noise model that will be applied to every input sentence of the training data. The general idea follows the intuitions behind the WER, according to which ASR errors can be categorized into substitutions, insertions, and deletions. Design goals are flexibility to capture various levels of refinement, and convenient control of the amount of noise and other properties. We first describe the vanilla model, and then present several refinements.

3.1. Vanilla Noise Model

The vanilla noise model, outlined in Algorithm 1, can be summarized as follows. For each sentence, we first decide on the number of edits, while considering the desired amount of overall noise. The edits are then randomly divided into substitutions, insertions and deletions. Finally, for each edit a position is randomly chosen along with a new word for substitutions and deletions.

More formally, let hyperparameter $\tau{\in}[0,1]$ denote the amount of noise to be induced, let V be a sampling vocabulary, and assume a sentence of length n as $\langle w_0{=}\text{sos}, w_1, \cdots, w_n, w_{n+1}{=}\text{eos}\rangle$. We first draw the number of edits e (line 1). The Poisson distribution is a suitable choice because it is defined over non-negative integers and has probability mass centered around its mean. For simplicity, we allow a maximum of n edits for a sentence of length n. Thus, we sample according to a n-truncated Poisson distribution [28], defined as $P_\lambda(k) \propto \exp(-\lambda)\frac{\lambda^k}{k!}$ with support $k \in \{0, \cdots, n\}$, where we set $\lambda := \tau \cdot n$. The mean of this distribution is approximately λ. Because of the finite support, this distribution reduces to a categorical distribution and is thus trivial to sample from.

Next, we draw the number of substitutions n_s, number of insertions n_i, and number of deletions n_d such that $n_\text{s}{+}n_\text{i}{+}n_\text{d} = e$ and $n_\text{s}, n_\text{i}, n_\text{d} \in \mathbb{N}^0$ (line 2). This defines a space over $\langle n_\text{s}, n_\text{i}, n_\text{d}\rangle$, known as the discrete 3-simplex [29]. We sample from a uniform distribution over this space (§3.1.1).

We then draw without replacement a position for each substitution, insertion, and deletion (lines $3, 4, 5$). Finally, we corrupt the original sentence accordingly (lines 6 through 16), sampling new words for substitutions and insertions uniformly from the sampling vocabulary (lines 7 and 14).

3.1.1. Sampling from the Discrete Simplex

In order to determine the number of edit operations $n_1, \cdots, n_d$ for each operation type (here: $n_\text{s}, n_\text{i}, n_\text{d}$, corresponding to substitutions, insertions, and deletions), we uniformly sample $\langle n_1, \cdots, n_d\rangle \sim \text{DiscrSimplex}\,(d, e)$ such that $\sum_{i=1}^{d} n_d = e$ and $n_i \in \mathbb{N}^0$. This can be accomplished by slightly adjusting the sampling approach for the continuous simplex [30] to the discrete simplex as follows. Sample auxiliary random variables $x_1, \cdots, x_{d-1}$ uniformly without replacement from $\{1, 2, \cdots, e{+}d{-}1\}$. Let $x_0{=}0, x_d{=}e{+}d$.

Algorithm 1 Vanilla Noise Model.
– given magnitude of noise: $\tau \in [0, 1]$
– given sentence $\langle w_0\text{=sos}, w_1, \cdots, w_n, w_{n+1}\text{=eos}\rangle$
– given vocabulary V

```
 1: sample distance $e \sim \text{TruncPoisson}(\tau \cdot n, n)$
 2: sample $\langle n_s, n_i, n_d\rangle \sim \text{DiscrSimplex}(3, e)$
 3: sample substitution positions $s_1, \cdots, s_{n_s}$ uniformly
    without replacement from $\{1, \cdots, n\}$
 4: sample insertion positions $i_1, \cdots, i_{n_i}$ uniformly without
    replacement from $\{0, \cdots, n\}$
 5: sample deletion positions $d_1, \cdots, d_{n_d}$ uniformly without
    replacement from $\{1, \cdots, n\} \setminus \{s_1, \cdots, s_{n_s}\}$
 6: for $i \leftarrow 1 \cdots n_s$ do
 7:     uniformly sample $\tilde{w} \sim V$
 8:     replace $w_i \leftarrow \tilde{w}$                ▷ substitution
 9: end for
10: for $i \leftarrow 1 \cdots n_d$ do
11:     replace $w_i \leftarrow \epsilon$                ▷ deletion
12: end for
13: for $i \leftarrow n_i \cdots 1$ do
14:     uniformly sample $\tilde{w} \sim V$
15:     insert $\tilde{w}$ between $w_i$ and $w_{i+1}$   ▷ insertion
16: end for
```

Finally, let $n_i = x_i - x_{i-1} - 1, \forall i \in \{1, 2, \ldots, d\}$. Proof of correctness directly follows argumentation in [30].

3.2. Refinements

The following discusses several simple steps, all aiming at making the sampled noise more similar to the ASR outputs. For more elaborate refinements, we refer to prior work [14, 15].

3.2.1. Sampling Vocabulary: Linguistic Conditioning

The vanilla model draws substitutions and insertions uniformly from the vocabulary (lines 7 and 14), causing a large portion of induced noise to be drawn from the long tail of rarely occurring words. As a more linguistically informed strategy, we can draw from a unigram instead of a uniform distribution over the vocabular, replacing lines 7 and 14 accordingly.

3.2.2. Sampling Vocabulary: Acoustic Conditioning

Preferably, substitutions would be chosen based on acoustic similarity to the original input. Here, we use negative character edit distance as an approximation for acoustic similarity, and sample according to exponentiated distances $p(\tilde{w}|w) \propto \exp(-\text{dist}(w, \tilde{w}))$, replacing lines 7 and 14.

3.2.3. Sampling Positions

ASR tends to err more often for certain types of words than others. For example, shorter tend to be confused more often because these words can suffer from linguistic and acoustic ambiguity. We can model this by substituting or deleting short words more often, again working with an exponentiated distribution $p(\text{pos} = j) \propto \exp(-|w_j|)$ (lines 3 and 5).

3.2.4. Proportion of Error Types

ASR usually produces more substitutions than insertions and deletions. We may wish to reflect this in our noise distribution, for example by drawing edit operations from a 7-simplex and assigning 1 bucket to insertions, 1 bucket to deletions, and 5 buckets to substitutions[2] (lines 1 and 2).

4. Experiments

We conduct experiments on the Fisher and Callhome Spanish–English speech translation corpus [12], a corpus of Spanish telephone conversations that includes ASR transcripts. The Fisher portion consists of telephone conversations between strangers, while the Callhome portion contains telephone conversations between relatives or friends. The training data size of Fisher/Train is 138,819 sentences, we do not make use of the much smaller Callhome/Train part of the corpus. We use Fisher/Dev as held-out testing data for most of our experiments, which has a WER of 41.3%. The relatively high WER is due to the spontaneous speaking style and challenging acoustics. It should also be noted that the ASR model used by [12] is slightly outdated by now and better WER are achieved with recent advancements [31, 32]. Here, our main concern is handling of noisy inputs, not achieving the most competitive end-to-end BLEU scores.

For preprocessing, we tokenized and lowercased source and target sides. We removed punctuation from the reference transcripts on the source side for consistency with the automatic transcripts which also do not contain punctuation. Although punctuation is removed, we use the manual segmentation as given in the corpus, and leave dealing with noisy segmentation boundaries to future work. Our source-side vocabulary contains all words from the automatic transcripts for Fisher/Train, replacing singletons by an unknown word token, totaling 14,648 words. Similarly, on the target side we used all words from the reference translations of Fisher/Train, replacing singletons by the unknown word, yielding 10,800 words in total.

Our implementation uses the eXtensible Neural Machine Translation (XNMT) toolkit,[3] which is based on DyNet [33]. We use a standard attentional encoder-decoder architecture with one encoder and decoder layer. The encoder is a bidirectional LSTM with 256 hidden units per direction, the decoder is an LSTM with 512 hidden units. We used 128-dimensional word embeddings. We use variational dropout [34] in encoder and decoder LSTMs (p=0.5). To obtain a more noise-

[2]This particular choice of distribution is motivated by our experimental data containing about 5 times as many substitutions as insertions or deletions.

[3]github.com/neulab/xnmt

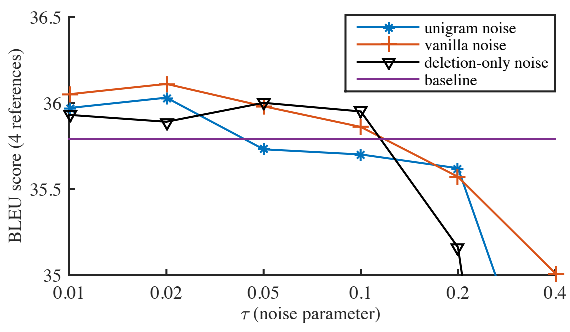

Figure 1: BLEU scores (4 references) on Fisher/Dev, using ASR transcripts as inputs, varying the amount of induced noise.

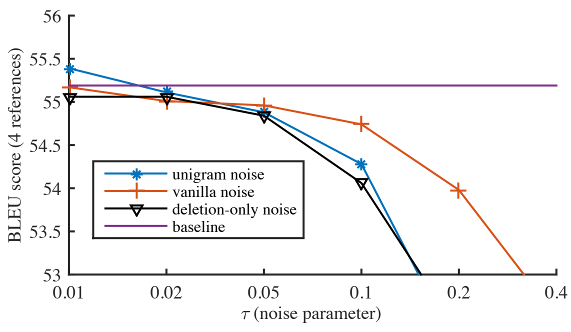

Figure 2: BLEU scores (4 references) on Fisher/Dev, using clean reference transcripts as inputs, varying the amount of induced noise.

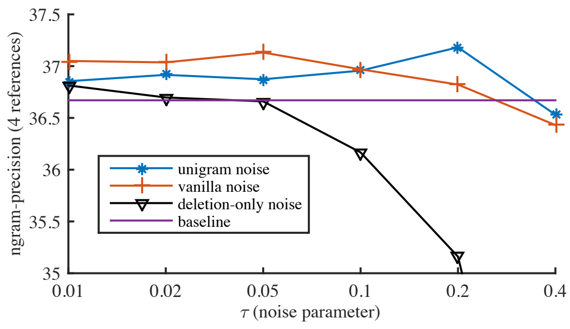

Figure 3: ngram precision (BLEU without brevity penalty) on Fisher/Dev, using ASR transcripts as inputs, varying the amount of induced noise.

robust baseline, we also apply word type dropout [34] to the source word embeddings (p=0.1).

Training was performed with Adam [35]. For all experiments, we first pretrained a model using reference transcripts only, starting from an initial learning rate of 0.0003, restarting Adam and halving learning rates when perplexities did not improve for 2 consecutive epochs [36]. We then fine-tuned the model weights by training on noisy data according to the proposed noise model. Fine-tuning used an initial learning rate of 0.00001 and the same learning rate decay and restarting strategy as during pre-training. The pretraining-finetuning scheme was used in part to make experimental effort manageable, and in part because we observed better BLEU scores in preliminary experiments.

4.1. Main Results

Figure 1 compares our baseline model against several models trained using our noise model. VANILLA NOISE induces varying amounts of noise using the basic model and yields substantial improvements over the BASELINE, which is trained only on clean data. UNIGRAM NOISE replaces the uniform sampling distribution with a unigram distribution and yields similar gains. Perhaps surprisingly, DELETION-ONLY NOISE, a simplified model that induces only deletions, produces strong results as well. We present a possible explanation later. Note that improvements are achieved only for small to moderate amounts of noise. For $\tau = 0.4$, which is close to the WER of the test data, results are rather poor. This indicates that we are facing a trade-off between better trainability for small values for τ, and better distributional similarity with the test data for higher values for τ. We also trained a model by fine-tuning on actual 1-best transcripts rather than using the proposed noise model. Results are rather poor at 32.55 BLEU points, which may be explained by the amount of noise being so high that trainability is compromised, and possibly by some proneness to overfitting because the same noise is used in every epoch.

Figure 2 shows performance of the same models when using clean reference transcripts as inputs. Translation of clean inputs is improved for one configuration of inducing noise, in which case the induced noise can be understood to act as a general-purpose regularizer.[4] However, note that performance drops quickly when increasing the noise parameter τ, again highlighting both the importance of distributional similarity between training and test data, and potential trainability issues.

Figure 3 evaluates models in terms of n-gram precision, which we compute identically to the BLEU score but drop the brevity penalty. Comparing results to Figure 1, we can clearly observe some interactions that lead to trading off precision for recall. Most notably, DELETION-ONLY NOISE performs substantially worse than VANILLA NOISE and UNIGRAM NOISE when measuring only precision. Closer analysis showed that models generally tend to produce shorter outputs the more noise is contained in the inputs. The BLEU metric's brevity penalty is known to punish such short outputs quite severely. DELETION-ONLY NOISE, on the other hand, is trained on inputs where words are deleted. In other words the training-time inputs are shorter than the test-time inputs, counteracting the tendency to produce shorter outputs and thereby avoiding a severe brevity penalty. While this helps BLEU score, arguably producing shorter outputs for

[4]This explanation is supported by prior work relating data noising to traditional smoothing methods [18].

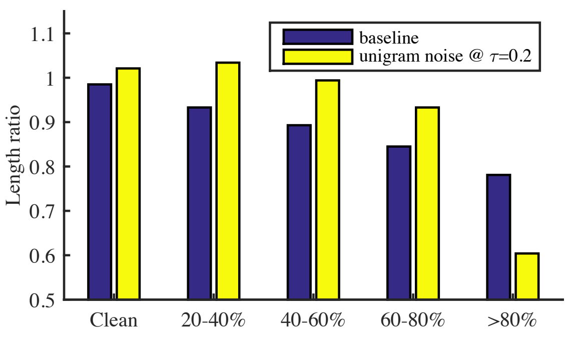

Figure 4: Length ratio of translations when binning test inputs according to their WER.

noisier inputs is a desirable behavior that we would expect also from a human translator, and BLEU may thus not be sufficient as ground for model selection in our task.

4.2. Impact of ASR Quality

For this experiment, we combined all available test data (Fisher/Dev, Fisher/Dev2, Fisher/Test, Callhome/Devtest, Callhome/Evltest), and divided it into bins according to ASR WER. Figure 4 shows the length ratio of translations produced for these inputs for two different models. It can be seen that both BASELINE and UNIGRAM NOISE produce length ratios close to 1.0 for clean inputs. However, when inputs contain even moderate amounts of noise, uncertainty in BASELINE seems to become problematic and outputs quickly become rather short. UNIGRAM NOISE on the other hand appears to handle noisier inputs much more gracefully, while also exhibiting a tendency for shorter outputs when inputs are noisy.

While this demonstrates greater robustness of the noise-induced model, it also raises the question as to what extent shorter outputs for noisy inputs are desirable. Arguably, a human translator may exhibit the same tendency, but further research is required to answer the question of what behavior is desired by a user: dropping uncertain inputs and thus erring on the side of better precision, or trying to guess translations for those inputs anyways and erring on the side of better recall.[5]

4.3. Negative Results for Model Refinements

Our analysis so far only considered the unigram-sampling refinement of the vanilla noise model. We also tested acoustic conditioning (§3.2.2), better sampling positions (§3.2.3), and more realistic proportion of error types (§3.2.4), but did not observe noticeable improvements and do not present details here. Future work may attempt using even more realistic error patterns along the lines of prior work [14, 15]. However, a possible difficulty when trying this may be that, unlike phrase-based machine translation, neural machine translation has been known to be ineffective at learning from rare training examples [11]. Permutations of error patterns potentially consist of mainly such hard-to-learn rarely occurring patterns. Counteracting this by increasing the amount of noise may lead to trainability issues as observed in our experiments as well. Instead, it may be necessary to represent knowledge about confusability more explicitly and efficiently in the model.

5. Conclusion

We identified robustness to noisy inputs as a challenge for neural sequence-to-sequence models, and proposed to introduce randomized noise into the training using a simple generative noise model. We found that this improves robustness when properly calibrating type and amount of noise, and that type and amount of noise at training and test time affect the length of the outputs. We highlighted the trade-off between trainability and distributional data similarity, and found that the amount of induced noise must be much smaller than the expected noise at test time for good results. Future work may investigate appropriate trade-offs between precision and recall when translating noisy inputs from a user perspective, use our method for different tasks such as translating user-generated content, and experiment with more refined types of noise or other ways of modeling acoustic similarity in the context of neural machine translation of ASR outputs.

6. Acknowledgments

We thank Sakriani Sakti and the anonymous reviewers for their valuable feedback that helped improve this work.

7. References

[1] N. Kalchbrenner and P. Blunsom, "Recurrent Continuous Translation Models," in *Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, Washington, USA, 2013, pp. 1700–1709.

[2] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, Montréal, Canada, 2014, pp. 3104–3112.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *International Conference on Representation Learning (ICLR)*, San Diego, USA, 2015.

[4] B. Chen, R. Kuhn, G. Foster, C. Cherry, and F. Huang,

[5]Consider a typical example we found in an English ASR transcript, *Boesch as ever his son decides to have a feast.* While the first 3 or 4 words are clearly recognition mistakes (caused by a rare name in the audio), the rest makes sense and a human might choose to only translate the latter part. Another example is *buildings and boundaries around the location very part*, where the last 2 words are easily recognizable as mistakes and could be dropped before translating. However, an experienced translator might guess correctly that *very part* should be replaced by *where to park*. We suggest investigation of desirable translation strategies from a usability perspective for future work.

"Bilingual Methods for Adaptive Training Data Selection for Machine Translation," in *Association for the Machine Translation in Americas (AMTA)*, 2016.

[5] G. Heigold, G. Neumann, and J. van Genabith, "How Robust Are Character-Based Word Embeddings in Tagging and MT Against Wrod Scramlbing or Randdm Nouse?" *arXiv:1704.04441*, 2017.

[6] Y. Belinkov and Y. Bisk, "Synthetic and Natural Noise Both Break Neural Machine Translation," *arXiv:1711.02173*, 2017.

[7] N. Ruiz, M. A. Di Gangi, N. Bertoldi, and M. Federico, "Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Stockholm, Sweden, 2017, pp. 2635–2639.

[8] C. Caramanis, S. Mannor, and H. Xu, "Robust Optimization in Machine Learning," in *Optimization for Machine Learning*. The MIT Press, 2011.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[10] L. Yu, P. Blunsom, C. Dyer, E. Grefenstette, and T. Kocisky, "The Neural Noisy Channel," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[11] P. Koehn, "Neural Machine Translation," in *Statistical Machine Translation*, 2017, ch. 13.

[12] M. Post, G. Kumar, A. Lopez, D. Karakos, C. Callison-Burch, and S. Khudanpur, "Improved Speech-to-Text Translation with the Fisher and Callhome Spanish–English Speech Translation Corpus," in *International Workshop on Spoken Language Translation (IWSLT)*, Heidelberg, Germany, 2013.

[13] P.-J. Chen, I.-H. Hsu, Y.-Y. Huang, and H.-Y. Lee, "Mitigating the Impact of Speech Recognition Errors on Chatbot using Sequence-to-sequence Model," *arXiv:1709.07862*, 2017.

[14] Y. Tsvetkov, F. Metze, and C. Dyer, "Augmenting translation models with simulated acoustic confusions for improved spoken language translation," in *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Gothenburg, Sweden, 2014, pp. 616–625.

[15] N. Ruiz, Q. Gao, W. Lewis, and M. Federico, "Adapting Machine Translation Models toward Misrecognized Speech with Text-to-Speech Pronunciation Rules and Acoustic Confusability," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Dresden, Germany, 2015, pp. 2247–2251.

[16] N. Jaitly and G. Hinton, "Vocal tract length perturbation (VTLP) improves speech recognition," in *International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, Atlanta, USA, 2013.

[17] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[18] Z. Xie, S. I. Wang, J. Li, D. Lévy, A. Nie, D. Jurafsky, and A. Y. Ng, "Data Noising as Smoothing in Neural Network Language Models," in *International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.

[19] C. Callison-Burch, P. Koehn, and M. Osborne, "Improved statistical machine translation using paraphrases," in *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, New York City, USA, 2006, pp. 17–24.

[20] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi, and S. Kuroiwa, "Reverberant speech recognition based on denoising autoencoder," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Lyon, France, 2013, pp. 3512–3516.

[21] P. Koehn and R. Knowles, "Six Challenges for Neural Machine Translation," *arXiv:1706.03872*, 2017.

[22] J. Li, W. Monroe, and D. Jurafsky, "Understanding Neural Networks through Representation Erasure," *arXiv:1612.08220*, 2017.

[23] M. Norouzi, S. Bengio, Z. Chen, N. Jaitly, M. Schuster, Y. Wu, and D. Schuurmans, "Reward Augmented Maximum Likelihood for Neural Structured Prediction," in *Neural Information Processing Systems Conference (NIPS)*, Barcelona, Spain, 2016, pp. 1723–1731.

[24] H. Ney, "Speech Translation: Coupling of Recognition and Translation," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Phoenix, USA, 1999, pp. 517–520.

[25] F. Casacuberta, H. Ney, F. J. Och, E. Vidal, J. M. Vilar, S. Barrachina, I. Garcia-Varea, D. Llorens, C. Martinez, S. Molau, F. Nevado, M. Pastor, D. Picco, A. Sanchis, and C. Tillmann, "Some approaches to statistical and finite-state speech-to-speech translation," *Com-*

puter Speech and Language, vol. 18, no. 1, pp. 25–47, 2004.

[26] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Brisbane, Australia, 2008, pp. 2342–2345.

[27] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural Lattice-to-Sequence Models for Uncertain Inputs," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

[28] P. G. Moore, "The Estimation of the Poisson Parameter from a Truncated Distribution," *Biometrika*, vol. 39, no. 3/4, pp. 247–251, 1952.

[29] J. Costello, "On the number of points in regular discrete simplex," *IEEE Transactions on Information Theory*, vol. 17, no. 2, pp. 211–212, 1971.

[30] N. Smith and R. Tromble, "Sampling uniformly from the unit simplex," Johns Hopkins University, Tech. Rep., 2004.

[31] G. Kumar, G. Blackwood, J. Trmal, D. Povey, and S. Khudanpur, "A Coarse-Grained Model for Optimal Coupling of ASR and SMT Systems for Speech Translation," in *Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal, 2015, pp. 1902–1907.

[32] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-Sequence Models Can Directly Transcribe Foreign Speech," in *Annual Conference of the International Speech Communication Association (InterSpeech)*, Stockholm, Sweden, 2017.

[33] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, K. Duh, M. Faruqui, C. Gan, D. Garrette, Y. Ji, L. Kong, A. Kuncoro, G. Kumar, C. Malaviya, P. Michel, Y. Oda, M. Richardson, N. Saphra, S. Swayamdipta, and P. Yin, "DyNet: The Dynamic Neural Network Toolkit," *arXiv preprint arXiv:1701.03980*, 2017.

[34] Y. Gal and Z. Ghahramani, "A Theoretically Grounded Application of Dropout in Recurrent Neural Networks," in *Neural Information Processing Systems Conference (NIPS)*, Barcelona, Spain, 2016.

[35] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, Banff, Canada, 2014.

[36] M. Denkowski and G. Neubig, "Stronger Baselines for Trustable Results in Neural Machine Translation," in *The First Workshop on Neural Machine Translation*, Vancouver, Canada, 2017.