

NEURAL INVERSE TEXT NORMALIZATION

Monica Sunkara, Chaitanya Shivade, Sravan Bodapati, Katrin Kirchhoff

Amazon AWS AI

ABSTRACT

While there have been several contributions exploring state of the art techniques for text normalization, the problem of inverse text normalization (ITN) remains relatively unexplored. The best known approaches leverage finite state transducer (FST) based models which rely on manually curated rules and are hence not scalable. We propose an efficient and robust neural solution for ITN leveraging transformer based seq2seq models and FST-based text normalization techniques for data preparation. We show that this can be easily extended to other languages without the need for a linguistic expert to manually curate them. We then present a hybrid framework for integrating Neural ITN with an FST to overcome common recoverable errors in production environments. Our empirical evaluations show that the proposed solution minimizes incorrect perturbations (insertions, deletions and substitutions) to ASR output and maintains high quality even on out of domain data. A transformer based model infused with pretraining consistently achieves a lower WER across several datasets and is able to outperform baselines on English, Spanish, German and Italian datasets.

Index Terms— speech recognition, inverse text normalization, multilingual

1. INTRODUCTION

Inverse Text Normalization (ITN) is the process of converting spoken form of output from an automatic speech recognition (ASR) system to the corresponding written form. The written form is more commonly consumed by users and automated downstream processes. This includes simple entities such as cardinals, ordinals, and currencies, as well as complex entities involving addresses, email id, dates, times, and URIs. ITN is closely related to the task of text normalization where the goal is to convert text from written form to the spoken form [1].

Exploiting the power of neural networks for text normalization has been challenging until recently when recurrent neural networks (RNNs) were the state of the art [2]. While RNNs produced great results when evaluated using standard metrics, their errors made them infeasible for adoption in practical systems. More recently, [3] successfully employed a memory augmented neural network architecture that is independent of FSTs for text normalization. Similar success has been noted where neural generation models replaced rule based systems for tasks such as ASR spelling correction [4] and grammar correction [5, 6].

Although FST based approaches to ITN work well [7], they are expensive to scale across languages since native language speakers are needed to curate transformation rules. We therefore investigate applicability of neural network based models for ITN and evaluate them using WER specifically for ITN and non-ITN parts of the text. While RNNs are powerful for sequence to sequence tasks, transformer based models [8] offer pretraining abilities using vast

amounts of data. However, incorporating pretrained models is not trivial and is often specific to the task [9].

In light of the gaps with existing methods and resources for ITN, we present results with a neural approach towards the task. Specifically, we make the following contributions:

- Since data required for training ITN models is not publicly available and is hard to collect, we employ a data generation pipeline for ITN using a text to speech (TTS) frontend.
- We propose a robust neural ITN solution based on seq2seq models that can perform well across domains and noisy settings such as conversational data.
- We explore different ways to leverage pretrained models for ITN and demonstrate their superior performance across several domains and on ASR transcribed data.
- Finally, we show the first results for three of the European languages in mono-lingual and multi-lingual settings.

2. TEXT PROCESSING PIPELINE

In conventional approaches, ITN is addressed by using a set of rules through FSTs. However, neural models can be data hungry and may require good amount of labelled data for supervised learning. Collecting large amount of annotations at scale is prohibitive in terms of cost, resources, and time. Therefore, we employ FST-based text normalization methods to automatically normalize written form of text. This is similar to synthetic data generation employed successfully in the past [10]. However, the data prepared in such a way poses a number of problems for modeling ITN:

- Table 1 shows examples of spoken-form input and written-form output. As shown in the first example, there can be several variations of a written form of text, but text normalization techniques always use one fixed variation. This problem is further aggravated with variations in locale, and language.
- Text normalization systems often ignore several punctuation symbols since the goal is to produce spoken form of text. However, punctuation is often relevant in the written form. For example, if the TTS system omits quotation marks during the conversion, it is important to restore them in spoken form output for efficient modeling of ITN.
- Text normalization techniques may introduce errors when normalizing numbers or expanding some short forms, as the conversion requires to disambiguate based on the context. e.g., Lakeside Dr. is normalized as Lakeside drive where as Dr. John is converted as Doctor John.

To address first issue, we create synthetic data by randomly sampling from cardinal numbers related sentences and introducing additional variations required for modeling ITN in spoken-form of text. This synthetic data is augmented to the text data created using text normalization techniques for experiments presented in Section 5. We used Levenshtein edit-distance to align written form with spoken form in order to restore punctuation marks.

Spoken form	Written form
Two thousand one hundred five	
Two thousand one hundred and five	
Twenty one oh five	2105
Two one zero five	
Two one oh five	
october twenty twenty twenty	October 20, 2020
four percent of five dollars is twenty cents	4% of \$5 is 20 cents

Table 1. Examples of spoken-form input and written-form output

3. MODELS

We treat the traditional FST model as a baseline and discuss architectures for neural models. Finally, we describe a hybrid model for production settings.

3.1. Finite State Transducer

Our conventional baseline approach is a Finite State Transducer (FST) constructed using JFLAP¹. Each state in the FST performs a series of edits to the input string to get its corresponding written format output string. Our FST model covers a wide range of entities which do not require contextual understanding or disambiguation such as: Cardinals, Fractions, Ordinals, Years, floats, Date, Time, Currency, Units and Measurements.

3.2. Neural ITN

We model inverse text normalization as a sequence-to-sequence problem where the source and target are spoken and written form of text respectively. Our seq-to-seq baseline model is similar to [7] and uses Bahdanau content-based additive attention [11] to align the output sequence with input. We also implement a non-recurrent transformer based sequence to sequence model based on Vasvani et al., [12], where multi-head self-attention layers are used in encoder and decoder. For all our transformer models, the source and target sentences are segmented into sub word sequences.

Copy Attention: For a task like ITN, there exists a significant overlap between source and target sentences. Standard sequence-to-sequence models which rely on content based attention are often prone to undesirable errors such as insertions, substitutions and deletions during conversion. In order to reproduce the target close to the source sentence, we adapt the transformer based sequence-to-sequence models by utilizing a copying mechanism as described in [13]. The copy mechanism uses a generation probability to choose between source sentence vocabulary and a fixed target vocabulary thus allowing to generate out-of-vocabulary (OOV) words. The generation probability for each time-step is computed from context vector, transformer decoder self-attention output and the decoder input.

Pretrained models for ITN: Recently, pre-trained models such as BERT [14] and ELMo [15] have achieved tremendous success and lead to significant improvements on various natural language understanding tasks. These models trained on large amounts of unlabelled data capture rich contextual representations of the input sentence. In this work, we attempt two strategies to incorporate pretraining

into ITN: (1) To use a pre-trained sequence-to-sequence model and fine-tune it for ITN, and (2) Using a pre-trained masked language model like BERT as context-aware embeddings for ITN. For the first strategy, following [16], we initialize the encoder and decoder of a sequence-to-sequence model with a pre-trained BART model, and then fine-tune the model on ITN datasets. Unfortunately, we did not observe good performance. For the second strategy, following the practice of [9], we use BERT to extract context-aware embeddings and fuse it into each layer of transformer encoder and decoder via an attention mechanism.

3.3. Hybrid solution

Despite incorporating the mechanisms we discussed in previous section, sequence-to-sequence models are still prone to some errors. Also, in a real world application, neural models require a good number of training examples to handle each new entity. In order to overcome these issues, we propose a novel hybrid approach combining neural ITN with an FST, where the spoken form output of ASR system is first passed through the proposed neural ITN model, followed by an FST. A confidence score emitted by the neural ITN model is used as a switch to make a decision in the run time whether to use neural ITN output. This solution has a three-fold advantage: (1) Having neural ITN as the first pass, helps to normalize the input sentence well based on a deeper contextual understanding, (2) With FST as the final component, we can correct any common neural ITN errors by adding rules to FST, and (c) In the case of any unrecoverable mistakes by ITN in the run time, the confidence score can be used as a safety measure from outputting erroneous neural ITN output.

4. EXPERIMENTAL SETUP

4.1. Data

All our experiments were conducted on data extracted from publicly available Machine Translation, Speech Translation, and Text Summarization data sources. For monolingual English models we used Wikipedia data subsampled from the publicly available release of parallel written-spoken formatted text from [2]. We also test our models on additional domains such as crawl, ted talk, and news. For this purpose, we extracted data from ParaCrawl, NewsCommentary (News-C) data sets provided by Shared Task: Machine Translation of News² and MuST-C dataset release from [17] and CNN-DailyMail data sets from DeepMind QA Dataset³. For multilingual experiments, we used the data extracted from ParaCrawl, NewsCommentary (News-C) and MuST-C [17] for German, Spanish and Italian languages.

We prepare parallel written-spoken form for our datasets in the following manner. We extract the text in written form from public sources and generate the corresponding spoken form using FST based text normalization of a commercial text to speech service⁴. Finally, we augment the English data with synthetically generated data as described in section 2. The English model data splits contain 600k, 130k sentences for training and tuning respectively. The Multilingual models train and tune splits contain 200k, 25k sentences respectively. Apart from the above mentioned datasets, as a check for robustness, we also evaluate our monolingual English models on ASR outputs (500 sentences each for broadcast and conversational

¹<http://www.jflap.org/>

²<http://www.statmt.org/wmt20/translation-task.html>

³<https://cs.nyu.edu/~kcho/DMQA/>

⁴<https://aws.amazon.com/polly>

Model	Syn	WER	I-WER	NI-WER
FST	✗	14.4	103.8	0.2
RNN-Word	✗	12.9	19.8	0.9
RNN-Subword	✗	2.2	14	0.4
Transformer	✗	1.3	6.2	0.6
	✓	1.1	5.2	0.4
Transformer				
+ Copy Attention	✓	1.0	5.2	0.3
+ BERT-fusion	✓	0.9	4.8	0.3
BART	✓	14.5	72.7	5.2

Table 2. Comparison of models on Wikipedia test set.

domains) generated from private data sets. The total number of ITN entities is 5% of words in the test data.

4.2. Metrics

Previous work on inverse text normalization and related tasks measures the performance of models by computing word error rate (WER) on the entire target sentence. In our work, we introduce two additional metrics namely ITN WER (I-WER) and Non-ITN WER (NI-WER) focusing on the quality of text normalization and to measure undesirable perturbations to original source text respectively. The metrics are computed as follows: First, each word in the written form reference in tagged with ITN or N-ITN indicating if a word should be normalized or should be copied to target respectively. In order to perform the tagging, we align the source (spoken) and reference (written) sentences using levenshtein distance. The words which are common in both sentences are tagged as N-ITN and the rest as ITN in the reference. ITN WER (I-WER) is computed by summing up the substitutions, insertions, and deletions between the target hypothesis (ITN output) and reference and dividing it with number of reference words considering only those tagged as ITN. Non-ITN WER (NI-WER) is computed similarly over only N-ITN tagged words. The overall WER is computed across all the reference words.

4.3. Model Configurations

As sequence-to-sequence baselines, we implement a word and subword based recurrent neural network models similar to [7]. The RNN model consists of a 2-layer bi-directional long short term memory network (bi-LSTM) encoder and a 2-layer LSTM decoder, with 512 hidden states for the bi-LSTM. Decoding uses the attention mechanism from [11]. The second sequence to sequence architecture is a subword transformer model [12] with 12 layers. We also implement two additional mechanisms in to our subword transformer model. The first mechanism is copy attention mechanism similar to [13]. The second mechanism is BERT-Fusion into the transformer based encoder and decoder layers following [9]. For English experiments we use bert-base-uncased to extract pretrained representations for an input sequence, whereas for multilingual experiments we use bert-base-multilingual-uncased⁵. In our experiments, for a given sentence, subwords are generated through a model trained using the SentencePiece toolkit⁶. We use a subword

⁵<https://github.com/bert-nmt/bert-nmt/blob/master/bert/modeling.py>

⁶<https://github.com/google/sentencepiece>

Dataset	ASR	Model	WER	I-WER	NI-WER
CNN	✗	Copy Att	2.0	29.7	0.4
		BERT-fusion	2.2	19.5	0.6
DailyMail	✗	Copy Att	2.4	17.8	0.4
		BERT-fusion	1.7	3.9	0.4
News-C	✗	Copy Att	1.5	19.4	0.2
		BERT-fusion	1.2	10.4	0.3
MuST-C	✗	Copy Att	2.7	23.2	1.4
		BERT-fusion	3.3	20.4	1.7
Paracrawl	✗	Copy Att	7.9	29.2	0.8
		BERT-fusion	3.9	1.8	3.9
BCN	✓	Copy Att	3.3	53.3	0.7
		BERT-fusion	3.2	44.5	1.1
TEL	✓	Copy Att	5.3	65.4	1.5
		BERT-fusion	4.3	46.9	1.6

Table 3. Performance across various domains including News, Ted Talks, Crawl etc. inventory size of 16k⁷ and a beam size of 5 for all our experiments.

5. RESULTS

Table 2 summarizes our experiments on Wikipedia dataset. We report an overall word error rate (WER) at sentence level, ITN WER (I-WER) and Non ITN WER (NI-WER). First, we compare the performance of the baseline Finite State Transducer (FST) model, word-based RNN, subword RNN and Transformer models. The RNN subword model performs better than the word based RNN and FST, with a significant relative improvement of 84.7% in terms of WER when compared to FST model. We must state that the FST model does not cover all normalization entities in the Wikipedia data leading to a higher I-WER. The transformer model outperforms others with a WER of 1.3. In addition, synthetic data augmentation leads to an improvement in both ITN and non-ITN WER and there by a relative overall WER improvement of 18%. We then present the experiments on transformer model by incorporating additional mechanisms such as copy attention and BERT-fusion for further improvement. Copy attention leads to a relative improvement of 25% in non-ITN WER. This is expected, as copy mechanism helps in reducing the out of vocabulary (OOV) problem by copying the words from source to text and enforces the target sentence to remain close to the source. Transformer model fused with pretrained BERT representations shows improvement in both ITN and non-ITN WER performing the best of all models. Since the BERT model is pretrained on large amounts of written text from public sources such as Wikipedia, fusing its representations into transformer model seems to result in gains than models trained from scratch. Also, since we trained and tested on the same domain data, a transformer with BERT-fusion performs comparably well even without copy mechanism. This motivated us to extend our experiments and test the performance of neural ITN models on several other domains as illustrated in section 5.1.

We also validate using a pre-trained sequence-to-sequence model and fine-tuning it for ITN task would help. For this purpose, we finetuned a pretrained BART model on ITN data and evaluated its performance on Wikipedia dataset. Unfortunately, we did not observe good performance by using the BART model out of the box. The decoder outputs had several insertions and failed to normalize for even simple entities.

⁷In our ablation study experiments conducted in the range of 2k-32k subwords, 16k gave best results on Wikipedia

Usecase	BCN			TEL		
	FST	Neural	Hybrid	FST	Neural	Hybrid
Numbers	2.8	2.6	2.5	5	4.5	4.2
Units	7.5	5.6	5.4	4.4	3.6	3.4
Date Time	5	1.8	1.6	7	6	5.8
Misc	10	3.5	3.2	15.6	11.1	11.1

Table 4. Performance on ASR outputs (Broadcast and Conversational) in terms of WER (%).

5.1. Towards a real world system

We carried out evaluations of our best neural ITN models namely copy attention and BERT-fusion across several domains. Results presented in Table 3 show that transformer model with copy attention (Copy Att) consistently outperformed BERT-fusion model w.r.t Non ITN WER (NI-WER) across all the domain testsets. Copy attention seems to play an effective role in copying the source sentence words into the target for out of domain datasets and OOV words, whereas using pretrained embeddings seem to be of limited help. On the other hand, w.r.t ITN WER (I-WER) BERT-fusion significantly outperformed copy attention across all test sets. This explains that, fusion with representations extracted from a model pretrained on large written datasets helps in improving the normalization. Moving towards a real world system, we evaluate on ASR outputs from broadcast (BCN) such as news, media etc and conversational (TEL) domains. BERT-fusion performs better than copy attention on both the datasets due to significant improvements in ITN WER. These observations motivated us to further explore their performance across different entities. We compare entity wise performance of FST, best neural ITN model (BERT-fusion) and hybrid solution described in section 3.3. Though neural ITN model significantly outperforms FST (see Table 4) across all entities, the model outputs often contain some common errors. Also, the higher WER for FST in Misc category shows the necessity to define manually curated rules for each entity. However, using a hybrid approach, we observe that few of the recoverable common errors can be corrected, there by leading to an additional improvement in performance. We also present some qualitative examples from our best model (BERT-fusion) in Table 5.

Input: contact number for us is *one eight hundred two five five seven eight two eight*

Output: contact number for us is *1-800-255-7828*

Input: any time not not at *noon* or *two* or *four*

Output: any time not not at *12:00 pm* or *2:00* or *4:00*

Input: it was priced at *three thousand six four nine point eight four dollars*

Output: it was priced at *\$3649.84*

FST: it was priced at *3006 4 \$9.84*

Input: *ten twenty nine* gmt november *twenty ninth* *twenty twelve*

Output: *10:29* gmt november *29* *2012*

Input: we think certainly *ten* to *fifteen thousand* dollars a month

Output: we think certainly *\$10-\$15000* a month

Input: *florida* *three three nine six oh* yusa or phone *nine four one four six five four three two one* fax *four six*

Output: *fl 33960* yusa or phone *941-465-4321* fax *46*

Table 5. Example utterances from Neural ITN outputs

5.2. Performance of Multilingual Neural ITN models

We extend our experiments to other European languages such as German, Spanish and Italian. For each language, we train two models: a vanilla transformer and a transformer model with BERT-fusion. In addition, we also train a multilingual model on data combined from all three languages. During training and inference, we add an artificial language token prefix to each sentence in both source (spoken-form) and target (written-form) similar to [18].

The results are presented in Table 6. From the results, we observe that BERT-fusion consistently outperformed vanilla Transformer and the multi-lingual model outperformed the models trained on data from each individual language. It is clear that the multilingual model is greatly benefited from having lexical similarity among all three languages trained. Also, from the qualitative analysis of the results, we infer that the coverage of subword units for each monolingual model is vastly improved in multilingual model by combining data from all three languages, thus leading to fewer number of OOVs.

Model	German	Spanish	Italian
Transformer	3.5	6.1	2.3
+ BERT-fusion	2.7	5.2	2.2
+ Multi-lingual	2.1	5	1.6

Table 6. Performance of neural ITN on multiple languages in terms of WER (%)

6. RELATED WORK

While ITN as a problem is not new [19], work addressing the task using end to end neural sequence generation has been very limited. Most recently, [20] used pointer generator networks for ITN in Japanese. They limit their evaluation to a single dataset and note that the copy mechanism from the architecture [13] is key to performance improvements. ITN is one of the many tasks addressed in the larger problem of ASR post-processing for readability (APR) proposed in [21]. They compare several transformer based models across different openly available datasets and show that RoBERTa [22] is a better model in most scenarios. While this is the closest study to our work, their evaluation is for the APR task which has a wider scope but is limited to monolingual English. In [23], a three step approach is proposed for modeling ITN as a labeling problem, first identifying the candidates for transformation, second mapping the spoken candidates to written form, and finally applying post-processing if necessary. They infuse machine learning into rule based models by adopting LSTMs for the first step and using FSTs for the second.

7. CONCLUSION

We introduced a neural ITN approach leveraging pretrained models that can perform well across various domains covering more than 30 entities. Our results suggest that the proposed neural ITN solution can significantly outperform traditional FST based solutions on ASR outputs of unstructured datasets like Conversational domain. We have presented entity wise performance analysis and a hybrid solution for combining FST with neural ITN to minimize any common recoverable errors made by pure neural models. We showed that our solution can be easily extended to other languages by presenting the first multilingual ITN results, without the need for a linguistic expert. Future work will focus on combining copy attention with pretrained models to reap benefits from both the approaches.

8. REFERENCES

- [1] Jonathan Allen, M Sharon Hunnicutt, Dennis H Klatt, Robert C Armstrong, and David B Pisoni, *From text to speech: The MITalk system*, Cambridge University Press, 1987.
- [2] Richard Sproat and Navdeep Jaitly, “RNN approaches to text normalization: A challenge,” *arXiv preprint arXiv:1611.00068*, 2016.
- [3] Subhojeet Pramanik and Aman Hussain, “Text normalization using memory augmented neural networks,” *Speech Communication*, vol. 109, pp. 15–23, 2019.
- [4] Jinxi Guo, Tara N Sainath, and Ron J Weiss, “A spelling correction model for end-to-end speech recognition,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5651–5655.
- [5] Oleksii Hrinchuk, Mariya Popova, and Boris Ginsburg, “Correction of automatic speech recognition with transformer sequence-to-sequence model,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7074–7078.
- [6] Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeol Yoon, “A neural grammatical error correction system built on better pre-training and sequential transfer learning,” in *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2019, pp. 213–227.
- [7] Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister, “Neural text normalization with subword units,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, 2019, pp. 190–196.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.
- [9] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu, “Incorporating bert into neural machine translation,” *arXiv preprint arXiv:2002.06823*, 2020.
- [10] Cal Peysers, Hao Zhang, Tara N Sainath, and Zelin Wu, “Improving performance of end-to-end asr on numeric sequences,” *Proceedings of Interspeech*, pp. 2185–2189, 2019.
- [11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [13] Abigail See, Peter J Liu, and Christopher D Manning, “Get to the point: Summarization with pointer-generator networks,” *arXiv preprint arXiv:1704.04368*, 2017.
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.
- [15] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, “Deep contextualized word representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.
- [16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *arXiv preprint arXiv:1910.13461*, 2019.
- [17] Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi, “Must-c: a multilingual speech translation corpus,” in *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019, pp. 2012–2017.
- [18] Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al., “Google’s multilingual neural machine translation system: Enabling zero-shot translation,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.
- [19] Maria Shugrina, “Formatting time-aligned asr transcripts for readability,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, USA, 2010, HLT ’10, p. 198–206, Association for Computational Linguistics.
- [20] M. Ithori, A. Takashima, and R. Masumura, “Large-context pointer-generator networks for spoken-to-written style conversion,” in *Proceedings of ICASSP*, 2020, pp. 8189–8193.
- [21] Junwei Liao, Sefik Emre Eskimez, Liyang Lu, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng, “Improving readability for automatic speech recognition transcription,” *arXiv preprint arXiv:2004.04438*, 2020.
- [22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019.
- [23] Ernest Pusateri, Bharat Ram Ambati, Elizabeth Brooks, Ondrej Plátek, Donald McAllaster, and Venki Nagesha, “A mostly data-driven approach to inverse text normalization,” in *INTER-SPEECH*, 2017.