Towards Continual Entity Learning in Language Models for Conversational Agents

Anonymous Author(s)
Affiliation
Address
email

Abstract

Neural language models (LM) trained on diverse corpora are known to work well on previously seen entities, however, updating these models with dynamically changing entities such as place names, song titles and shopping items requires re-training from scratch and collecting full sentences containing these entities. We aim to address this issue, by introducing entity-aware language models (EALM), where we integrate entity models trained on catalogues of entities into the pre-trained LMs. Our combined language model adaptively adds information from the entity models into the pre-trained LM depending on the sentence context. Our entity models can be updated independently of the pre-trained LM, enabling us to influence the distribution of entities output by the final LM, without any further training of the pre-trained LM. We show significant perplexity improvements on task-oriented dialogue datasets, especially on long-tailed utterances, with an ability to continually adapt to new entities (to an extent).

1 Introduction

Neural Language Models have become the de facto standard for modeling natural language; however, they still lack some of the flexibilities offered by their n-gram counterparts [3] in being able to continually update to new entities at a minimal cost [30] or quickly adapt to personalized content [1]. This issue is more prominent in task-oriented dialogue systems where utterances are typically rich in entities such as place names, shopping items, song titles, person names etc. To address some of these issues, we introduce entity-aware language models (EALM), where we attempt to decouple entity models from the pre-trained LMs. We achieve this by linearly interpolating the final layer output representations of the pre-trained LM and independently trained entity models. The interpolation probabilities at each timestep are determined by a contextual fusion layer based on the sentence context. This approach allows us to better predict entities present in text, using the information provided by the entity models along with the distribution learned by the pre-trained LM. Our final language model consists of three components (1) A pre-trained language model, (2) Entity models, and (3) A Contextual fusion layer to combine the pre-trained LM with entity models.

Our entity models are neural LMs trained on catalogues containing entities. They share the input and output layers with the pre-trained LM and these parameters are frozen throughout the training, essentially mapping entities to the pre-trained LM's embedding space. This modular approach allows us to update our entity models with new information, independently of the pre-trained LM. The improved entity models can be directly plugged back into the final LM without any further training. This enables our LM to continually learn new entities without requiring to update the pre-trained LM.

It is important to note that our final LM should not always rely on entity models while predicting entities in text. Entities in natural language can be classified into two categories based on the context in which they appear: (1) fact-based entities that depend on real-world facts, and (2) preference-based entities that depend on external factors such as time, location, speaker, etc. For example, in the utterances *play hello by adele* and *play any song by adele*, artist *adele* in the first utterance can be classified as a fact-based entity and is determined by the utterance context alone (plus some real-world knowledge), whereas, *adele* in the second is a preference-based entity that depends on auxiliary information outside of the utterance. Unlike [15, 28, 44], that are desgined to improve the representation of fact-based entities, in this paper, we only focus on preference-based entities.

Our main contributions are as follows: (1) We propose an approach to integrate entity models trained on catalogues of entities into a pre-trained LM, (2) We show that this improves our LM's capability in predicting long-tailed entities and continually learn new entities to an extent.

1.1 Our Approach

Given utterance context $w_{0...t-1} = (w_0, ..., w_{t-1})$, the goal of an autoregressive language model is to estimate the probability distribution $p(w_t|w_{0...t-1})$ of the next token w_t, where w_0 is typically the start token `<s>` used to predict the first token. We can factorize this probability distribution as,

$$p(w_t|w_{0...t-1}) = \sum_{i=0}^{N} p(w_t|w_{0...t-1}, E_i) \cdot p(E_i|w_{0...t-1}) \tag{1}$$

where E_0 represents the pre-trained LM and $E_1, ...E_N$ represent the entity LMs. However, entity models cannot accurately predict $p(w_t|w_{0...t-1}, E_i)$ as they are only trained on entity catalogues. As a result, we use the utterance context $w_{0...t-1}$ to determine the entity context $w_{t-l^*...t-1}$[1] containing only entity terms and approximate $p(w_t|w_{0...t-1}, E_i)$ as,

$$p(w_t|w_{0...t-1}, E_i) \approx p(w_t|[w_0; w_{t-l^*...t-1}], E_i) \quad \forall i = 1...N$$
$$l^* = \arg\max_{l \in [0,t-1]} p(l|w_{0...t-1}, E_i) \tag{2}$$

where $[w_0; w_{t-l^*...t-1}] = (w_0, w_{t-l^*}, ..., w_{t-1})$. We assume $[w_0; w_{t-0...t-1}] = w_0$ for notational simplicity. We further simplify Eq. 2 using markov assumption on the entity models by only considering k tokens preceding the current timestep, $l^* = \arg\max_{l \in [0,k]} p(l|w_{0...t-1}, E_i)$. We show the utterance contexts of our entity models at different timestamps using an example in Table 4

To convert this into a continuous optimization problem, we rewrite Eq. 2 as an expectation over all possible entity contexts and our updated equation looks like,

$$p(w_t|w_{0...t-1}, E_i) \approx \sum_{l=0}^{k} p(l|w_{0...t-1}, E_i) \cdot p(w_t|[w_0; w_{t-l...t-1}], E_i) \quad \forall i = 1...N \tag{3}$$

2 Entity-Aware Language Models

In this section, we describe our proposed Entity-Aware Language Models (EALM). Our LM has three components that are trained sequentially: (1) A pre-trained language model, (2) Entity models and, (3) A contextual fusion layer. Our LM components and their interactions are outlined in Figure 1

Given an utterance context $w_{0...t-1} = (w_0, ..., w_{t-1})$, our LM first converts these tokens into a dense representation $X = (x_0, ..., x_{t-1})$, where $X \in \mathbb{R}^{t \times d}$, using an input embedding layer. The pre-trained LM and the entity models process this input representation as described below.

---

[1]$t - l^*$ denotes the start index of the entity in the utterance, if the current phrase is an entity, $t - l^* = 0$ otherwise.
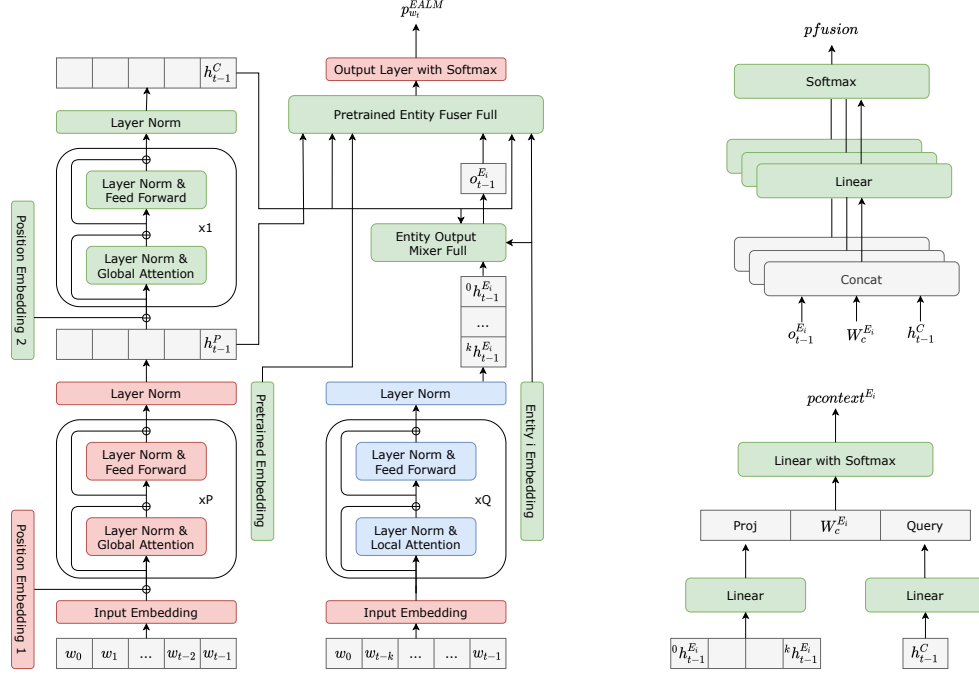
Figure 1: Figure showing the EALM architecture. Pre-trained entity fuser is shown in top right and entity output mixer is shown in the bottom right. Parameters of the pre-trained LM, entity model and contextual layer are highlighted in red, blue and green respectively. We have only shown one entity model in the figure, additional models follow the same structure.

2.1 Pre-trained Language Model

We use a multi-layer Transformer decoder similar to [39, 31] as our pre-trained LM. Given the input representation X, we first add positional information to it, $H_0^P = X + P_e$ where $P_e \in \mathbb{R}^{t \times d}$.

$$H^P = GlobalAttentionTransformer(H_0^P) \tag{4}$$

The last token representation h_{t-1}^P of H^P, where $h_{t-1}^P \in \mathbb{R}^d$ and $H^P \in \mathbb{R}^{t \times d}$, encodes $w_{0...t-1}$.

2.2 Entity Models

Unlike the pre-trained LM that takes the entire utterance context $w_{0...t-1}$ as input, entity models only process the last k tokens as described in Eq. 3. We, therefore, use local attention [26] and relative positional embeddings [35] for our entity model self-attention layers. In addition to this, our entity models generate $k + 1$ outputs at every timestep, by varying the utterance context length from 0 through k based on Eq. 3. To summarize, our entity models take $X' = (x_0, x_{t-k}, ..., x_{t-1})$ as input and generate the current timestep's output $h_{t-1}^{E_i} \in \mathbb{R}^{k+1 \times d}$ [2] by effectively masking the attention context based on the utterance context length l, where $l \in [0, k]$,

$$h_{t-1}^{E_i} = LocalAttentionTransformer(H_0^{E_i}) \tag{5}$$

---

[2]Note that $k + 1$ here does not represent the time dimension, but rather represent the $k + 1$ different outputs obtained by changing the utterance context l length from 0 to k.

2.3 Contextual Fusion Layer

The contextual fusion layer integrates information from the entity models into the pre-trained LM by linearly interpolating their final layer output representations based on the utterance context [3]. The fusion layer represents the entity models and the pre-trained LM using a class embedding matrix $W_c \in \mathbb{R}^{N+1 \times d}$, where N is the number of entity models. It also creates a new encoding $h_{t-1}^C \in \mathbb{R}^d$ of the utterance context $w_{0...t-1}$ using an additional transformer decoder on top of Eq. 4.

The goal of the contextual fusion layer is twofold: (1) Approximate the probability distribution of the next token given by our entity models by identifying their respective entity contexts (Eq. 3) and (2) Determine the probability distribution of the next token given by our final LM by estimating the interpolation weights (Eq. 1).

Entity Contexts Given $h_{t-1}^{E_i}$, $W_c^{E_i}$ and h_{t-1}^C, we first compute the probability distribution $pcontext^{E_i} \in \mathbb{R}^{k+1}$ over all possible entity contexts at the current timestep as shown in Figure 1. We then create a compact representation $o_{t-1}^{E_i} \in \mathbb{R}^d$ of the entity model's output as follows,

$$pcontext^{E_i} = EntityOutputMixer(W_c^{E_i}, h_{t-1}^{E_i}, h_{t-1}^C)$$
$$o_{t-1}^{E_i} = \sum_{l=0}^{k} {}^l pcontext^{E_i} \cdot {}^l h_{t-1}^{E_i} \quad \forall i = 1...N \tag{6}$$

where the left superscript l indicates the lth row of the matrix or vector.

Entity Fusion Given h_{t-1}^P, $o_{t-1}^{E_i}$, W_c and h_{t-1}^C we compute the interpolation probabilities $pfusion \in \mathbb{R}^{N+1}$ at every timestep as shown in Figure 1. We combine the output representations of the entity models and the pre-trained LM as follows,

$$^i pfusion = {}^i Softmax(PretrainedEntityFuser(o_{t-1}^{E_i}, W_c^{E_i}, h_{t-1}^C))$$
$$h_{t-1}^{EALM} = \sum_{i=0}^{N} {}^i pfusion \cdot o_{t-1}^{E_i} \tag{7}$$

where E_0 represents the pre-trained LM and $o_{t-1}^{E_0} = h_{t-1}^P$. We use the interpolated representation h_{t-1}^{EALM} to obtain the probability distribution of the next token using,

$$p^{EALM}(w_t|w_{0...t-1}) = Softmax(h_{t-1}^{EALM} W_o) \tag{8}$$

We freeze the parameters of the pre-trained LM, input and output embedding layers while training the entity models. We also freeze entity models during our contextual fusion layer training. All our LMs and the Fusion Layer are trained to minimize the negative log-likelihood of the utterances or entities.

3 Experiments

Our training data consists of transcribed utterances from user interactions with a voice assistant. We use entity catalogues to train entity models. Please refer A.1 for our detailed experimental setup. We evaluate our model on its ability to represent the long-tail and its effectiveness in continually learning new entities without forgetting old ones. We provide a description of the test sets used for evaluation in Table 1. All our test sets are created from a much bigger general test set unless specified. We make an assumption that entities are new if they are not present in our training data and catalogues. We evaluated all our test sets on the pre-trained model and Table 2 shows the relative percentage difference in perplexity of our tail and new test sets compared to the entity seen test set. We use the tail/new test sets as the reference (denominator) for this table.

---

[3]This is slightly different from Eq. 1 where the interpolation is done on the probability distributions.

Table 1: Description of our testsets that are de-identified, transcribed and annotated.

Name	Size	Entites are seen during training	Entities are present in our catalogues	Every utterance contains an entity
General	$48k$	some of them	some of them	no
EntitySeen	$18k$	yes (a few times)	some of them	yes
EntityTail	$8k$	only individual words	yes	yes
EntityNew	$14k$	only individual words	no	yes
EntitySpanish	$23k$	no	no	yes
EntityTailNew	$23k$	no	yes	yes

Table 2: Relative difference in perplexities of our test sets when evaluated on our pre-trained LM.

General	EntitySeen	EntityTail	EntityNew	EntitySpanish	EntityTailNew
-11.46%	0 %	47.55%	54.83%	85.8%	88.75%

Evaluation on Long-Tail We show the relative reduction in perplexity obtained by our baseline EALM compared to the pre-trained LM, on all our test sets, in the first part of Table 3. We can see that our baseline EALM consistently outperforms the standalone pre-trained model on all our test sets. The improvement is much more significant on tail entities (10.37%), that are unseen during training, but are present in our catalogues.

Continual Learning of Entities To evaluate our model on whether it can continually learn entities, we re-trained all our entity models by adding entities from the tail and new test sets (including dev sets), to the top five percent (ranked by popularity) of our catalogues. We did not retrain the contextual fusion layer. We see a big improvement from 5.63% to 14.38% on utterances containing new entities, refer Table 3. While this is encouraging, the improvement needed to match the performance of our pre-trained LM on popular entities, assuming we have full utterances containing these entities can be inferred as 54.83% from Table 2. We also evaluated our method on whether it can learn popular entities from other languages, which is not atypical in task-oriented dialogue systems. We can see that our model achieved a huge overall perplexity reduction of 39.12%. Again, this is still significantly lower compared to the 85.8% (refer Table 2) improvement needed to match the performance on existing popular entities. Refer A.2 for further discussion on continual learning.

4 Future Work

Our language models can be directly used for shallow fusion [38, 4] and n-best hypothesis rescoring [14, 36] in seq2seq based speech recognition systems. Our approach can also be extended to integrate entity models directly into the decoders of these systems. To improve the recognition of tail entities, we can adapt our entity models based on context such as user, location, season etc [21]. To scale this, we might have to explore sharing parameters across entity models. We can also use our approach for quick hotfixes in production systems. Another interesting line of work would be to use our EALM for intent classification and entity extraction tasks by making use of the interpolation probabilities Eq. 7 as additional cues indicating the presence of entities. Our LM may not have a huge impact on tasks external [40] to the dialogue systems which benefit more from fact-based entities and their relations.

5 Related Work

External Memory Adding external memory to the neural networks was first proposed in [42, 37] where they used attention mechanism [2] to access memory, based on the current context. [22, 13] extended this to LMs to increase the probability of recently seen words and later [12] scaled this to larger contexts using k-nearest neighbors to retrieve from memory. Recently, [15] proposed to learn

Table 3: Table showing the effectiveness of EALM on different testsets. All results shown are percentage reduction in perplexity compared to the pretrained LM. We removed the % sign.

Testset	All	Song	Album	Celeb	Video	Item	Place	Person
Our base EALM. Entity models are trained on full catalogues.								
General	1.56	3.77	1.88	3.32	2.81	2.72	1.91	0.99
EntitySeen	5.46	8.83	5.42	9.33	0.52	2.15	1.38	2.74
EntityTail	10.37	16.52	9.82	16.19	4.22	7.69	4.02	5.14
EntityNew	5.63	8.66	6.46	8.39	3.54	5.69	3.57	0.81
EntitySpanish	21.9	28.32	23.24	29.32	18.92	14.13	14.24	13.22
Entity models are re-trained by adding tail and new entities and plugged back into the EALM.								
General	1.73	3.71	2.46	3.38	3.32	3.98	1.57	1.48
EntitySeen	5.34	8.36	5.45	8.92	0.31	2.57	1.5	2.45
EntityTail	14.79	19.89	13.29	20.44	6.18	13.89	7.23	5.43
EntityNew	14.38	16	13.66	18.03	9.35	15.73	9.54	7.2
Entity models are re-trained by adding spanish entities and plugged back into the EALM. We did not add the tail and new entities for this experiement.								
General	1.25	2.99	1.45	2.95	2.95	2.97	1.39	0.84
EntitySpanish	39.12	40.53	37.93	48.14	24.37	41.21	28.03	14.02

new information by just populating the memory with more data using a pre-trained LM trained on a small corpus. These methods are orthogonal to our work as they can only improve the prediction of fact-based or recently seen entities and cannot be easily extended to preference-based entities, using only a few nearest neighbors for interpolation or computing attention over millions of entities [29].

Continual Learning   Continual learning approaches focus on learning new information without forgetting the previously acquired knowledge. Current continual learning approaches can be classified into three categories: (i) regularization [16, 5], (ii) memory [6, 7] and (iii) expansion [32, 33] based approaches. They are still an active area of research and require full sentences to learn new entities. It is also not straightforward on how to use these approaches for personalization and contextualization.

Entity Models   Earlier work on building separate models for entities and combining them with a general language model used annotated data and class n-gram LMs [41]. [27] extended this to neural models, by predicting the entity type along with the next word and improved the representation of tail entities. [19] followed a similar approach but instead encoded the type information in a latent variable with the help of type specific vocabularies and did not use any labeled data. However, these approaches only work for entities seen during training and cannot be extended to support personalization or continual learning. Our work resembles [9], where they interpolate component models trained on entity catalogues with a pre-trained LM. Unlike our approach, they explicitly reset the hidden states of the entity models, using a binary activation process, to indicate the start of entities in the utterance context and use reinforcement learning to learn the discrete activation policy.

6   Conclusion

We proposed a technique to integrate entity models trained on catalogues of entities into pre-trained LMs, significantly improving the perplexity on long-tailed entities. We also showed that our LM can continually learn new entities at a low cost, outperforming the pre-trained LM on these entities, without any signs of forgetting. We discussed the limitations of our continual learning approach in matching the performance of a pre-trained LM directly trained on these new entities. Nevertheless, our LMs can still be used to learn new entities or hotfix trending entities between deployment cycles and can be easily extended to support personalization and contextualization in conversational agents.

References

[1] Petar Aleksic, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach, and Pedro Moreno. Bringing contextual information to google speech recognition. In *Interspeech 2015, International Speech Communications Association*, 2015.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

[3] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480, 1992.

[4] Rodrigo Cabrera, Xiaofeng Liu, Mohammadreza Ghodsi, Zebulun Matteson, Eugene Weinstein, and Anjuli Kannan. Language model fusion for streaming end to end speech recognition. *arXiv preprint arXiv:2104.04487*, 2021.

[5] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 556–572, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01252-6.

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

[7] Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f8d2e80c1458ea2501f98a2cafadb397-Paper.pdf.

[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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[9] Denis Filimonov, Ravi Teja Gadde, and Ariya Rastrow. Neural composition: Learning to generate from multiple models. *arXiv preprint arXiv:2007.16013*, 2020.

[10] Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/2f4059ce1227f021edc5d9c6f0f17dc1-Paper.pdf.

[11] Aditya Gourav, Linda Liu, Ankur Gandhe, Yile Gu, Guitang Lan, Xiangyang Huang, Shashank Kalmane, Gautam Tiwari, Denis Filimonov, Ariya Rastrow, Andreas Stolcke, and Ivan Bulyko. Personalization strategies for end-to-end speech recognition systems. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7348–7352, 2021. doi: 10.1109/ICASSP39728.2021.9413962.

[12] Edouard Grave, Moustapha Cisse, and Armand Joulin. Unbounded cache model for online language modeling with open vocabulary. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6044–6054, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[13] Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=B184E5qee`.

[14] Ke Hu, Ruoming Pang, Tara N Sainath, and Trevor Strohman. Transformer based deliberation for two-pass speech recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 68–74. IEEE, 2021.

[15] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *CoRR*, abs/1911.00172, 2019. URL `http://arxiv.org/abs/1911.00172`.

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1611835114. URL `https://www.pnas.org/content/114/13/3521`.

[17] Anjishnu Kumar, Arpit Gupta, Julian Chan, Sam Tucker, Bjorn Hoffmeister, Markus Dreyer, Stanislav Peshterliev, Ankur Gandhe, Denis Filiminov, Ariya Rastrow, et al. Just ask: building an architecture for extensible self-service spoken language understanding. *arXiv preprint arXiv:1711.00549*, 2017.

[18] Duc Le, Mahaveer Jain, Gil Keren, Suyoun Kim, Yangyang Shi, Jay Mahadeokar, Julian Chan, Yuan Shangguan, Christian Fuegen, Ozlem Kalinli, et al. Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion. *arXiv preprint arXiv:2104.02194*, 2021.

[19] Angli Liu, Jingfei Du, and Veselin Stoyanov. Knowledge-augmented language model and its application to unsupervised named-entity recognition. 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)*, pages 1142–1150, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1117. URL `https://www.aclweb.org/anthology/N19-1117`.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[21] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Haşim Sak, Alexander Gruenstein, Françoise Beaufays, et al. Personalized speech recognition on mobile devices. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5955–5959. IEEE, 2016.

[22] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Byj72udxe`.

[23] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 1999.

[24] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[25] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6301. URL `https://www.aclweb.org/anthology/W18-6301`.

[26] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4055–4064. PMLR, 10–15 Jul 2018. URL `http://proceedings.mlr.press/v80/parmar18a.html`.

[27] Md Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. Building language models for text with named entities. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2373–2383, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1221. URL `https://www.aclweb.org/anthology/P18-1221`.

[28] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1005. URL `https://www.aclweb.org/anthology/D19-1005`.

[29] Golan Pundak, Tara N. Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. Deep context: End-to-end contextual speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 418–425, 2018. doi: 10.1109/SLT.2018.8639034.

[30] Ernest Pusateri, Christophe Van Gysel, Rami Botros, Sameer Badaskar, Mirko Hannemann, Youssef Oualil, and Ilya Oparin. Connecting and Comparing Language Model Interpolation Techniques. In *Proc. Interspeech 2019*, pages 3500–3504, 2019. doi: 10.21437/Interspeech.2019-1822. URL `http://dx.doi.org/10.21437/Interspeech.2019-1822`.

[31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[32] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[33] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4535–4544. PMLR, 2018. URL `http://proceedings.mlr.press/v80/schwarz18a.html`.

[34] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL `https://www.aclweb.org/anthology/P16-1162`.

[35] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association*

9

for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL `https://www.aclweb.org/anthology/N18-2074`.

[36] Ashish Shenoy, Sravan Bodapati, Monica Sunkara, Srikanth Ronanki, and Katrin Kirchhoff. Adapting long context nlm for asr rescoring in conversational agents. 2021.

[37] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf`.

[38] Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu. A comparison of techniques for language model integration in encoder-decoder speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 369–375, 2018. doi: 10.1109/SLT.2018.8639038.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

[40] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf`.

[41] Wayne Ward and Sunil Issar. A class based language model for speech recognition. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 416–418. IEEE, 1996.

[42] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1410.3916`.

[43] Minghao Wu, Fei Liu, and Trevor Cohn. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1310. URL `https://www.aclweb.org/anthology/D18-1310`.

[44] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL `https://www.aclweb.org/anthology/P19-1139`.

[45] Ding Zhao, Tara N. Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. Shallow-Fusion End-to-End Contextual Biasing. In *Proc. Interspeech 2019*, pages 1418–1422, 2019. doi: 10.21437/Interspeech.2019-1209. URL `http://dx.doi.org/10.21437/Interspeech.2019-1209`.

A Appendix

A.1 Experimental Setup

Our training pipeline consists of three stages: (1) Pre-training a language model (2) Training entity models and (3) Training a contextual fusion layer to combine the pre-trained model with entity models. We only use transformer decoder blocks [39] and linear layers in all our networks. We use T_{layers} to represent the number of layers in the transformer, d_{model} for its hidden, input and output dimensions, d_{ff} for the feedforward dimension, T_{heads} as the number of attention heads and $T_{dropout}$ for dropout between the layers. All our models use the AdamW optimizer [20] with a weight decay of 0.1 and a step-wise decaying learning rate scheduler [10] with warmup. We use lr_{warmup} to define number of tokens until warmup and lr_{decay} to define the number of tokens after which we decay the learning rate by a factor of 0.9. We use lr_{start}, lr_{max} and lr_{end} to define the starting, maximum and final learning rates respectively. We conduct all our experiments on utterances belonging to Entertainment, Shopping, LocalSearch and Communication domains.

Pre-trained Language Model Training   Our training data consists of transcribed utterances from user interactions with a voice assistant. The total number of tokens available for training is 40 million. All the utterances are de-identified to preserve the privacy of the users. Our pre-trained model parameters are $T_{layers} = 12$, $d_{model} = 512$, $d_{ff} = 1024$, $T_{heads} = 8$ and $T_{dropout} = 0.1$. The parameters of our learning rate scheduler are $lr_{start} = 1e^{-6}$, $lr_{max} = 6e^{-4}$, $lr_{end} = 1e^{-6}$, $lr_{warmup} = 2^{22}$ and $lr_{decay} = 2^{24}$. Our vocabulary consists of 8192 character level subword byte-pair encodings trained using [34]. We truncate all utterances that are longer than 32 subword units after tokenization and use absolute positional embeddings of size 512 to represent the position of these tokens. We accumulate gradients for two steps before updating the parameters of our network [25]. We do not use any bias parameters for the output linear layer before softmax.

Entity Models Training   We used the following entity models for our experiments: (1) song name, (2) album name, (3) celebrity name, (4) video name, (5) shopping item, (6) place name and (7) person name. The size of our entity catalogues ranges from from 2 million to 10 million entries depending on the entity model. The entities in each catalogue are assigned a popularity score using some simple heuristics and we sample the entities based on their popularity during training. Our entity model parameters are $T_{layers} = 4\text{-}6$, $d_{model} = 512$, $d_{ff} = 1024$, $T_{heads} = 4$ and $T_{dropout} = 0.1$. Our learning rate scheduler parameters are same as the pre-trained model except for $lr_{decay} = 2^{25}$. We use a markov dependency length (Eq. 3) of $k = 4$ for all our entity models.

Contextual Fusion Layer Training   We only use 8% of our training data to train the contextual fusion layer. Our catalogues cover around 40% of the entities present in this data. We use a one layer transformer with new positional embeddings to represent the utterance context from Section 2.3. The output dimension of all our linear layers is 512 except for the ones preceding a softmax layer. We use a dropout of 0.25 on the entity model outputs and 0.1 for the rest of the network. The number of epochs used for training is 4 and our learning rate scheduler parameters are $lr_{start} = 1e^{-6}$, $lr_{max} = 6e^{-5}$, $lr_{end} = 1e^{-7}$, $lr_{warmup} = 2^{22}$ and $lr_{decay} = 2^{22}$. Our batch size is 256 and gradient accumulation is done for two steps before parameter updates. We freeze all the parameters of the pre-trained LM and entity models during training. Our approach is fully self-supervised once we collect decent sized catalogues, containing some percentage of the entities present in the data used to train this layer. Our method works better if we train this layer on entities not seen by the pre-trained LM, however, we did not do that for this paper. We also found that the contextual fusion layer ignores certain entity models if there is a huge imbalance in the data used to train this layer, especially, for entities appearing in similar contexts such as songs and videos.

A.2 Discussion

We discuss the strengths and drawbacks of the EALM in the following sections. We evaluate our model on its ability to represent the long-tail and its effectiveness in continually learning new entities without forgetting old ones.

A.2.1 Evaluation on Long-Tail

We first evaluate our EALM trained using our full catalogues on all our test sets. Entities from the test sets are not manually added to the catalogues and the presence of them is purely coincidental. We have divided our test sets into different groups as shown in Table 1 based on this information. We show the relative reduction in perplexity obtained by our baseline EALM compared to the pre-trained LM, on all our test sets, in the first part of Table 3. We also show a breakdown of these results by entity type obtained by evaluating only on utterances which contain at least one entity of that type. Note that the tail and new test sets only contain entities where each word is seen at least once in the training data. We do this to avoid evaluating on bad transcriptions with misspelled words.

We can see that our baseline EALM consistently outperforms the standalone pre-trained model on all our test sets. The improvement is much more significant on tail entities (10.37%), that are unseen during training, but are present in our catalogues. Surprisingly, we found that our model also does well on new entities, which are not present in our catalogues. We assume this is because of the regularization effect of our entity models on the pre-trained LM, which probably overfits to seen entities. Our pre-trained model performs very poorly on the spanish entities test set as expected, since these entities contain new words not seen during training combined with the fact that our catalogues contain other spanish entities explains the significant boost seen on this test set.

A.2.2 Continual Learning of Popular Entities

Next, we evaluate our model on whether it can continually learn popular entities. To test this, we re-trained all our entity models by adding entities from the tail and new test sets (including dev sets), to the top five percent (ranked by popularity) of our catalogues. We did not retrain the contextual fusion layer. We see a big improvement from 5.63% to 14.38% on utterances containing new entities, refer Table 3. While this is encouraging, we are still significantly worse compared to the performance of our pre-trained LM on the already existing popular entities in the general test set. From Table 2, we can infer that the improvement needed on the entity new test set to match the performance of our pre-trained LM on the entity seen test set is close to 54.83%. However, retraining the pre-trained LM frequently and collecting utterances containing these entities is expensive.

We also evaluated our method on whether it can learn popular entities from other languages, which is not atypical in task-oriented dialogue systems. We can see that our model achieved a huge overall perplexity reduction of 39.12%. Again, this is still significantly lower compared to the 85.8% (refer Table 2) improvement needed to match the performance on existing popular entities.

Notice that even though we see a further reduction in perplexity on the general test set when we retrained our entity models with tail and new entities, because of some possible overlap, we see that our improvement on the general test set reduced from 1.56% to 1.25% after adding the spanish entities. This is expected because the contextual fusion layer was trained on previous entity models. We measure this more accurately in the next section. Unlike other current continual learning approaches, that suffer from significant forgetting, our method only showed a minor degradation on the general test set. This can be explained by Table 6 where we see that the interpolation probabilities of our entity models are almost always capped around 0.05. We have also observed that this number can only reach 0.2 to 0.4 for new/tail entities explaining the difficulty of our system in being able to truly continually learn new entities, and is always limited by the performance of the pre-trained LM on these entities. However, we have seen cases where this number can touch 0.4 to 0.8, in a controlled setting, using fewer entity models, a weaker pre-trained LM and some supervision.

A.2.3 Continual Learning of Tail Entities

Finally, we evaluate whether we can add entities to the tail of our catalogues and still see significant improvements. To test this, we have created three sets of catalogues containing top 25%, 50% and 100% of entities from the original catalogues. To further control our experiments, we removed all entities that are present in the training data and added back 50% of them to these catalogues. We trained two sets of entity models with different random seeds using the 25% catalogue (Entity25, Retrained25) and one set each with the 50% and 100% catalogues (Retrained50, Retrained100). We also artificially created a test set (EntityTailNew) using entities randomly picked from the bottom 75% of our original catalogues. We train our contextual fusion layer using Entity25 and replace it with others for our experiments. We show our results in Table 5.

Our results suggest that replacing the entity models comes at a small cost, since the fusion layer is trained on the previous entity models. However, we are still better than the standalone pre-trained model. We are able to continually learn tail entities but the improvements are smaller compared to learning the popular entities. Note that a lot of our tail entities contain both words that are not seen during training and misspelled words, which explains why our pre-trained model does poorly on these entities compared to the EntityNew test set, refer Table 2.

Table 4: Example showing the relevant utterance context and interpolation probabilities using $k = 4$. We have created this utterance artificially and it is not taken from our de-identified customer data.

	Play	a	sky	full	of	stars	by	coldplay
Columns indicate the relevant utterance contexts at different timesteps.								
Song Model	\<s\>	\<s\>	a \<s\>	sky a \<s\>	full sky a \<s\>	of full sky a	stars of full sky	\<s\>
Celeb Model	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>
Other Models	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>	\<s\>

Table 5: Table showing percentage reduction in perplexity compared to the pre-trained LM.

Test set	EALM trained with Entity25	EALM + Retrained25	EALM + Retrained50	EALM + Retrained100
General	1.55	1.32	1.33	1.33
EntityTailNew	14.52	14.02	16.95	17.97

Table 6: Table showing the interpolation probabilities of our EALM based on Eq. 7. 'a sky full of stars' is a song and 'a sky full of stars for a roof' is an album (soundtrack) present in our catalogues.

	Play	a	sky	full	of	stars	by	coldplay
Pre-trained LM	1	0.92	0.94	0.92	0.93	0.85	0.97	0.97
Song Model	0	0.02	0.03	0.03	0.03	0.06	0.01	0.01
Album Model	0	0.01	0.02	0.02	0.03	0.08	0.02	0.01
Celeb Model	0	0.02	0.01	0.02	0	0	0	0.01
Video Model	0	0.01	0	0.01	0	0.01	0	0
Item Model	0	0.02	0	0	0	0	0	0
Place Model	0	0	0.01	0	0	0	0	0
Contact Model	0	0.01	0	0	0	0	0	0