E2E Spoken Entity Extraction for Virtual Agents

Karan Singla *
Whissle
karan@whissle.ai

Yeon-Jun Kim
Interactions-AI
ykim@interactions.com

Srinivas Bangalore
Interactions-AI
sbangalore@interactions.com

Abstract

In human-computer conversations, extracting entities such as names, street addresses and email addresses from speech is a challenging task. In this paper, we study the impact of fine-tuning pre-trained speech encoders on extracting spoken entities in human-readable form directly from speech without the need for text transcription. We illustrate that such a direct approach optimizes the encoder to transcribe only the entity relevant portions of speech ignoring the superfluous portions such as carrier phrases, or spell name entities. In the context of dialog from an enterprise virtual agent, we demonstrate that the 1-step approach outperforms the typical 2-step approach which first generates lexical transcriptions followed by text-based entity extraction for identifying spoken entities.

1 Introduction

Enterprise Virtual Agents (EVA) provide automated customer care services that rely on spoken language understanding (SLU) in a dialog context to extract a diverse range of intents and entities that are specific to that business (Price et al., 2020). Gathering various entities like names, email, street address from human callers become a part of large range of virtual agents. In order to minimize the error in recognition and extraction of names, designers of speech interfaces often design prompts that request the user not only to say their name but spell it as well to address issues of homophones. (eg. *Catherine* or *Katheryn*). Such a behavior to spell carries over to other entities such as street and email addresses, without the users being explicitly prompted to do so.

Extensive research has been done to recognize entities in spoken input (Favre et al., 2005; Béchet et al., 2004; Sudoh et al., 2006; Gupta et al., 2005; Kim and Woodland, 2000). Similar to text-based NER, approaches for Spoken NER often involve predicting entity offsets and type in text provided by an automatic speech recognizer (ASR) (Ghannay et al., 2018; Palmer and Ostendorf, 2001; Kubala et al., 1998) or recognizing directly as a part of E2E ASR output. Significantly limited research has been done on spoken entity extraction in dialogs (Kurata et al., 2012; Kaplan, 2020), and even fewer in enterprise virtual agents (Béchet et al., 2004; Gupta et al., 2005). Some methods proposed include using a predefined list of entity names (Price et al., 2021) in a speech recognizer, fuzzy refinement by exploiting knowledge graphs (Das et al., 2022), or to using a large vocabulary speech recognizer to obtain the transcript further processed using text-based NER tools. Such techniques are difficult to adapt to caller responses to the prompt "say and spell your first/last name" in a spoken dialog system, as illustrated by the following example.

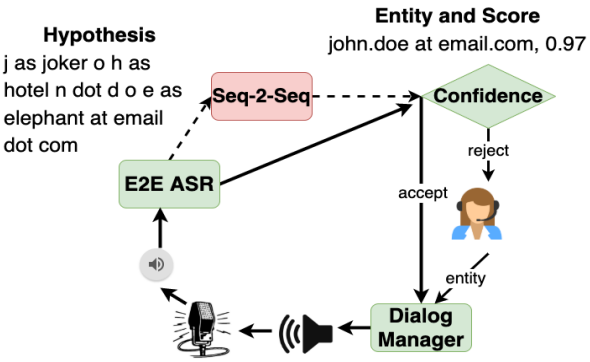

Figure 1: Overview of our proposed EVA system with human-in-the-loop for entity extraction.

```
s as in sam k as in kipe i as in
   ina ia b as in boy o as in
   over --> skibo
```

Until recently, where (Singla et al., 2022) adapt a standard Seq-2-Seq architecture to extract person names from automatically transcribed text. However, this *2-step* approach means two sys-

*Work done while working at Interactions-AI

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 567–574
December 6-10, 2023

tems to maintain and adapt, but also, possibly loss of acoustic-prosodic information. In this paper, we propose a novel method for extracting human-readable spoken entities *directly* from speech with a single model (*1-step* approach) that is optimized for the entity extraction task. We hypothesize CTC loss (Graves et al., 2006), widely used for training E2E ASR systems in an non-autoregressive manner, can be re-imagined to map audio events to text events. By generating only entity relevant tokens, our system learns to perform more intelligent entity extraction, instead of just performing literal lexical transcription as done by all existing ASR systems. We believe this opens the door to more interesting use-cases where E2E ASR systems learns to perform task-specific generation directly from speech.

We acquire data from a production EVA system which has human-in-the-loop for automation and data-collection purposes. Section 3 describes dataset in detail. We found that our proposed 1-step approach significantly outperforms the 2-step approach for extracting names, address and emails from users of an EVA system. The contributions of our paper are as follows:

- We adapt standard E2E ASR architectures optimized using CTC loss and inferred using greedy decoding to transcribe only entity relevant tokens directly from speech.
- We show that our proposed method performs better when compared to 2-step cascading approach and also better than human annotators in a fully automated human-in-the-loop dialog system.

2 Related Work

A common practice is to convert normalized token sequence in spoken form produced by ASR into a *written form* better suited to processing by downstream components in dialog systems (Pusateri et al., 2017). This written form is then used to extract *structured information* in the form of intent and slot-values to continue a dialog (Radfar et al., 2020). Recently, there is a growing tread to use neural encoders optimizing directly using speech input, popularly known as *E2E SLU* approaches (Serdyuk et al., 2018; Haghani et al., 2018).

Inverse Text normalization: Information extraction systems generally use an Inverse text normalization (ITN) component to convert a token sequence in spoken form produced by ASR into a written form suitable for downstream components – NLU and dialog. Transforming spoken language to written form involves altering entities like cardinals, ordinals, dates, times, and addresses (Sak et al., 2016; Pusateri et al., 2017). Methods proposed for ITN include: using language models (LM) to decode written-form hypothesis (Sak et al., 2016), a finite-state verbalization model (Sak et al., 2013), leveraging rules and handcrafted grammars to cast ITN as a labeling problem (Pusateri et al., 2017).

E2E SLU: Several E2E approaches which directly act on speech, have been proposed for named entity recognition, a closely related task to the entity extraction studied in this work (Ghannay et al., 2018; Tomashenko et al., 2019; Caubrière et al., 2020; Yadav et al., 2020; Shon et al., 2022). Ghannay et al.(Ghannay et al., 2018) fine-tune an E2E ASR pre-trained with the CTC loss (Amodei et al., 2016) with a set of special character labels enclosing the named entities in the transcription using CTC (Amodei et al., 2016; Ghannay et al., 2018; Caubrière et al., 2020; Tomashenko et al., 2019; Yadav et al., 2020; Shon et al., 2022).

Unlike these previous works that typically need both the text transcript along with entity type and offset tags, our approaches only need the normalized entity for supervision. Our system transparently only use pairs of audio and the target normalized entities to extract. Thus, removing a significant amount of cost and effort needed to obtain transcription and entity tags.

3 Method

We rethink ASR not only to be a transcription system but an E2E speech based encoder that can extract human-readable entities thus, learning to ignore, normalize and generate only target entity tokens directly from speech.

3.1 Non-Autoregressive Speech Based Extraction

We re-purpose an E2E ASR fine-tuned for standard transcription task and fine-tune it using *(Speech-input, Entity)* pairs using CTC loss (Graves et al., 2006). It does this by summing over the probability of possible alignments of input steps to target entity relevant tokens, producing a loss value which is differentiable with respect to each input node. We use NeMo library (Kuchaiev et al., 2019) for all training and testing purposes.

In this work, we pick an off-the-shelf Citrinet (Majumdar et al., 2021) model downloaded from NeMo library [*]. It is trained on a 7k hour collection of publicly available transcribed data and uses SentencePiece (Kudo and Richardson, 2018) tokenizer with vocabulary size of 1024 (L)[*]. We fine-tune it again for the transcription task using additional 800 hrs of transcribed speech from a collection of enterprise virtual agent applications. This model achieves a word accuracy of 93.1% on a 28k utterance test set consisting of user utterances that are in response to "How may I help you?" opening prompt from various enterprise virtual agent applications (Singla et al., 2022).

E2E Citrinet ASR is then re-purposed and fine-tuned for entity extraction. For entities (email and postal addresses) which contain vocab tokens not part of ASR tokenizer (digits, special symbols) we initiate the classification head using a sentence piece tokenizer learnt on fine-tuning data. Vocabulary size is kept as 1024 in all experiments. We fine-tune this E2E encoder for direct entity extraction from speech using a standard CTC loss.

3.1.1 From Network output to Entities

We use the same mathematical formulation as CTC (Graves et al., 2006) to classify unseen speech input sequences to minimise task specific error measure. Similar to standard practice, our CTC network has a softmax layer with one more unit than there are labels in L. The activation of the extra unit is the probability of observing a 'blank' or no label. The activation of the first L units are interpreted as the probabilities of observing the corresponding labels at particular times. But contrary to standard interpretation of CTC, our system output is contextualized over larger time-steps to output only entity relevant tokens.

More formally, for an input sequence x of length T (steps in a sample) define a speech DNN encoder with m inputs, n outputs and weight vector w as a continous map $N_w : (R^m)^T \mapsto (R^n)^T$. Let $y = N_w(x)$ be the sequence of network output, and denote by y_k^t the activation of output unit k at time t. y_k^t is interpreted as the probability of observing label k at time t, thus, defining a distribution over the set $L^{'T}$ of length T sequences over the alphabet $L^{'} = L \cup \{blank\}$:

$$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t \tag{1}$$

we refer to the elements of the $L^{'T}$ as paths, and denote them π

(Graves et al., 2006) makes an implicit assumption in Equation 1 that outputs at different times are conditionally independent. However, feedback loops within encoders connecting different position information makes them conditionally dependent. One possibly important reason behind the success of CTC based E2E ASRs using convolutional or transformer blocks.

Many−to−one map B is defined as $L^{'} \mapsto L^{\leq T}$, where $L^{\leq T}$ refers to set of sequences of length less than or equal to T over the original label alphabet L. All blanks and repeated labels are removed from the paths. Thus when optimized to output only entity relevant tokens, system outputs blanks for those time-steps instead of mapping them to a token in L (step-wise CTC outputs in Figure 2). Finally, B is used to define the conditional probability of an entity $l \in L^{\leq T}$ as sum of probabilities of all the paths corresponding to it:

$$\sum_{\pi \in B^{-1}(e)} p(\pi|x) \tag{2}$$

Figure 2 shows sample with output tokens at each time step (80ms for Citrinet) along with probability of that token. It shows system learns to ignore parts of speech to focus on spell, ignore phrases and also interpret *j as in jeery -> j*. Thus, CTC loss helps to output contextualized tokens and also align them to steps in audio without any supervision. Our experiments suggest that other E2E ASR architectures, like Conformer (Gulati et al., 2020) show similar results, when fine-tuned with non-autoregressive CTC loss.

At training time, classifier construction is done according to (Graves et al., 2006) and implemented in NeMo Library. We refer the reader to original Citrinet paper for more implementation details (Majumdar et al., 2021). For decoding, we use simple greedy CTC decoding (best path method) where the $argmax$ function is applied to the output predictions and the most probable tokens are concatenated to form a preliminary output. CTC decoding rules i.e remove blank symbols and repeated tokens, are applied to obtain readable entity.

[*]https://tinyurl.com/ykzmwhre
[*]https://tinyurl.com/y3n9drj2

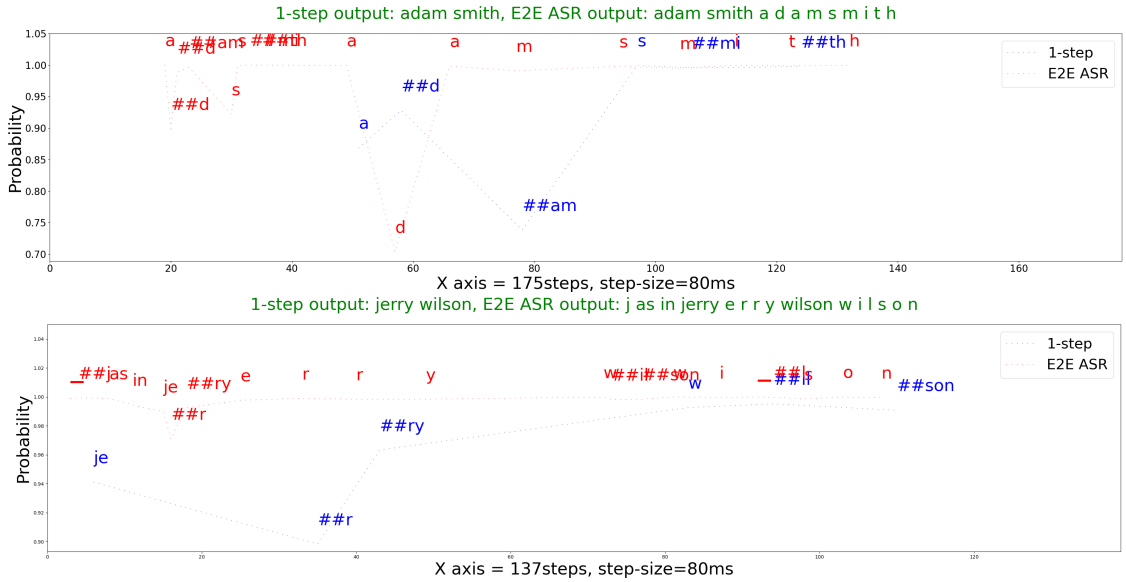

Figure 2: Samples showing output of greedy decode comparing an E2E Citrinet ASR with our proposed 1-step approach. Time steps which are not marked with any token are predicted as *blanks*. Blue tokens which mark our 1-step system's output which learns to ignore and interpret non-relevant tokens.

3.2 Baseline: Cascading ASR and NLU systems

In this 2-step approach, we first transcribe the speech provided by humans into text using same pretrained E2E ASR checkpoint used by (Singla et al., 2022). We then extract entities from the transcribed text by learning to translate using *(transcription, entity)* pairs. We use a standard off-the-shelf transformer based Seq-2-Seq system to extract entities from transcribed text baseline[*]. We found using 4 multi-headed instead of 2 performs better for all entities. Table 1 shows sample input H and desired output M.

The ASR hypothesis is provided in the form of byte-pair encoded (BPE) tokens as input to the Seq-2-Seq model, while the decoder generates entity relevant BPE tokens. We use a shared embedding layer for both encoder and decoder tokens. We use fastBPE[*] to learn a shared vocabulary for both the encoder and decoder. We use Adam optimizer with a fixed batch size of 32 and a fixed learning rate of $1.0e-5$. We do not perform any pre-training of text-based seq-2-seq model but instead train our system from a random initialization. We provide confidence-score as the sum of log-probability assigned to the BPE entity sequence.

[*]https://github.com/mead-ml/mead-baseline
[*]https://github.com/glample/fastBPE

4 Dataset

While some public datasets like the OGI collection (Cole et al., 1995) include a small subset of spelled names. EVAs for multiple industry verticals, record millions of user utterances responding to different prompts. In these dialog systems customers are prompted to provide information at various dialog turns using different authored prompts.

What's your name?
H: a l e x u s last name k i n g
M: alexus king
H: l i s a s t a n t as in tom o n
M: lisa staton
What's your street address?
H: four three eight three remo crescent road
M: 4383 remo crescent rd.
H: six forty six eighteenth street apartment one
M: 646 state st. apt 1
What's your Email-id ?
H: k as kite i n as nancy nine one five at gmail.com
M: kin915@gmail.com
H: a n as in nancy girl tower e e at outlook dot com
M: angtee@outlook.com

Table 1: Sample responses from users of an EVA system when prompted with the question.

We collect training data from several production EVA applications including banking, insurance, mobile service, and retail from callers based in the United States. Our collections are user responses in the form of audio samples and labels by human-in-the-loop agents. Human agents listen to

Keyword	FNAME	LNAME	FULLNAME	STREET	EMAIL
as	9.8	27.8	60.6	0.4	502.4
in	9.3	28.7	59.9	1.3	493.6
apple	1.8	2.0	6.9	0.2	50.6
nancy	1.1	1.7	4.3	0.0	25.9
sam	0.6	1.9	2.9	0.0	21.3
like	2.1	3.3	3.9	0.1	11.7
tom	0.5	1.4	2.4	0.0	15.2
elephant	0.3	0.5	2.1	0.0	17.1
mary	0.8	0.8	1.8	0.1	12.6
boy	0.4	1.5	1.9	0.1	11.1
dog	0.3	0.7	1.6	0.1	11.9
edward	0.5	1.1	1.7	0.0	10.5
igloo	0.0	0.2	1.4	0.0	12.0
cat	0.1	0.5	1.2	0.0	11.2
for	1.1	0.7	3.0	0.1	8.6

Table 2: Keywords (freq > 20k) and their total frequency for each entity normalized with total samples for that entitiy type in training set

customer inputs, then either type a human-readable entity or report an invalid input provided by a user. We remove the samples where user doesn't provide a meaningful inputs (keeping 70%-85% utterances). Thus creating a data with *(speech, entity)* pairs used by automatic extraction systems. Table 3 show statistics for each prompt type we use namely, first name, last name, full name, postal and email address for experiments. We keep additional valid sets which are size of 10% of train sets for model selection. Table 3 also shows median duration in the training set and also 95% percentile range for it.

Type	Dur (95% perc.)	#Train	#Test
First name (FNAME)	7.0s (3.8 - 15.6)	89k	835
Last name (LNAME)	6.5s (4.0 - 13.3)	522k	1k
Full name (FULLNAME)	10.1s (3.4 - 21.1)	241k	1k
Street address (STREET)	6.5s (2.6 - 15.4)	1.2m	4.3k
Email address (EMAIL)	12.8s (3.8 - 31.2)	620k	1k

Table 3: Statistics of training and evaluation set.

For testing purposes, we randomly sampled audio from a large pool of data, which is collected at different time frames than train data, but from same set of applications. We imitate an human-in-the-loop scenario where annotators listen to user inputs, type the entity by listening to audio only once in the limited time. Our test participants are a mix of native and non-native speakers who could be less exposed to *European* names. Table 3 shows size for test data for each type. It is often observed that the test participants introduce errors when labeling in a constraint setting like an EVA. Later, we employ native speakers of English to verify and correct entity labels. The last column in Table 5 shows human-in-the-loop performance in a constraint setting.

We merge transcriptions of training data obtained using an E2E ASR for all entities to create a list of most frequent keywords (excluding characters, number words, email-address-providers). Table 2 shows 15 most frequent keywords and their total % frequency normalized by total samples for each entity type. For example: word *as* is used in providing Email 502.4 times at average by a caller in 100 inputs. Callers take help of additional words and phrases most for providing Email address, followed by Fullname, and least for street address.

5 Experiments and Evaluation

We use only 1 NVIDIA A100 GPU for all fine-tuning purposes. We keep batch-size of 32 with starting learning rate of .001. We use weight decay of .001 and update the model with 8 accumulated batches with adam back-propagation algorithm. We will share our experiment configuration in the final revision. We report results for average of 5 runs. We provide entity confidence score by summing over the posterior probability of non-blank predicted tokens, a method originally proposed by (Kumar et al., 2020).

Results for our proposed 1-step extraction outperforms 2-step entity extraction approach, as shown at Table 5. The difference in performance is significant (permutation tests, n = 10^5, all $p < 0.05$). Results also show that our systems achieve better performance than human annotators for most prompt type except email addresses. We found extracting email addresses is hardest of all types. This is in line with our hypothesis Email-IDs are

ASR Transcription	ASR → S2S	E2E extraction
jack smith j a k s m i t h	jack smith	jak smith
fingh s i n g h	fingh	singh
lunscarard l u n d s t a a r d	lundstaard	lundsgaard
o leary o capital l apostrophe e a r y	leary	ol'eary
fourty one hundred twenty third street	4100 23rd street	41 123rd street

Table 4: Few samples cases where our proposed 1-step approach performs better than 2-step approach. Text in red highlights output is wrong, while green is correct.

	Accuracy (in %)			
Entity	2-step	1-step	1-step-joint	Human
FNAME	85.0	86.9	**89.3**	84.0
LNAME	89.0	**92.2**	92.1	89.1
FULLNAME	65.6	77.1	**82.4**	75.0
STREET	77.8	**81.9**	80.2	73.2
EMAIL	61.5	66.1	68.2	**73.6**

Table 5: Results for correctly extracting entities.

hardest to extract because of possibly infinite combinations humans can make to describe a unique ID (approximately 70% of email training data used some form of carrier phrase like "as", "in" and "like"). Joint model which pools training data for all entities shows improved performance for first-name, full-name and email extraction.

Varying amount of training data: Our proposed approach depends upon supervised data for automation. We analyze the amount the data needed before the system starts showing results which are useful to replace humans in an EVA. Figure 3 shows variation in Accuracy for full-name extraction test set. We measure accuracy at the level of words i.e: word is either first name or the last name, and at the level of characters. We found that system achieves high accuracy at the level of characters with less training data but needs more data to get complete name correct.

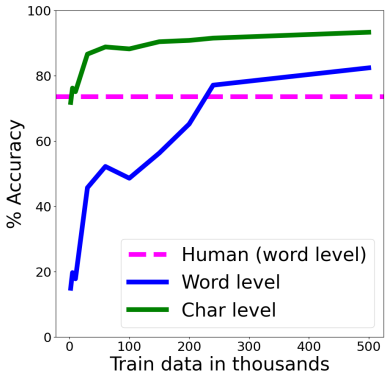

Figure 3: Varying training data and measuring accuracy for 1-step approach.

Effect of transcription quality: Results in Table 6 show that performance of cascade approach is better when human transcribed text is fed to S2S system. Performance of 2-step S2S trained on the same noisy data as E2E extraction system seems more robust as it produces high quality results if correct transcriptions are provided. However, generating transcriptions with no errors is practically impossible and also acquiring data to fine-tune an ASR for this task will be costly.

Type	2-step Transcription		1-step -
	Human	ASR	
First name	**89.0**	85.0	86.9
Last name	**92.4**	89.0	92.0
Steet address	**84.0**	77.8	81.9

Table 6: Comparing performance when human transcribed text is used instead of ASR output.

6 Observations

Linguistic analysis: We found humans break their answer into spell with or without language descriptions e.g: s as in sam more for email than other entities. Table 4 shows output for both 2-step and 1-step approach for extracting entities. We found cascading approach using S2S performs better if transcribed text provided by ASR has less errors. We believe some of these errors in transcription are due to pre-bias in language of ASR training data vocabulary. Improved performance of E2E extraction system indicates it can learn to resolve ambiguities for efficient entity extraction.

Automation Rate: Virtual agents use confidence score provided by an automatic module to decide whether call should be routed through a human agent. 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 4 shows error rejec-

tion for fullname extraction. Our 1-step approach shows 12% error rate at 20% rejection, while 2-step approach shows 25% error at 20% rejection. It also performs better than human-in-the-loop at 20% rejection rate.

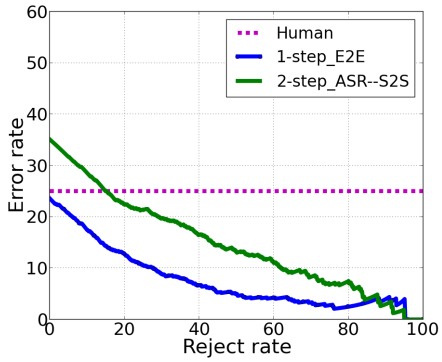

Figure 4: Error-rejection curves for full name extraction. Setting a threshold helps dialog system designer control automation rate.

7 Conclusions

In this paper we show high-quality spoken entities can be extracted directly from speech by fine-tuning E2E ASR systems. The proposed 1-step model may not be influenced by ASR mistakes while carrying the critical token sequence to the final entity extraction phase. We didn't do hyper-parameter search for the models, due to GPU limitations.

For complete automation of prompts in customer calls a system also needs to extract intent for samples (10-15%) with no entities in it. Our early experiments suggest this can be done by mixing intent labelled data (intent label used as single vocab token) or transcriptions of samples with no entities along with entity extraction data. This leads to minor loss in performance for each entity.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML 2016*, pages 173–182. PMLR.

Frédéric Béchet, Allen L Gorin, Jeremy H Wright, and Dilek Hakkani Tür. 2004. Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How may i help you? sm, tm. *Speech Communication*, 42(2):207–225.

Antoine Caubrière, Sophie Rosset, Yannick Estève, Antoine Laurent, and Emmanuel Morin. 2020. Where are we in named entity recognition from speech? In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4514–4520, Marseille, France. European Language Resources Association.

Ronald A Cole, Mike Noel, Terri Lander, and Terry Durham. 1995. New telephone speech corpora at cslu. In *Eurospeech*, pages 1–4. Citeseer.

Nilaksh Das, Duen Horng Chau, Monica Sunkara, Sravan Bodapati, Dhanush Bekal, and Katrin Kirchhoff. 2022. Listen, know and spell: Knowledge-infused subword modeling for improving asr performance of oov named entities. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7887–7891. IEEE.

Benoît Favre, Frédéric Béchet, and Pascal Nocéra. 2005. Robust named entity extraction from large spoken archives. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 491–498.

Sahar Ghannay, Antoine Caubrière, Yannick Estève, Nathalie Camelin, Edwin Simonnet, Antoine Laurent, and Emmanuel Morin. 2018. End-to-end named entity and semantic concept extraction from speech. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 692–699. IEEE.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Narendra Gupta, Gokhan Tur, Dilek Hakkani-Tur, Srinivas Bangalore, Giuseppe Riccardi, and Mazin Gilbert. 2005. The at&t spoken language understanding system. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):213–222.

Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. 2018. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. IEEE.

Micaela Kaplan. 2020. May i ask who's calling? named entity recognition on call center transcripts for privacy law compliance. *arXiv preprint arXiv:2010.15598*.

Ji-Hwan Kim and Philip C Woodland. 2000. A rule-based named entity recognition system for speech input. In *Sixth International Conference on Spoken Language Processing*.

Francis Kubala, Richard Schwartz, Rebecca Stone, and Ralph Weischedel. 1998. Named entity extraction from speech. In *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, pages 287–292. Citeseer.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Ankur Kumar, Sachin Singh, Dhananjaya Gowda, Abhinav Garg, Shatrughan Singh, and Chanwoo Kim. 2020. Utterance confidence measure for end-to-end speech recognition with applications to distributed speech recognition scenarios. In *INTERSPEECH*, volume 2020, pages 4357–4361.

Gakuto Kurata, Nobuyasu Itoh, Masafumi Nishimura, Abhinav Sethy, and Bhuvana Ramabhadran. 2012. Leveraging word confusion networks for named entity modeling and detection from conversational telephone speech. *Speech Communication*, 54(3):491–502.

Somshubra Majumdar, Jagadeesh Balam, Oleksii Hrinchuk, Vitaly Lavrukhin, Vahid Noroozi, and Boris Ginsburg. 2021. Citrinet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition. *arXiv preprint arXiv:2104.01721*.

David D Palmer and Mari Ostendorf. 2001. Improving information extraction by modeling errors in speech recognizer output. In *Proceedings of the first international conference on Human language technology research*.

Ryan Price, Mahnoosh Mehrabani, and Srinivas Bangalore. 2020. Improved end-to-end spoken utterance classification with a self-attention acoustic classifier. In *ICASSP 2020*, pages 8504–8508.

Ryan Price, Mahnoosh Mehrabani, Narendra Gupta, Yeon-Jun Kim, Shahab Jalalvand, Minhua Chen, Yanjie Zhao, and Srinivas Bangalore. 2021. A hybrid approach to scalable and robust spoken language understanding in enterprise virtual agents. In *NAACL 2021: Industry Papers*, pages 63–71.

Ernest Pusateri, Bharat Ram Ambati, Elizabeth Brooks, Ondrej Platek, Donald McAllaster, and Venki Nagesha. 2017. A mostly data-driven approach to inverse text normalization. In *INTERSPEECH*, pages 2784–2788. Stockholm.

Martin Radfar, Athanasios Mouchtaris, and Siegfried Kunzmann. 2020. End-to-end neural transformer based spoken language understanding. *arXiv preprint arXiv:2008.10984*.

Haşim Sak, Françoise Beaufays, Kaisuke Nakajima, and Cyril Allauzen. 2013. Language model verbalization for automatic speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8262–8266. IEEE.

Hasim Sak, Yun-hsuan Sung, and Cyril Georges Luc Allauzen. 2016. Written-domain language modeling with decomposition. US Patent 9,460,088.

Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE.

Suwon Shon, Ankita Pasad, Felix Wu, Pablo Brusco, Yoav Artzi, Karen Livescu, and Kyu J Han. 2022. Slue: New benchmark tasks for spoken language understanding evaluation on natural speech. In *ICASSP 2022*, pages 7927–7931. IEEE.

Karan Singla, Shahab Jalalvand, Yeon-Jun Kim, Ryan Price, Daniel Pressel, and Srinivas Bangalore. 2022. Seq-2-seq based refinement of asr output for spoken name capture. In *INTERSPEECH 2022*.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2006. Incorporating speech recognition confidence into discriminative named entity recognition of speech data. In *ACL*, pages 617–624.

Natalia Tomashenko, Antoine Caubrière, Yannick Estève, Antoine Laurent, and Emmanuel Morin. 2019. Recent advances in end-to-end spoken language understanding. In *International Conference on Statistical Language and Speech Processing*, pages 44–55. Springer.

Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. 2020. End-to-End Named Entity Recognition from English Speech. In *Proc. Interspeech 2020*, pages 4268–4272.