On-The-Fly Information Retrieval Augmentation for Language Models

Hai Wang **David McAllester**
Toyota Technological Institute at Chicago, Chicago, IL, USA
{haiwang,mcallester}@ttic.edu

Abstract

Here we experiment with the use of information retrieval as an augmentation for pre-trained language models. The text corpus used in information retrieval can be viewed as form of episodic memory which grows over time. By augmenting GPT 2.0 with information retrieval we achieve a zero shot 15% relative reduction in perplexity on Gigaword corpus without any re-training. We also validate our IR augmentation on an event co-reference task.

1 Introduction

We are interested in exploring the value of long term episodic memory in language modeling. For example, a language model can be used in January to assign a probability distribution over the statements that will appear in the newspaper in March. But one month later, in February, the distribution over the predictions for March should be updated to take into account factual developments since the previous prediction. Long term episodic memory should be taken into account when assigning a probability to a statement.

Here we take a simple approach in which a pre-trained GPT language model (Radford et al., 2018a, 2019) is zero-shot augmented with an episodic memory consisting simply of a corpus of past news articles. Conceptually the past news articles are viewed as additional training data which can be legitimately accessed when evaluating on future text. In our most basic experiment we calculate the probability of a future article by first calculating the probability of its first k sentences using the pre-trained GPT model. We then use the first k sentences as a query in an information retrieval system to extract a relevant past article. We then insert the past article following the first k sentences when calculating the probability of the remainder of the future article using the same pre-trained GPT model. This is a zero-shot augmentation in the sense that there is no additional training or fine tuning of the pre-trained model. Our results show that this augmentation significantly reduces perplexity. We also present various other experiments including results on fine-tuning the model in the presence of the memory and the effect of this memory on event co-reference.

2 Related Work

Various language models have utilized external knowledge or long contexts (Paperno et al., 2016; Yang and Mitchell, 2017; Peng et al., 2019; Khandelwal et al., 2018; Ghosh et al., 2016; Lau et al., 2017; Grave et al., 2016; Parthasarathi and Pineau, 2018). But these papers do not address the question of whether additional context or external knowledge is useful as a zero-shot augmentation of large scale pre-trained NLP models.

The value of external knowledge has previously been demonstrated for NLP tasks such as natural language inference (Chen et al., 2018; Yang et al., 2019), language generation (Parthasarathi and Pineau, 2018), knowledge base completion (Toutanova et al., 2015; Das et al., 2017) and question answering (Sun et al., 2019, 2018; Dhingra et al., 2017). However, all those prior works assume the model is small and trained from scratch.

As large scale pre-trained models have become more powerful it is not immediately clear whether external resources can still add value. The only work we know of on using external resources in modern large scale models is Yang et al. (2019) where a human curated external lexical resource is used to improve BERT.

Our approach bears some resemblance to neural cache models (Grave et al., 2016). However, neural cache models store past hidden states as memory

Proceedings of the 1st Joint Workshop on Narrative Understanding, Storylines, and Events, pages 114–119

and accesses them through a dot product with the current hidden states. This is different from retrieving knowledge from a corpus-sized memory.

Our approach is also somewhat related to memory networks (Weston et al., 2014). Memory networks have a memory module which can be learnt jointly with other components. It has shown success in applications such as machine reading comprehension (Kumar et al., 2016a,b; Shi et al., 2016) and visual question answering (Na et al., 2017; Ma et al., 2018; Su et al., 2018). Significant progress in memory networks has been achieved in both architecture (Chandar et al., 2016; Miller et al., 2016; Gulcehre et al., 2017) and model scale (Rae et al., 2016; Lample et al., 2019).

Several papers have formulated, and experimented with, scalable memory networks — memory networks that employ some method of efficiently reading and writing to very large neural memories. This is done with approximate nearest neighbor methods in Rae et al. (2016) and with product keys in Lample et al. (2019). These large memories are used to provide additional model capacity where the memory contents are trained over a large data set using gradient descent training, just as one would train the parameters of a very large network. It is shown in Lample et al. (2019) that it is possible to insert a large memory as a layer in a transformer architecture resulting a model where the same number of parameters and the same performance can be achieved with half the layers and with much faster training time than a standard transformer architecture. Here, however, we are proposing zero-shot augmentation with an external data source used as an episodic memory.

The use of key-value memories in Miller et al. (2016) is particularly similar to our model. Key-value memories were used there in treating a corpus of Wikipedia movie pages as a memory for answering questions about movies. As in our system, articles were extracted using word based information retrieval. Each article was encoded as a vector which was then given to a question answering architecture. This was shown to improve on automated knowledge base extraction from the same corpus but was still not competitive with human curated knowledge graphs for movies. Here we give the text of the retrieved article directly to the language model architecture and focus on augmenting large scale language models.

3 Model

We use the pre-trained transformer GPT 2.0 (Radford et al., 2019). Let W_w and W_p be the subword and position embeddings respectively. Let M denote the total number of layers, for a token at time step t, the m-th layer's hidden state h_t^m is given by:

$$h_t^m = \begin{cases} W_w + W_p & \text{if } m = 0 \\ \text{TB}(h_t^{m-1}) & \text{if } 1 \leq m \leq M \end{cases}$$

where TB stands for Transformer Block. We use last layer's hidden state h_t^M as the presentation H_t for the token at time step t. We augment GPT 2.0 with a large episodic memory component, and the overall architecture is shown in Figure 1.

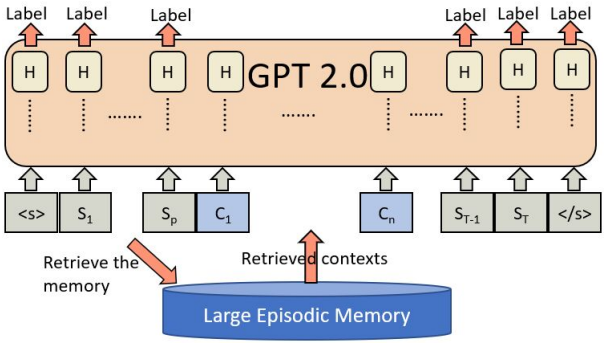

Figure 1: GPT with large episodic memory component

For a sequence S with T tokens, let $S_1, \ldots, S_p$ be the tokens of the first k sentences. Let C be a sequence (article) retrieved from memory using the first k sentences as the query, the vector H_t is:

$$H_t = \begin{cases} \text{GPT}(S_1, \ldots, S_t), \text{if } t \leq p \\ \text{GPT}(S_1, \ldots, S_p, C, \ldots, S_t), \text{otherwise} \end{cases}$$

That's to say, for the first k sentences, we directly feed them to GPT to obtain their representations. For remaining sentences, their representations are conditioned on both the first k sentences and the retrieved context C. Table 1 compares features of our simple memory augmentation with those of other memory models.

4 Experiments

We focus on two tasks: document level language modelling and event co-retrieved . In both tasks we take a document as input and use first k sentences to query the memory. To calculate the perplexity of a document, we compute the log-probability of a document by multiplying byte level probability,

Model	episodic	search	memory size
DMN	yes	exact	∼1K words
SAM:	no	approx	∼100K slots
KVM:	yes	exact	$\leq$ 1M slots
LMN:	no	exact	∼1M slots
Ours:	yes	approx	∼10M documents

Table 1: Comparison between different models. DMN: Dynamic Memory Network (Kumar et al., 2016b); SAM: Sparse Access Memory (Rae et al., 2016); KVM: Key Value Memory (Miller et al., 2016); LMN: Large Memory Network (Lample et al., 2019). Memory size is measured in their own words.

then divide the log-probability by the actual word count in the *query* document.

We use Gigaword (Parker et al., 2011) as both our language modeling test set and as our external memory. Gigaword contains news from different sources such as NY Times and XinHua News etc. For language modelling we use the NY Times portion because it is written by native English speakers. Since GPT 2.0 is trained on Common Crawl which contains news collections started from 2008. To avoid testing on GPT-2 training data, we use Gigaword articles collected prior to 2008. For the pre-trained language model we use GPT 2.0 (Radford et al., 2019) [1]. It contains three pre-trained models: GPT Small, Medium and Large.

For information retrieval we use Lucene due to its simplicity. Given a query document we first do sentence and word tokenization and then use the first k sentences to retrieve top 20 retrieved documents with the default TF-IDF distance metric provided by Lucene. Since too distant document pairs are uninformative and too related document pairs tends to be duplicates of the test article, we further filter those top ranked documents by time stamp, news source and cosine similarity. More specifically, we choose the highest ranked retrieved document that simultaneously satisfies the following three conditions: it comes from a different news source; it appears earlier but within two weeks time window of the test document, and the bag of word cosine similarity between the test and the retrieved cannot be larger than 0.6α where α is the largest bag of word cosine similarity between the test article and any retrieved articles. To support fine-tuning experiments we constructed a corpus of pairs of a *query* article and a cached *retrieved* document. We split the dataset into train/dev/test by query document's time stamp. The train/dev/test size is: 79622,16927,8045. For zero-shot experiments we use the test set of 8045 articles. We do experiments with $k \in \{1, 2, 5\}$.

To check the quality of *query-retrieved* pairs, we randomly sample 100 pairs from dev set and compute the bag of word cosine similarity between the two documents. The mean cosine similarity is 0.15. We also manually inspect them: we ask two NLP researchers to annotate the *query-retrieved* pair as "BAD" or "OK" independently, i.e., if two documents are almost duplicates or totally unrelated, then it's "BAD", otherwise, it's "OK". Among 100 pairs, 83 pairs are "OK", 17 pairs are "BAD" due to irrelevance. The Cohen's kappa coefficient between two annotations is 0.94.

4.1 Language modelling

For language modeling we try zero-shot memory augmentation, fine-tuned memory augmentation, and training a small memory-augmented network from scratch. When training, we use the Adam optimizer from GPT 1.0 (Radford et al., 2018b). The learning rate is 0.001, weight decay parameter is 0.01, the warm up proportion is 0.1. For other parameters, we use the default values from GPT 2.0. The fine-tuning on Gigaword takes less than one day with a single GPU.

Zero-shot and fine-tuning results Following Radford et al. (2019), we first evaluate our model on Gigaword with zero-shot setting and then fine-tune the model. The results are given in Table 2.

Model Size	woc	k=1	k=2	k=5
GPT-Small	35.15	29.29	30.54	32.38
GPT-Medium	22.78	19.84	20.54	21.48
GPT-Large	19.90	17.41	18.00	18.80
GPT-Small	23.03	21.01	21.89	22.66

Table 2: Perplexity for zero-shot (top 3 rows) and fine-tuning (last row) settings when use different k to retrieve the context. **woc**: without retrieved context.

From Table 2, we see that with additional context retrieved from episodic memory, for all different GPT models, we obtain significantly lower perplexity than using original GPT 2.0. When fine tuning the model with context, we can further reduce the overall perplexity. We only fine tune GPT small due to our GPU memory constraints. Preliminary

[1] https://github.com/huggingface/pytorch-transformers

analysis indicates that most of the perplexity reduction comes at content words and semantically rich words where predictions require broader context. This is consistent with the phenomena found in Khandelwal et al. (2018). We further find that smaller k leads to slightly worse retrieval quality, however, more continued sentences will benefit from the retrieved context. Since Gigaword contains newswire, the first several sentences usually are importation summarizations, thus overall, smaller k will result in lower perplexity.

Train from scratch We also investigate training this form of memory-augmented model from scratch on our query-retrieved pairs. For these experiments we train smaller transformers and the results are given in Table 3. From Table 3, we see that additional context still helps and we can get decent perplexity even with quite small models.

Model Config	**woc**	**k=1**	**k=2**	**k=5**
E=384,H=6,L=6	35.62	31.94	33.18	35.26
E=384,H=8,L=8	33.67	29.62	30.76	32.73
E=576,H=8,L=8	31.32	27.38	28.54	30.63

Table 3: Perplexity when train from scratch. E: hidden states dimensionality; H: # of head; L: # of layer. GPT-Small has the configuration: E=764, H=12, L=12.

When context is irrelevant We also evaluate our method on Wikitext-2/103, in which the retrieved context is irrelevant due to domain difference between Wikipedia and Gigaword. In this case, we use the most top ranked document from Gigaword as reference. Table 4 shows that irrelevant contexts have very little impact on perplexity.

Dataset	**woc**	**k=1**	**k=2**	**k=5**
Wikitext-2	28.67	28.96	28.95	28.70
Wikitext-103	25.38	25.68	25.56	25.39

Table 4: Zero-shot perplexity using GPT-Small

4.2 Event Co-reference

Intuitively episodic memory is useful because it contains information about the particular events mentioned in the test document. With this in mind we evaluate our approach on the event co-reference dataset ECB+ (Cybulska and Vossen, 2014). ECB+ contains 982 documents clustered into 43 topics, and has two evaluation settings: coreferring mentions occurring within a single document (within document) or across a document collection (cross document). For the event co-reference pipeline, we follow the joint modeling method of Barhom et al. (2019) where they jointly represented entity and event mentions with various features and learned a pairwise mention/entity scorer for coreference classification. We augment their mention features with the mention's vector representations extracted from either GPT 2.0 or our zero-shot augmented GPT 2.0. For event co-reference, we use the whole test document to retrieve the context from Gigaword. From Table 5, we see that the context can help boost the CONLL F1 score.

System	MUC	B^3	CONLL
Within Document			
KCP	63.0	92.0	81.0
JM	70.9	93.5	85.1
JM+GPT	80.1	93.5	85.2
JM+GPT+CTX♣	80.2	93.9	85.4
Combined Within and Cross Document			
CV	73.0	74.0	73.0
KCP	69.0	69.0	69.0
JM	80.9	80.3	79.5
JM+GPT	81.2	80.2	79.6
JM+GPT+CTX♣	81.3	80.5	79.8

Table 5: F1 score on ECB+ dataset. KCP: Kenyon-Dean et al. (2018) where they add a clustering-oriented regularization term; CV: Cybulska and Vossen (2015) where they add the feature calculated from "event template"; JM: Barhom et al. (2019). ♣: we also feed the retrieved context to GPT to get the representation.

5 Conclusion

In this paper we propose a method to augment a pre-trained NLP model with a large episodic memory. Unlike previous work, we use information retrieval to handle a large external corpus of text and feed retrieved documents directly to language models. Evaluation results on language modelling and event co-reference show the promise of our method. To the best of our knowledge, this is the first work that augments pre-trained NLP models with large episodic memory. In principle, the memory-augmented GPT-2 can be used as a variant of GPT-2 for any downstream tasks, such as GLUE tasks (Wang et al., 2018), although we have not experimented with that here.

References

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. pages 4179–4189.

Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesauro, and Yoshua Bengio. 2016. Hierarchical memory networks. *arXiv preprint arXiv:1605.07427*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2406–2417.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 4545–4552.

Agata Cybulska and Piek Vossen. 2015. Translating granularity of event slots into features for event coreference resolution. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 1–10.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, and Andrew McCallum. 2017. Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning. *arXiv preprint arXiv:1711.05851*.

Bhuwan Dhingra, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2017. Linguistic knowledge as memory for recurrent neural networks. *arXiv preprint arXiv:1703.02620*.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2016. Improving neural language models with a continuous cache. *arXiv preprint arXiv:1612.04426*.

Caglar Gulcehre, Sarath Chandar, and Yoshua Bengio. 2017. Memory augmented neural networks with wormhole connections. *arXiv preprint arXiv:1701.08718*.

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 1–10.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016a. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016b. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2019. Large memory layers with product keys. In *Advances in Neural Information Processing Systems*, pages 8546–8557.

Jey Han Lau, Timothy Baldwin, and Trevor Cohn. 2017. Topically driven neural language model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 355–365.

Chao Ma, Chunhua Shen, Anthony Dick, Qi Wu, Peng Wang, Anton van den Hengel, and Ian Reid. 2018. Visual question answering with memory-augmented networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6975–6984.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409.

Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. 2017. A read-write memory network for movie story understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 677–685.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. 2016. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword. *Linguistic Data Consortium*.

Prasanna Parthasarathi and Joelle Pineau. 2018. Extending neural generative conversational model using external knowledge sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 690–695.

Haoruo Peng, Qiang Ning, and Dan Roth. 2019. KnowSemLM: A Knowledge Infused Semantic Language Model. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018a. Improving language understanding by generative pre-training. *OpenAI Blog*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018b. Improving language understanding by generative pre-training. In *Preprint*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Jack Rae, Jonathan J Hunt, Ivo Danihelka, Timothy Harley, Andrew W Senior, Gregory Wayne, Alex Graves, and Timothy Lillicrap. 2016. Scaling memory-augmented neural networks with sparse reads and writes. In *Advances in Neural Information Processing Systems*, pages 3621–3629.

Jing Shi, Yiqun Yao, Suncong Zheng, Bo Xu, et al. 2016. Hierarchical memory networks for answer selection on unknown words. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2290–2299.

Zhou Su, Chen Zhu, Yinpeng Dong, Dongqi Cai, Yurong Chen, and Jianguo Li. 2018. Learning visual knowledge memory networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7736–7745.

Haitian Sun, Tania Bedrax-Weiss, and William W Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. *arXiv preprint arXiv:1904.09537*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4231–4242.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916*.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1436–1446.

Xiaoyu Yang, Xiaodan Zhu, Huasha Zhao, Qiong Zhang, and Yufei Feng. 2019. Enhancing unsupervised pretraining with external knowledge for natural language inference. In *Canadian Conference on Artificial Intelligence*, pages 413–419. Springer.