



# Seq-2-Seq based Refinement of ASR Output for Spoken Name Capture

Karan Singla, Shahab Jalalvand, Yeon-Jun Kim, Ryan Price, Daniel Pressel, Srinivas Bangalore

Interactions, LLC, New Providence, NJ, USA

{ksingla, sjalalvand, ykim}@interactions.com

## Abstract

Person name capture from human speech is a difficult task in human-machine conversations. In this paper, we propose a novel approach to capture the person names from the caller utterances in response to the prompt "say and spell your first/last name". Inspired from work on ASR spelling correction models, disfluency removal and text normalization, we propose a lightweight Seq-2-Seq system which generates a name spellings from varying user input. Our proposed method outperforms a strong baseline which is based on a LM-driven rule-based approach.

**Index Terms:** spoken language understanding, name capture, Seq-2-Seq neural network

## 1. Introduction

In order to authenticate a user and provide personalized services, most enterprise virtual agents (EVA, also known as Intelligent virtual agents (IVA)) rely on capturing the name of the user accurately. However, for a multinational enterprise with a customer base varied in nationality and accents, the challenge of extracting the name of a customer from their spoken utterance accurately is immensely challenging. With repeated re-prompting in an attempt to capture the user's names, most virtual agents deliver a frustrating user experience. In addition to the large vocabulary size of names, the problem is compounded by homophonic names – names that sound the same but have distinct orthography [1, 2, 3, 4]. While general purpose transcription of speech has seen significant advances in the past decade, high accuracy recognition and extraction of names remains a persistent issue.

A conventional pipeline of speech recognition (ASR) – typically using name lists encoded in grammars, and/or supplemented with statistical language models (SLMs) – are followed by spoken language understanding (SLU) systems that extract names from the user's response to a prompt from a virtual agent. In order to minimize the error in recognition and extraction of names, designers of EVA speech interfaces often design prompts that request the user not only say their first or last name but spell it as well. With the spelling of names, similar sounding names such as "Stuard", "Stuart", and "Stewart", might be correctly captured, as attested to by the experimental results in this paper. Such techniques are employed in human-human conversations as well, in order to minimize errors in the capture of names between interlocutors.

Spoken person name recognition combines ASR and NLP and traces its history to the early nineties with the first collection of spoken and spelled names over the telephone in 1992 [5] in English and more recently, in languages such a Dutch [6]. There have been several approaches [7] that leverage the spelled letters to alleviate the problem of open vocabulary nature of name recognition. [8] leveraged spelled letters to alter the character sequence using a Finite State Machine (FSM) based approach.

**Machine:** Say your first / last name and spell it.

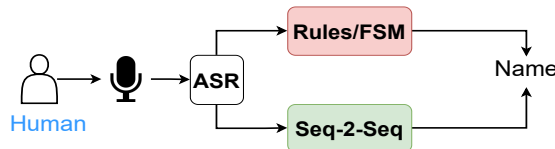


Figure 1: Automatic name capture pipeline using sequence-2-sequence system instead of FSM or rule-based approaches.

They use the forward-backward algorithm to learn transition probabilities for editing spelled characters. An edited spelled name is then derived using the A\* search algorithm. They reject 10-15% data with high conversational variation (including descriptors like *as in, like*) as a part of incorrect data.

Different designs have been proposed to capture names in dialogue. [9] present usability evaluation of three different dialogue designs including say only, one-stage say and spell and two-stage say and spell. [10] use a combination of character description recognition and syllable spelling recognition to predict Chinese names. The most recent work on spoken say and spelled name recognition [11] proposed the use of a dynamic hierarchical language model to accurately recognise the spelling and an LM-driven rule-based approach is used to predict the name. While these research works used hybrid ASR systems (a combination of neural network-based acoustic model with an n-gram language model), a recent approach [12] modified the loss criteria to specifically emphasize proper noun recognition in the context of an end-to-end (E2E) ASR system.

In recent work [13, 14], Seq-2-Seq models have successfully been used for correction of ASR errors resulting from E2E ASR models. In such approaches, the first-pass hypothesis from ASR is given to an attention-based, encoder-decoder Seq-2-Seq model which outputs a final version of the hypothesis, correcting any recognition errors, if present. Our proposed approach also applies Seq-2-Seq to the text hypothesis output by an E2E ASR model in the first pass. However, unlike previous works for general correction of ASR errors, our approach both *corrects and extracts* spoken names from the ASR hypothesis. Concretely, our approach generates a character sequence of the name from an ASR hypothesis which may contain more than the just the name and spelling.

Our proposed Seq-2-Seq modeling for name capture is a light-weight adaptation of the standard transformer based Seq-2-Seq architecture, similar to the one proposed by [15] for neural machine translation. This Seq-2-Seq system learns using (*asr hypothesis, name*) pairs recorded from an enterprise-grade virtual assistant. We believe the proposed approach can be generalized to capture other difficult entities such as emails, and addresses from user's speech to a virtual agent.

## 2. Automatic Name Capture

The goal of our work is to capture a user’s name from their spoken utterance that is in response to a virtual agent’s prompt: *Please say and spell your first/last name.* The table below shows several illustrative examples of ASR hypotheses and the intended names, demonstrating the need for an automatic name capture system that can not only correct errors in the character sequence produced by ASR, but also ignore or interpret the use of the NATO phonetic alphabet and other complex patterns associated with the disambiguation of spelled letters.

ASR Hypothesis	Name
jennifer j e n n i s e r	jennifer
d a r e n darren	daren
sara s a r a last name weber w e v e r	sara
v a s victor e r a s robert azera	vera

Table 1: *Sample name capture data.*

Our system is a pipeline combining ASR and an NLU module. The ASR recognizes the user’s utterance to generate N-best hypotheses and the NLU module extracts the caller’s name from the hypotheses. In this paper, we compare two ASR architectures and two different NLU approaches for name capture. In addition, we experiment with different confidence scoring methods to improve the precision of name extraction by rejecting the low confidence results.

### 2.1. Seq-2-Seq models to translate ASR Hypothesis to Names

Our Seq-2-Seq system generates a spelled name from the ASR hypothesis. We adopt a standard transformer based encoder-decoder setup to train from (*ASR hypothesis*, *Name*) pairs. The ASR hypothesis is provided in the form of BPE tokens as input to the Seq-2-Seq model, while the decoder generates a character sequence representing the name. We found that generating names as a character sequence performs better than using BPEs for the output sequence. We use a shared embedding layer for both encoder and decoder tokens.

Our encoder-decoder setup is an adaptation of [15] originally proposed for neural machine translation. The encoder is comparatively lightweight with a stack of  $N(=2)$  identical layers. Each layer first has a multi-head self-attention mechanism [16, 17] with 2 heads, and the second is a simple, position-wise, fully-connected, feed-forward network. Similar to [15], we employ a residual connection [18] around each of the two sub-layers. But instead of normalizing after each sub-layer, following [19], we perform PreNorm where we perform layer normalization before attention layers and before the fully connected dense layer. To facilitate residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs with a hidden dimension of 64. We use fastBPE [20] to learn shared vocabulary for both encoder and decoder.

The decoder is also composed of a stack of  $N(=2)$  identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Similar to the encoder, we use normalization employ residual connections around each of the sub-layers. We also modify the self-attention sub-layer in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with the fact that the output embeddings are offset by one position, ensures that the predictions for position  $i$  can depend only on the

known outputs at positions less than  $i$ . We use Adam with a fixed batch size of 32 and with a fixed learning rate of  $1.0e - 5$ . We do not perform any pre-training and instead train our system from a random initialization.

In the Seq-2-Seq based approach, the confidence-score is the log-probability assigned to the character name sequence.

### 2.2. Baseline: LM Driven Rule-based Approach

For our baseline system, we simply concatenate the spelled letters recognized by ASR and filter out other carrier phrases in the ASR hypothesis, in order to capture the name [11]. The confidence score for this prediction is computed by averaging the ASR word-confidence scores of the spelled letters. If the predicted name matches any of the words in the utterance, then we consider it to be a high confidence prediction, and as a result set the confidence score of the prediction to 1.0. For example, if the ASR hypothesis is **”john/0.01 j/0.7 o/0.6 n/0.5 e/0.8”**, then the predicted label/score is **”jone/0.65”**.

The overall process is as follows:

1. Input the ASR hypothesis
2. Find the spelled letters and concatenate them to predict the name
3. Compute the confidence score by averaging the ASR scores of each spelled letters
4. Set the confidence score to 1.0, if the predicted name matches a recognised word in the hypothesis
5. If there is no match using the 1-best hypothesis, then try the 2-best and 3-best hypotheses

However, there are some major drawbacks to this approach. The first drawback is its dependency on the letters. If the caller says **”tim t as in tango i as in i m as in man”**, then the recognized name will be **”tiim”**. Also the confidence scores become noisy as they also account for additional words which are not characters like **”tim, in, tango, as”**. Despite its simplicity, this approach is shown to be efficient and effective in our experiments, making it quite suitable for real-time EVA applications.

## 3. Experiments and Results

We collect say and spell name data set from production IVA systems across a range of enterprise-grade virtual agents and measure the performances of different state-of-art techniques for name capture. We compare our proposed approach to a rule-based approach for automatic name capture.

### 3.1. Dataset

While some public datasets like the OGI collection [1] include a small subset of spelled names, with the growing demand for EVAs for multiple industry verticals, we have millions of user utterances responding to the prompt *please say and spell your name* which have been annotated with the name spoken in the utterance.

We collect audio samples with corresponding name labels for both first name and last name capture tasks. In total, we collect 101K samples for first name and 580K samples for last names. Table 2 shows the statistics of the first and last name datasets, which illustrates that the variety of person names is quite large both in first names (26.2K) and last names (25.7K). Consequently, it is not feasible to use a simple classifier to predict the names as shown in [11]. These datasets are collected

from different applications including banking, insurance and retail from callers based in USA. We use 90% of this data for training and 10% for validation purposes. The number of words in the training and development sets are average words in the ASR hypothesis from different ASR systems used in this paper and IA labels. For test purposes, we employ native speakers of English to annotate name labels. In total, there are 580 and 856 unique samples for first and last names respectively in the test set. We found there were around 10-15% labeling errors corrected by annotators in the test set. For ASR model training purposes, we also collect additional 400 hours of human transcribed IVA data.

Dataset	#Utterances	#Words	#UniqueNames
IVA (In-domain)	570K (800hrs)		–
firstname-train	89K	1.2M	26.2K
firstname-dev	9.8K	138.9K	5K
firstname-test	835	7.2K	580
lastname-train	522K	7.6M	25.7K
lastname-dev	58K	882.5K	12.6K
lastname-test	1K	9.2K	856

Table 2: Data statistics.

### 3.2. ASR performance

To study the impact of ASR model selection on word accuracy and ultimately performance on the downstream task of name capture, we compare two different ASR modeling approaches for transcribing spoken name utterances. First, we use a traditional hybrid DNN-HMM acoustic model with 4-gram language model and an FSM based decoder [21, 22]. Our second ASR system is a state-of-the-art end-2-end (E2E) deep residual convolutional neural network architecture [23]. It utilizes a Google sentence-piece [24] tokenizer with vocabulary size of 1024, and transcribes text in the lower case English alphabet along with spaces, apostrophes and a few other characters. This model is pre-trained on 7000 hours of publicly available transcribed data [23].

Based on ASR architectures and training corpora, we have the following acoustic models to evaluate.

- **Hybrid-in-domain-AM** consists of hybrid DNN acoustic models trained to predict tied context-dependent triphone HMM states with cross-entropy and sequential loss functions using 81-dimensional log-spectrum features. This model is trained on  $\approx$  400 hours of transcribed IVA speech data. We found that using in-domain dataset from IVA shows low error rate than using general purpose datasets available for ASR training.
- **E2E-pretrained-AM** is the off-the-shelf Citrinet model [23] from Nemo toolkit [25] with beam-search decoder (beam=64). No language model is used to re-score the hypotheses.
- **E2E-finetuned-AM** refers to the same Citrinet model, but fine-tuned on the  $\approx$  400 hours of in-domain transcribed IVA dataset, same with *Hybrid-in-domain-AM* (refer Table 2).

We also experiment with two different language models:

- **In-domain-LM** refers to a language model which is an interpolation of five 4-gram Katz backoff models [26] each trained on a specific IVA vertical including banking, insurance, retail, hospitality and telecommunication.

In total, we use a large text corpora which is obtained using an in-house production ASR containing 569 million words accounting for 41K unique words.

- **Names-LM** is a similar 4-gram Katz backoff model which uses additional data of name and spell capture described in Table 2, along with In-domain IVA corpora. As shown in Table 2, using this LM shows relative improvement of 1-3% in ASR Word-Error-Rate (WER) for name capture for both Hybrid and E2E ASR.

The fine-tuned E2E ASR shows significant error reductions from 27.6% to 9.7% for first name capture and 32.7% to 7.9% for last name capture. This highlights the need to fine-tune E2E ASR models on in-domain datasets with supervised transcription. Using *name-LM* for rescoring the hypotheses significantly reduces the WER further, for the name capture test set by 0.5%-1.3%.

	AM	LM	% Word Error Rate	
			FirstName	LastName
<b>Hybrid ASR</b>	In-domain	In-domain	17.7	19.1
	In-domain	Names	14.7	19.1
	Pretrained	No LM	27.6	32.7
<b>E2E ASR</b>	Fine-tuned	No LM	9.7	7.9
	Fine-tuned	Names	<b>9.2</b>	<b>6.6</b>

Table 3: WER results on first and last name utterances.

In our experiments we attempted to fine-tune the AM with data from an additional name and spell  $\approx$  semi-supervised set. For this semi-supervised set, we choose the samples in which the concatenation of the spells match the name labels. We found that hybrid ASR performance improves for say and spell utterances but E2E ASR performance does not. Hence, we only use supervised in-domain data for AM training. We also found that none of the above-mentioned ASR systems do a good job in recognising the name as a whole word because most of these names are not in-vocabulary for the language model nor are in the training set of the E2E ASR, consequently leading to errors shown in Table 3. Thus, callers are directed to spell their names, since spelling recognition is more accurate, resulting in more reliable ways to predict the name [27].

The WER results in Table 3 show that the E2E ASR outperforms the traditional hybrid ASR when the E2E model is fine-tuned with in-domain data. Even without using a language model, the fine-tuned E2E ASR provides 5-11.2% absolute reduction in WER compared to the hybrid ASR.

### 3.3. Results for name capture

Table 4 shows error rates for name capture on the first and last name test data. We achieve better results for name capture using the ASR system which had lower error rates, thus highlighting the importance of a high accuracy ASR system for name capture. Our proposed Seq-2-Seq approach performs significantly better than the LM-driven rule-based approach for the different ASR systems we compared. When dealing with hybrid ASR hypotheses, Seq-2-Seq captures names in 3-8% more utterances correctly than LM-driven approach does. Similarly, the name capture pipeline using E2E ASR shows 3-7% relative improvement over the baseline. We notice that gains of Seq-2-Seq over LM-driven approach becomes larger, when ASR performance is poor.

The LM-driven approach is highly dependent on ASR output, thus if the WER is high then name capture error will also

	AM	LM	FirstName		LastName	
			LM-driven	Seq-2-Seq	LM-driven	Seq-2-Seq
<b>Hybrid ASR</b>	In-domain	In-domain	37.8	<b>29.2</b>	31.0	<b>24.8</b>
	In-domain	Names	30.9	<b>24.4</b>	16.3	<b>13.1</b>
<b>E2E ASR</b>	Pretrained	No LM	53.9	<b>46.7</b>	62.5	<b>48.3</b>
	Fine-tuned	No LM	26.3	<b>22.2</b>	12.2	<b>11.8</b>
	Fine-tuned	Names	25.3	<b>21.9</b>	10.0	11.0

Table 4: 1-best classification error rate scores.

be high. However, Seq-2-Seq seems to recover frequent ASR errors to capture names correctly. We believe seq-2-seq errors can be further reduced if the quality of training data is improved. Our study showed our training data has around 10-15% labeling errors made by human annotators.

#### 4. Observations

Table 5 shows examples where the seq-2-seq approach correctly filters input, corrects ASR errors and generates the correct letter sequence to capture names. It also shows examples where Seq-2-Seq performs worse than the baseline. We observe that the Seq-2-Seq system for first name capture can filter-out first names from the full name provided by speaker.

ASR hypothesis (Input)	LM-driven	Seq-2-Seq
b as in boy o w d as in dog i c as in cat h	bowdich	bowdich
jennifer j e n n i s e r	jenniser	jennifer
r o s l i n d r i s l i n r a n k i n franks	roslindrakin	roslind
r i r i p p e e	ririppee	rippee
s d o v s e d o z	sedoz	sedov
u m b a s k a l b a s c a l	bascal	bascal

Table 5: Sample output for automatic name capture. Green means correct, red means name captured wrong.

In order to maximize the user experience of an EVA application, we have a human-in-the-loop solution which recruits humans when a model confidence is inadequate for automation. The confidence threshold determines the error versus rejection curve and a suitable operating point is chosen that optimizes the rejection at a given error rate.

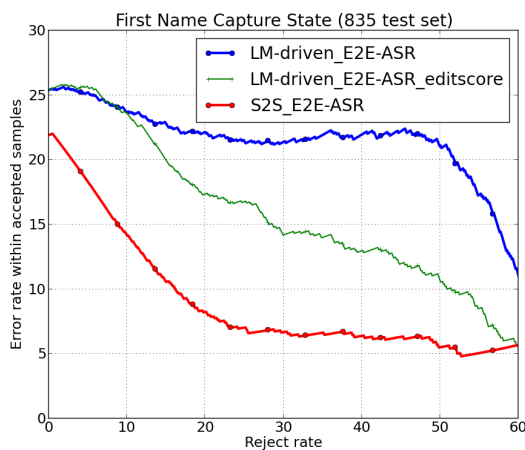


Figure 2: Error vs Reject rate curve for First Name Capture.

Figures 2 and 3 show the Error-Rejection (ER) curve for the test set. We sort all the samples with the confidence score for

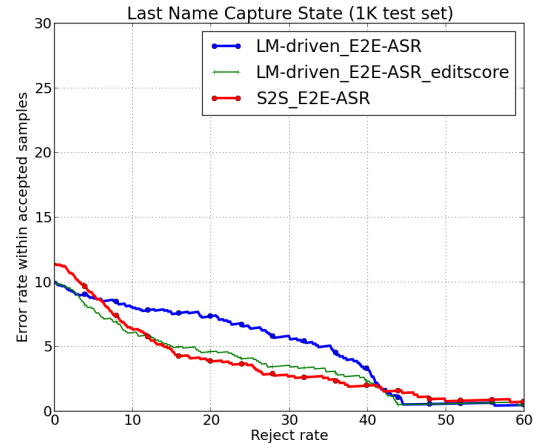


Figure 3: Error vs Reject rate curve for Last Name Capture.

each name capture approach, then plot variation in error rates by removing samples with a given confidence score.

The ER curve for first name capture shows Seq-2-Seq significantly improves over the baseline. For a low rejection rate of 20% Seq-2-Seq performs at a significantly lower error rate of 8% vs 21.5% for the LM-driven approach. Similarly, for the last name capture, Seq-2-Seq shows significantly lower error rate at 20% rejection rate. The big gap in performance could be due to the fact that confidence scores from LM-driven approach become noisy as they account for additional words which are not characters, as in "tim, t, as, in, tango", unlike Seq-2-Seq which generates a letter sequence of caller's name. However when we replace LM-driven's confidence score with edit distance between the spelled name and first word in the utterance, LM-driven's performance improves (Blue vs. Green ER curves). This supports our earlier hypothesis that additional words in ASR hypothesis add noise to the confidence scores of LM-driven approach for name capture.

#### 5. Conclusions and Future Work

Seq-2-Seq shows promising results to automatically capture first and last names from spoken utterances. We believe our results can have direct gains from exploiting ASR lattice output in form of N-best lists. We have not as yet extensively explored different hyperparameters, thus, sweeping over certain model parameters can also lead to obvious improvements as well.

We believe the biases in user demographics can effect how we speak [28], therefore, we plan to do a wider and more thorough evaluation in the future. We will extend this approach to email and address capture prompts as we continue to explore Seq-2-Seq modeling techniques for data capture from spoken utterances.

## 6. References

- [1] R. A. Cole, M. Noel, T. Lander, and T. Durham, "New telephone speech corpora at cslu." in *Eurospeech*. Citeseer, 1995, pp. 1–4.
- [2] D. Yu, K. Wang, M. Mahajan, P. Mau, and A. Acero, "Improved name recognition with user modeling," in *Eighth European Conference on Speech Communication and Technology*, 2003.
- [3] H. Raghavan and J. Allan, "Matching inconsistently spelled names in automatic speech recognizer output for information retrieval," in *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada: Association for Computational Linguistics, Oct. 2005, pp. 451–458. [Online]. Available: <https://www.aclweb.org/anthology/H05-1057>
- [4] T. Bruguier, F. Peng, and F. Beaufays, "Learning personalized pronunciations for contact names recognition," in *Interspeech*, 2016.
- [5] R. A. Cole, K. Roginski, and M. A. Fanty, "A telephone speech database of spelled and spoken names." in *ICSLP*, vol. 92, 1992, pp. 891–895.
- [6] H. Heuvel, J.-P. Martens, K. D'hanens, and N. Konings, "The autonomata spoken names corpus," 2008.
- [7] H. Hild and A. Waibel, "Recognition of spelled names over the telephone," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 1. IEEE, 1996, pp. 346–349.
- [8] D. Jouvet and J. Monné, "Recognition of spelled names over the telephone and rejection of data out of the spelling lexicon," in *Sixth European Conference on Speech Communication and Technology*, 1999.
- [9] N. Davidson, F. McInnes, and M. A. Jack, "Usability of dialogue design strategies for automated surname capture," *Speech Communication*, vol. 43, no. 1, pp. 55–70, 2004.
- [10] N. Wang, C.-H. Tsai, P. Huang, and J.-L. Shen, "Chinese large-vocabulary name recognition system using character description and syllable spelling recognition," in *2004 International Symposium on Chinese Spoken Language Processing*, 2004, pp. 17–20.
- [11] R. Price, M. Mehrabani, N. Gupta, Y.-J. Kim, S. Jalalvand, M. Chen, Y. Zhao, and S. Bangalore, "A hybrid approach to scalable and robust spoken language understanding in enterprise virtual agents," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*. Online: Association for Computational Linguistics, Jun. 2021, pp. 63–71. [Online]. Available: <https://aclanthology.org/2021.naacl-industry.9>
- [12] C. Peyser, T. N. Sainath, and G. Pundak, "Improving proper noun recognition in end-to-end asr by customization of the mwer loss criterion," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7789–7793.
- [13] J. Guo, T. N. Sainath, and R. 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.
- [14] O. Hrinchuk, M. Popova, and B. 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.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.
- [16] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
- [17] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.
- [18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [19] T. Q. Nguyen and J. Salazar, "Transformers without tears: Improving the normalization of self-attention," *arXiv preprint arXiv:1910.05895*, 2019.
- [20] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.
- [21] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 559–584.
- [22] 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*, vol. 29, no. 6, pp. 82–97, 2012.
- [23] S. Majumdar, J. Balam, O. Hrinchuk, V. Lavrukhin, V. Noroozi, and B. Ginsburg, "CitriNet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition," *arXiv preprint arXiv:2104.01721*, 2021.
- [24] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.
- [25] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Krizan, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.
- [26] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE transactions on acoustics, speech, and signal processing*, vol. 35, no. 3, pp. 400–401, 1987.
- [27] I. Levin, S. Shatil-Carmon, and O. Asif-Rave, "Learning of letter names and sounds and their contribution to word recognition," *Journal of Experimental Child Psychology*, vol. 93, no. 2, pp. 139–165, 2006.
- [28] G. Stevens, "The social and demographic context of language use in the united states," *American Sociological Review*, pp. 171–185, 1992.