Improving RNN-T ASR Performance with Date-Time and Location Awareness

Swayambhu Nath Ray *, Soumyajit Mitra *, Raghavendra Bilgi *, and Sri Garimella

Alexa Speech, Amazon, India
{swayar,ssomit,rrbilgi,srigar}@amazon.com

Abstract. In this paper, we explore the benefits of incorporating context into a Recurrent Neural Network (RNN-T) based Automatic Speech Recognition (ASR) model to improve the speech recognition for virtual assistants. Specifically, we use meta information extracted from the time at which the utterance is spoken and the approximate location information to make ASR context aware. We show that these contextual information, when used individually, improves overall performance by as much as 3.48% relative to the baseline and when the contexts are combined, the model learns complementary features and the recognition improves by 4.62%. On specific domains, these contextual signals show improvements as high as 11.5%, without any significant degradation on others. We ran experiments with models trained on data of sizes 30K hours and 10K hours. We show that the scale of improvement with the 10K hours dataset is much higher than the one obtained with 30K hours dataset. Our results indicate that with limited data to train the ASR model, contextual signals can improve the performance significantly.

Keywords: End-to-End Speech Recognition, RNN-T, Contextual ASR, Contextual RNN-T

1 Introduction

Humans often use contextual information to disambiguate a particular utterance and understand incoming speech. The contextual information forms prior knowledge which can be the knowledge about a particular user or world knowledge acquired from many users. In use cases such as voice assistants, there is a lot of prior information about ASR queries. Since we train ASR on data collected from multiple users, which have been said at different contexts, some contextual information is implicitly captured and learned by the model. However, effective use of context may further improve ASR performance. For RNN-T based ASR, there is not much prior art in leveraging contextual information such as state of the device, dialog state, time at which the utterance was spoken, and state or country of origin etc.

In this paper, we focus on providing date-time and geographical information to RNN-T based ASR [1, 4, 5]. We hypothesize that date-time can be an useful signal for ASR as it carries information about type of utterances, e.g. Christmas related queries

---

* Equal contribution

will occur frequently in December. Similarly, geographical location may encapsulate user accent, and therefore benefits ASR. We demonstrate the efficacy of explicitly providing contextual information to RNN-T based ASR using up to 30K hours of de-identified queries from smart speakers.

The rest of the paper is organized as follows. We review prior work around the use of context in end-to-end (E2E) ASR in Section 2. Our context representation techniques and details of the models are outlined in Section 3. Section 4 contains the experimental details. Results and discussions are presented in Section 5. Finally, Section 6 concludes the paper.

2 Prior Work

In the literature, contextual information has been successfully used in the language modelling. In [11], location and spoken queries are used for on-the-fly adaptation of the n-gram language model. In neural models, context is often supplied either via embeddings or one-hot vectors. In [8], RNN language model is adapted based on input contextual information. Where as in [6], context embeddings are used to control a low-rank transformation of the recurrent layer weight matrix. For document classification task, temporal information has been shown to be useful [13]. Explicitly extracting contextual information also improved the results [10]. In Knowledge graphs, time information is used to learn relation between entities [3]. For RNN-T ASR, using contextual embeddings as input has been recently explored [14]. This work incorporated contextual meta data such as music playing state, and dialog state information to RNN-T encoder. Since dialog state information is available only at the end of first turn (or utterance), it can be applied to improve recognition of subsequent turns (or utterances). Where as, in our work presented here, we explore using context that is applicable to all utterances.
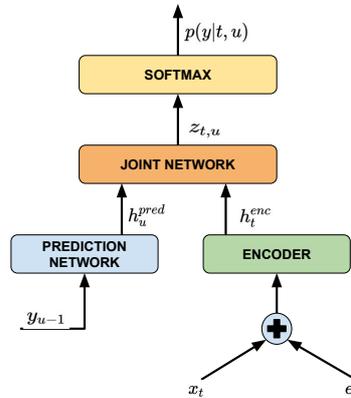
Fig. 1. *Incorporating context embeddings into RNN-T ASR. Audio features at each frame are concatenated with e_t which is either per-frame context embeddings or one-hot vector*

3   Context Representation with RNN-T

In order to use contextual information such as date-time and geo-location in RNN-T ASR, it first needs to be transformed from textual representation to continuous representation. RNN-T model consists of encoder network h^{enc}, prediction network h_u^{pred} and joint network $z_{t,u}$. A typical RNN-T network follows the following operations:

$$h_t^{enc} = f^{enc}(x_1, x_2, \cdots, x_t) \tag{1}$$

$$h_u^{pred} = f^{pred}(y_1^{pred}, \cdots, y_{u-1}^{pred}) \tag{2}$$

$$z_{t,u} = f^{join}(h_t^{enc}, h_u^{pred}) \tag{3}$$

$$P(y_u | x_1, \cdots, x_t, y_1, \cdots, y_{u-1}) = softmax(z_{t,u}) \tag{4}$$

where $x_1 \cdots x_t$ are the audio feature inputs to the RNN-T encoder and $y_1 \cdots y_u$ are the corresponding label sequence.

We extend this by adding contextual information to the encoder network by concatenating feature vector x with context vector e. Context vector e can be derived or presented to the network in multiple ways, such as:

1. one-hot representation of the context (o)
2. transforming context (c) using a contextual embedding matrix W
3. feature engineered constant sized vectors (f).

The advantage of representing context as embedding is that it provides the flexibility of combining multiple contextual signals in a lower dimensional space. In our experiments, we use 64 dimensional contextual embeddings.

$$h_{ctx_{one-hot}}^{enc} = f^{enc}(x; o) \tag{5}$$

$$h_{ctx_{embed}}^{enc} = f^{enc}(x; Wc) \tag{6}$$

$$h_{ctx_{feature-engg}}^{enc} = f^{enc}(x; f) \tag{7}$$

3.1   Date Time as Context

A typical date-time information in our dataset looks like - $2020 - 01 - 01T13 : 21$. We extract the following information from this datum:

Hour - 13, Weekday - Wednesday, Week No. - 1, Month - 1

In order to bias RNN-T recognition with temporal information, we consider two methods to convert the above information into a continuous vector representation.

Embedding Representation In this method, we learn embedding matrices for hour (24), weekday (7), week number (53) and month (12), where the numbers in the bracket indicate the maximum number of embedding vectors we use for representing corresponding information. These contexts are passed through an embedding layer to generate contextual vectors which are then averaged to represent the complete date-time information. This averaged embedding is used as biasing signal within RNN-T, and is learnt along with RNN-T model training. Assuming embedding vectors of hour, weekday, week number, and month to be h_t, wd_t, wn_t and m_t respectively, then

$$e_t = (h_t + wd_t + wn_t + m_t)/4 \qquad (8)$$

In the rest of the paper, this embedding method is addressed as TimeEmbeddingLookUp

Positional Encoding The above embedding approach (Sec 3.1) does not explicitly encode the temporal proximity and cyclical nature of time information. To capture this, we represent the date-time information using an 8-dimensional feature-engineered vector with the following entries:

$$\begin{bmatrix} sin(\frac{2\pi.hour}{24}), & cos(\frac{2\pi.hour}{24}) \\ sin(\frac{2\pi.weekday}{7}), & cos(\frac{2\pi.weekday}{7}) \\ sin(\frac{2\pi.weeknum}{53}), & cos(\frac{2\pi.weeknum}{53}) \\ sin(\frac{2\pi.month}{12}), & cos(\frac{2\pi.month}{12}) \end{bmatrix}$$

The above representation can clearly express the repetitive behaviour of temporal information. In the following sections, this embedding method is referred to as TimePositionalEncoding.

3.2   Location as Context

In this work, we used location information up to the state level in the US. The state information for utterances are collected from de-identified user specified information. Given that accent typically varies across the US states, location information is a strong signal to adapt the model to learn these variations. Instead of using all available location information, we clustered utterances with location information to form 20 clusters. The number of clusters are decided empirically with the objective of avoiding multiple centres getting mapped to the same state. Approximate geo-location information available from latlong[1] is used to obtain the state-level geo-location, and euclidean distance is used as the distance metric to learn the cluster centroids. With this we got 20 cluster centroids which are closer in distance. The clusters also include locations outside the US, which correspond to small percentage of users using the devices outside the main region. For some utterances, the location information is not available and we assign it to None cluster. We explored transforming geo-location using an embedding layer (GeoEmbeddingLookUp), and also encoding it as a one-hot vector (GeoOneHot) to bias the RNN-T model.

[1] https://www.latlong.net/category/states-236-14.html

3.3   Combination of Context

We also ran experiments combining date-time and location information to bias the RNN-T search. We expect the date-time and location together to be a much stronger signal than either individual signals alone. We use embedding approach to combine the context (CombinedTimeGeo), where the embedding matrix is learned to map combined context into lower dimension contextual embedding vector.

4   Data and Experimental Setup

4.1   Datasets

For our experiments, we used de-identified human-labelled speech data collected from queries to voice controlled far-field devices. The dataset was randomly split into train, dev and eval. The training set comprised of 30K hours of de-identified human-labelled US English recordings. Each recording includes meta information such as time stamp and optional US state from which it originated. The eval set consists of approximately 100 hours of generic utterances. We also evaluate our models on a communication specific test set of 23 hours of utterances. Both the evaluation test sets are mutually exclusive. We refer to the former as Eval test set and the latter as Comms test set.

4.2   Experimental Setup

Full Resource RNN-T ASR The baseline RNN-T model consists of 5 encoder layers of 1024 hidden units, with a final layer output dimension of 512. The prediction network has an embedding layer of 512 units, 2 LSTM layers of 1024 units, and a final output dimension 512. The joint network is a feed forward network of 512 hidden units and a final output dimension of 4001. The 4001 dimensional output, corresponds to the number of subword tokens, is passed through a final softmax layer. The subword vocabulary was generated using the byte pair encoding algorithm [12]

The contextual RNN-T ASR model has an additional embedding layer generating embedded representation of 64 dimensions which are appended to the input of the encoder at every time step. The two exceptions being:

1. the positional time encoding has an 8 dimensional context vector
2. one-hot geographical information has a 21 dimensional context vector

All the models are trained on 30K hours of training data.

Low Resource RNN-T ASR We also trained both the baseline and contextual models on 10K hours of data. The main motivation for this study is to analyze the effect of context in the low training data regime. In order to prevent over-fitting, we scaled down the number of parameters of the models. Number of hidden units of both the encoder and decoder layers were reduced to 760. The feed-forward joint network is also removed. The encoder and decoder outputs are summed and provided as an input to the softmax layer. All other specifications are kept consistent with the full-resource models.

Both full-resource and low-resource models use a 64-dimensional log filter bank energy features computed over 25ms window with 10ms shift. Each feature vector is stacked with 2 frames to the left and down sampled to a 30ms frame rate. We also augment the acoustic training data with SpecAugment [9] to improve the robustness. All models are trained using the Adam optimizer [7], with a learning rate schedule including an initial linear warm-up phase, a constant phase, and an exponential decay phase [2]. These hyper-parameters are not specifically tuned for this work.

5   Results and Discussion

5.1   Overall WER Comparison

Table 1 shows the overall Relative Word Error Rate Reduction (WERR) with respect to the baseline RNN-T model without context. Performance of our baseline system is below 10% WER absolute. The magnitude of improvement on Comms test set is more than that of Eval test set, which signifies that these contextual signals are more favourable for communication specific utterances. Overall, incorporating geo-location information as one-hot provides the maximum WERR of 3.48%. Based on the performance, we chose TimeEmbeddingLookUp and GeoOneHot models for further analyses.

Table 1. *Relative WERRs of full resource contextual models w.r.t baseline*

Model	Eval	Comms	#params
Baseline	—	—	58.4M
TimeEmbeddingLookUp	**1.73%**	**2.68%**	58.7M
TimePositionalEncoding	1.33%	2.39%	58.4M
GeoEmbeddingLookUp	1.47%	1.6%	58.6M
GeoOneHot	**2.27%**	**3.48%**	58.5M

5.2   Domain-wise WER Comparison

We show the WER improvement of Eval set for various domains in Table 2. In general, we see gains on all top domains of interest with both approaches. Geo-location exhibits superior performance in domains like Music and CallingAndMessaging etc. These domains capture region specific preferences for music and video along with accent variations of proper nouns. On the other hand temporal context shows improvement on domains where queries come mostly at a certain point of time in a day, e.g. DailyBriefing, Weather etc.

Table 2. *Per-domain relative WERR (%) breakdown on top 10 frequent domains selected from the test set. Analysis was done using the full resource model.*

Domain	TimeEmbeddingLookUp	GeoOneHot
Music	1.72	**3.57**
Shopping	2.03	1.49
CallingAndMessaging	2.04	**5.68**
Global	0.86	2.58
DailyBriefing	**4.55**	0.91
Knowledge	2.14	1.97
Video	3.13	**6.26**
Weather	**5.92**	4.67
Information	1.44	1.96
ScienceAndTechnology	-0.49	-2.17

Table 3. *Relative WERRs of combined contextual model w.r.t baseline. Both the models are trained on complete 30K hours of data*

Model	Eval	Comms
Baseline	—	—
CombinedTimeGeo	**3.6%**	**4.62%**

5.3   Combined Context

The effect of combining the two contextual signals is captured in Table 3. Combining the contextual information shows superior performance compared to individual contextual models (Table 1). This establishes the additive effect of the location and date-time signals. In domain-wise study, we see a similar additive effect on several domains like CallingAndMessaging (6.05%), Knowledge (4.44%), Video (7.86%) and Weather (11.56%) etc. Moreover, in domains like ScienceAndTechnology, where neither of the individual contextual models showed any improvement, the combined model performed significantly better (6.61% WERR).

5.4   Low Resource Simulation

In practice we often face data scarcity while developing ASR for a new language or locale. In such cases, we can easily leverage contextual information as they are readily available to gain additional performance benefits. We simulated this situation by randomly selecting a 10K hours subset from the full training data, and trained both the baseline and the contextual models on this subset. The model sizes have also been reduced to avoid over-fitting as described in Section 4.2. Table 4 shows that the magnitude of gains have increased as compared to full resource, which demonstrates the efficacy of these contextual signals for low resource scenarios.

Table 4. *Relative WERRs of Low Resource contextual models wrt baseline*

Model	Eval test set	Comms test set	#params
Baseline	—	—	38M
TimeEmbeddingLookUp	4.03%	3.66%	38.2M
GeoOneHot	4.29%	3.76%	38.1M

Table 5. *WERR (%) for top 3 and bottom 3 performing month and geographical state/country*

Model	Top 3 (WERR)	Bottom 3 (WERR)
TimeEmbeddingLookUp	December (11.61) February (10.49) January (5.24)	November (0.25) August (-0.64) September (-4.86)
GeoOneHot	Germany (9.48) Hawaii (5.38) Washington (4.77)	Ohio (1.52) Florida (1.02) California (0.79)

5.5   Month and State-wise WER Comparison

To further understand the effect of our proposed methods, we performed a month-wise and state-wise WERR analyses for date-time and geo-location based models respectively. We have captured the best and worst performing month/state for date-time/geo-location models in Table 5. The date-time information enhances the performance of RNN-T for some winter months considerably. On the other hand, geo-location signal significantly enhance the performance on utterances coming from low resource regions like Germany, Hawaii and its nearby locations. This can be mostly attributed to difference in acoustics of utterances coming from these regions as they are different from that of other regions captured by the geo-location clusters. Even for the worst performing region, we do not see any degradation of performance with geo-location as context.

Table 6. *Comparison of contextual model output and baseline output*

Reference	Baseline output	Contextual output	Context used
good night and **happy hanukkah**	good night and happy hobbiter	good night and **happy hanukkah**	Time
what's the **christmas cat story**	what's the christmas car story	what's the **christmas cat story**	Time
call **guillermo**	call galermo	call **guillermo**	Geo-location
turn to my **kirk franklin** radio	turn to my park franklin radio	turn to my **kirk franklin** radio	Geo-location

5.6 Baseline and Contextual Model Outputs

In Table 6 we compare a few example predictions from baseline and contextual model. We see that, the time information helps in recognition of phrases like "hanukkah" and "christmas cat", which the baseline model fails to recognize. These phrases are seen in December which is implicitly captured by the contextual model.

Similar to time, when we use location information as context, it captures the accent variations and local preferences of music and videos. The contextual model was able to capture the local accent variation and correctly output "guillermo" while the baseline model outputs "galermo" which is somewhat phonetically similar to the correct phrase. Similarly "kirk franklin" was correctly recognized compared to incorrect baseline output of "park franklin" which shows that the model was able to capture local variation in music preferences without any external supervision.
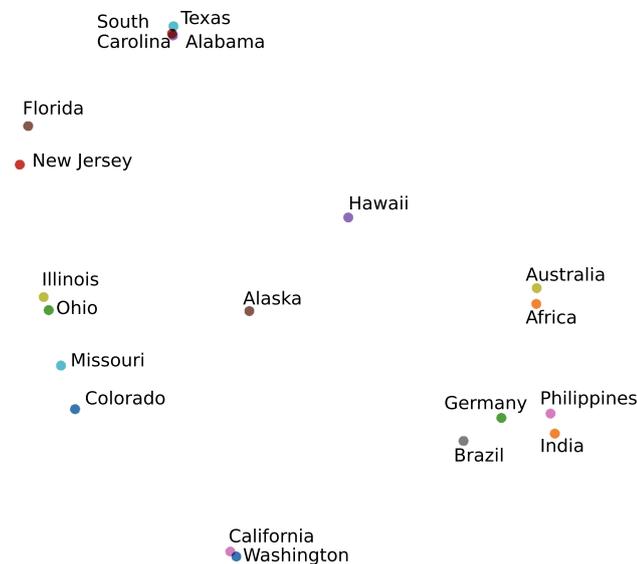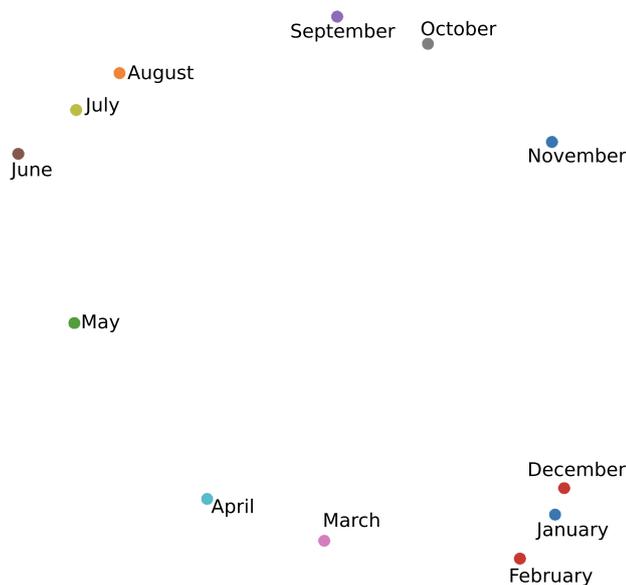

Fig. 2. *t-SNE plot for geo-embeddings*

5.7 t-SNE Plot Analysis

Embeddings are meant to capture some implicit information about the context it represents. To understand the significance of the embedding vectors learnt by the models, we projected the 64 dimensional geo-location and month embedding vectors from GeoEmbeddingLookUp and TimeEmbeddingLookUp on 2-D space using t-SNE with default parameters of Embedding Projector[2].

---

[2] https://projector.tensorflow.org/

Fig. 3. *t-SNE plot for month embeddings*

In Figure 2 we show the t-SNE plot for the learnt geo-embedding vectors. We can see that the geographically close states in the US have formed clusters in the embedding space (e.g - California:Washington, Illinois:Ohio etc.) which seems to capture local variations like regional movies and song preference and also local accents. We can also see a clear demarcation between the US locations and the non-US locations like Brazil, Germany and India. Users across the US and the non-US will have different accents which are captured by the model. This proves that the geographical location distribution is important for the model and the model has learnt that without any external supervision.

Figure 3 shows the t-SNE plot for embedding vectors corresponding to month. We can observe a clear temporal ordering among the learnt month vectors which demonstrate the capacity of our models to implicitly learn the ordering among months from data. This phenomenon also proves that the temporal ordering of months is somewhat important for the task of ASR. Note that, we have not imposed any ordering constraint on any of our models.

6   Conclusions

In this paper, we explored the benefits of using contextual signals to improve the overall performance of end-to-end ASR based on RNN-T. We demonstrated the effectiveness of date-time and location as context by building ASR models on 30K and 10K hours of data. We provided empirical evidence that biasing ASR using contextual signals improves the overall accuracy. The use of individual contextual signals improved the ASR

WER up to 3.48% relative, and where as their combination resulted in about 4.62% relative gain. Our analysis with t-SNE plot of embedding vectors for both geo-location and date-time context showed that the model was able to extract meaningful information from these signals and improving ASR, thereby possibly reducing the need for additional training data which is now critical for performance improvement. As a part of future work, we would like to add dynamic contextual signals along with these static ones to further enhance RNN-T ASR performance.

References

1. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4960–4964. IEEE (2016)
2. Chen, M.X., Firat, O., Bapna, A., Johnson, M., Macherey, W., Foster, G., Jones, L., Parmar, N., Schuster, M., Chen, Z., et al.: The best of both worlds: Combining recent advances in neural machine translation. arXiv preprint arXiv:1804.09849 (2018)
3. Dasgupta, S.S., Ray, S.N., Talukdar, P.: HyTE: Hyperplane-based temporally aware knowledge graph embedding. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 2001–2011. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1225, https://www.aclweb.org/anthology/D18-1225
4. Graves, A.: Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711 (2012)
5. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376 (2006)
6. Jaech, A., Ostendorf, M.: Low-rank RNN adaptation for context-aware language modeling. CoRR **abs/1710.02603** (2017), http://arxiv.org/abs/1710.02603
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. In: 2012 IEEE Spoken Language Technology Workshop (SLT). pp. 234–239 (2012). https://doi.org/10.1109/SLT.2012.6424228
9. Park, D.S., Chan, W., Zhang, Y., Chiu, C.C., Zoph, B., Cubuk, E.D., Le, Q.V.: Specaugment: A simple data augmentation method for automatic speech recognition. arXiv preprint arXiv:1904.08779 (2019)
10. Ray, S.N., Dasgupta, S.S., Talukdar, P.: Ad3: Attentive deep document dater (2019)
11. Scheiner, J., Williams, I., Aleksic, P.: Voice search language model adaptation using contextual information. In: 2016 IEEE Spoken Language Technology Workshop (SLT). pp. 253–257 (2016). https://doi.org/10.1109/SLT.2016.7846273
12. Shibata, Y., Kida, T., Fukamachi, S., Takeda, M., Shinohara, A., Shinohara, T., Arikawa, S.: Byte pair encoding: A text compression scheme that accelerates pattern matching. Tech. rep., Technical Report DOI-TR-161, Department of Informatics, Kyushu University (1999)
13. Vashishth, S., Dasgupta, S.S., Ray, S.N., Talukdar, P.: Dating documents using graph convolution networks (2019)
14. Wu, Z., Li, B., Zhang, Y., Aleksic, P.S., Sainath, T.N.: Multistate encoding with end-to-end speech rnn transducer network. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7819–7823. IEEE (2020)