Symbolic Semantic Memory in Transformer Language Models

Robert Morain
Department of Computer Science
Brigham Young University
Provo, Utah
rmorain2@byu.edu

Kenneth Vargas
Department of Computer Science
Brigham Young University
Provo, Utah
kenneth.vargas.rivas@gmail.com

Dan Ventura
Department of Computer Science
Brigham Young University
Provo, Utah
ventura@cs.byu.edu

2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA) | 978-1-6654-6283-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICMLA55696.2022.00166

***Abstract*—This paper demonstrates how transformer language models can be improved by giving them access to relevant structured data extracted from a knowledge base. The methods for doing so include identifying entities in a text corpus, sorting the entities using a novel attention-based approach, linking entities to a knowledge base, then extracting and filtering the knowledge to create a knowledge-augmented dataset. We evaluate these methods with the WikiText-103 corpus using standard language modeling objectives. These results show that even simple additional knowledge augmentation leads to a reduction in validation perplexity by 81.04%. These methods also significantly outperform common ways of improving language models such as increasing the model size or adding more data.**

***Index Terms*—language modeling, knowledge integration, natural language processing**

I. Introduction

Currently, transformer language models are the gold standard [1] for most language tasks. The strength of these models comes from their extensive pretraining on statistical language modeling tasks.

In recent years, many improvements have been made to transformer language models. These improvements include adding layers and parameters [2] as well as making changes to the model architecture [3]. There has also been work put into prompt engineering [4] to help guide the model to produce some desired output. The work presented here demonstrates a different approach aimed at improving a transformer language model's ability to access semantic information.

One problem with traditional self-supervised language modeling tasks is that semantic knowledge is only tangentially acquired. To gain more semantic knowledge, these models typically become larger and are trained with more data. It has been argued elsewhere that these models have no way of reasoning about the knowledge they have acquired and instead are "haphazardly stitching together sequences of linguistic forms...observed in...vast training data, according to probabilistic information about how they combine, but without any reference to meaning" [5]. This work designs simple experiments to demonstrate how language models can be improved by giving them access to additional structured data rather than strictly relying on statistics.

[1]The top 10 models on SuperGLUE's [1] leaderboard are all transformer-based models

To proceed, we draw inspiration from research on semantic memory as a cognitive process [6]. Unlike transformer models, humans rely on their semantic and episodic memory to understand language and decide how to respond. While there are connectionist and symbolic models for semantic memory [7], this work loosely draws inspiration from the symbolic approach. Also, since transformers are connectionist models already, using a symbolic model of semantic memory allows us to create a connectionist-symbolic hybrid which has proven to be effective on certain symbolic tasks [8].

Another significant influence on this work comes from Daniel Kahneman's research on reasoning and decision making. Kahneman proposes a cognitive model which distinguishes between System 1 and System 2 thinking [9]. While System 1 is fast, instinctive, and emotional, System 2 is slower, more deliberative, and more logical. The base transformer model can loosely be compared to Kahneman's System 1 fast thinking, while knowledge base integration resembles the slower System 2. Similar to how the SOAR cognitive architecture attempts to replicate various cognitive processes, a transformer-knowledge base hybrid seems appropriate in this case [10].

The idea of combining connectionist and symbolic models has recently become more popular. Bosselut *et al.* introduced Commonsense Transformers (COMET) [11], a GPT-2 model trained on ATOMIC [12] and ConceptNet [13] to automatically construct knowledge graphs. Miller *et al.* introduces key-value memory networks which are trained on structured data from Wikipedia to improve performance on question answering tasks [14]. The CLEVR dataset [15] is a visual question answering dataset specifically designed to test a model's reasoning abilities by minimizing biases that models can exploit. This work shares the goal of improving a model's reasoning abilities through symbolic methods.

The contributions of this paper include:

- Methods for augmenting language datasets with knowledge base data including attention-based entity selection.
- Modification to GPT-2's causal mask to attend to additional knowledge data.
- Demonstrating the effectiveness of knowledge augmented data on language modeling, which reduces perplexity by 81.04%.

978-1-6654-6283-9/22/$31.00 ©2022 IEEE
DOI 10.1109/ICMLA55696.2022.00166

II. Related Work

There is an extensive body of research on knowledge augmented language models from which this paper takes inspiration. REALM uses a knowledge retriever which allows a language model to attend to relevant documents from a large corpus such as Wikipedia [16]. While incorporating increasingly larger contexts into language models has proven to be a fruitful vein of research, Guu *et al.* claim to extend this progression beyond sentences and paragraphs all the way to the corpus level. We follow their lead by integrating knowledge from a large knowledge source. However, this work explores the potential of structured knowledge from a knowledge base rather than a document database.

KEPLER integrates knowledge into a language model by jointly training on language modeling and knowledge embedding objectives [17]. This causes the language model to build entity representations that contain an entity's description and are close to other linked entities in a knowledge base. Here, we utilize structured knowledge without requiring auxiliary objective functions.

KALM augments a recurrent neural network [18] by allowing the model to sample from vocabularies collected from a knowledge base and grouped by entity type. While KALM learns to identify entities in an unsupervised fashion, we follow the lead of models like KnowBert [19] which requires entity selection and linking. KnowBert implements a Knowledge Attention and Recontextualization mechanism which can be added to a layer in the language model to integrate knowledge from a knowledge base. This method combines pretrained entity embeddings with BERT-encoded token hidden states. Unlike KnowBert, our methods utilize a single model to encode and attend to the added knowledge.

III. Methods

These methods outline one way to augment a text corpus with semantic knowledge. While this process is general to any dataset or knowledge base, a description of the specific implementation details are provided.

A. Augmented Dataset Creation

The augmented dataset creation process is composed of a *knowledge base*, an *entity selection* algorithm, a *knowledge extraction* process, and a *merge* between the extracted knowledge and the original dataset. Figure 1 shows how these systems work together to select the supplementary data to include in the augmented dataset. For each sequence in the dataset, a set of entities is extracted from the sequence and filtered based on the entity selection algorithm to a single entity. This entity is then used to query the knowledge base to return a set of knowledge statements. The knowledge statements are filtered and the remaining statements are concatenated to the end of the original text sequence. Once the entire dataset is augmented, it is ready to be used for training the knowledge transformer model.

Dataset: The WikiText-103 dataset [20] is used to train each of the models. After removing rows with no entities in the knowledge base, this dataset consists of a split with 628,965 (99.8%) training rows and 1,320 (0.2%) validation rows[2]. The WikiText language modeling dataset is a collection of over 100 million tokens extracted from the set of verified Good and Featured articles on Wikipedia. It makes sense to use this dataset for fine-tuning GPT-2 because WebText excludes Wikipedia articles from it's training dataset [21].

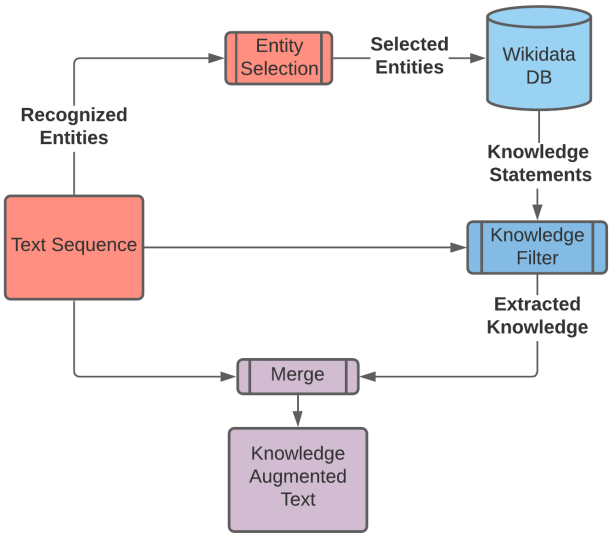

Fig. 1. For each sequence in a language dataset, spaCy is used to identify the entities. Based on the entity selection criteria, a single entity is used to query the Wikidata database to acquire a set of knowledge statements. These statements are then filtered based on their relevance to the corresponding sequence. Lastly, the knowledge for each sequence is concatenated to the end of the original text sequence.

Knowledge base: Wikidata [22] is a free and open knowledge base of structured [23] Wikimedia data. While there is an API to access the official version hosted by Wikidata, this approach proves to be too slow to be used on large datasets. To reduce the time of each Wikidata query, a downloaded JSON data dump of the knowledge base can be converted into a database. While Wikidata supports many languages, only the English entities and properties are used.

In Wikidata, *items* refer to entities in the knowledge base, including people, topics, concepts, and objects. For example, the "1988 Summer Olympics", "love", "Elvis Presley", and "gorilla" are all *items* in Wikidata. A *statement* is defined as a relation between an *item* and a *value* by way of a *property*. *Statements* follow the resource description framework (subject-predicate-object) [24]. Generally, *values* are other *items* but can also be unknown or quantitative values.

Attention-based Entity Selection: The first step in the entity selection process is to identify words in the sequence that may have items in the knowledge base. This problem can be framed as a traditional named-entity recognition task to

[2]These are standard splits from Huggingface's datasets library at https://huggingface.co/datasets/wikitext

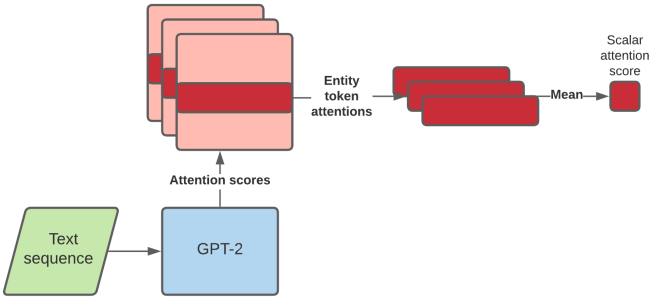

Fig. 2. To compute an entity-specific score: a sequence is input to a pretrained GPT-2 model and the token attention scores are returned in an $n \times n \times h$ tensor, where n is the sequence length and h is the number of heads; the scores of each entity are isolated by selecting only the rows corresponding to the tokens of the entity in question; the scores are averaged across heads, the sequence length, and the entity tokens. The result is a single average attention score value for each entity in the sequence. This allows the entities to be sorted by their average attention score.

identify predefined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc. For example, with the sentence "Apple is looking at buying U.K. startup for $1 billion", Apple, U.K., and $1 billion are considered entities. While there are many methods for named-entity recognition, spaCy[3] is used to extract the named-entities in each sequence.

Once a set of entities are identified in a text sequence, a single entity is included in the augmented dataset based on the entity selection criteria. In these experiments, an entity's average attention score is determined by inputting a sequence of text into a pretrained GPT-2 small model and returning the attention scores for each model layer. These attention scores are then averaged across layers, heads, sequence, and entity tokens—resulting in a single attention score for each entity in the sequence. These entities are then sorted by their (averaged) attention score to determine the maximum, median, and minimum attention-score entity. We interpret this ordering of entities to indicate their relative importance as determined by GPT-2 in the context of the input sequence. Figure 2 provides more details about how these attention scores are calculated.

Knowledge Extraction: In general, the knowledge extraction process consists of querying the knowledge base for information and then filtering that information based on the input sequence. For these experiments, all of the knowledge is filtered out except for the description of the entity. This information is recorded in JSON format like so: $\{$*label : description*$\}$.

While this knowledge extraction process is simple, the purpose of this work is not to optimally select the best possible knowledge for a given input sequence. For now, it is sufficient to show that even naive semantic data can improve performance on language tasks. The task of developing more sophisticated knowledge extraction methods is left to future work.

Merge: The augmentation process is complete once the extracted knowledge is combined with the original dataset. This is done by concatenating the knowledge text to the end of the input text.

IV. Experiments

The knowledge augmentation process is evaluated on a language modeling task. Since a language model is often indirectly asked to exploit the semantic knowledge it has acquired through pre-training, it stands to reason that language modeling would benefit from adding supplementary knowledge to the dataset.

A. Knowledge-augmented Language Modeling

This section describes the details for handling a language modeling task with additional semantic knowledge added to the dataset. While modifications to a model's architecture are not always necessary, some changes may be required to allow the model to exploit this added knowledge.

1) Knowledge Model: GPT-2 is used as the base model for causal language modeling on each knowledge augmented dataset. The causal mask of Huggingface's [25] implementation of the standard GPT-2 architecture must be modified to allow the model to attend to the knowledge tokens at the end of the input (see Figure 3).

While the text tokens are masked normally, the knowledge tokens are always attended to. One could argue that this gives an unfair advantage to the knowledge model over the baseline GPT-2 model because the added tokens bias the model early on in the sequence. First of all, we argue that this bias is a good thing because it more closely resembles how a human might rely on their semantic memory while reading or listening. In spite of this, these results include experiments with a smaller, filtered dataset where additional contiguous text populates the knowledge tokens buffer (a sliding window over the entire dataset is not used). This filtered dataset only consists of rows that have excess knowledge tokens (333,753 train, 877 validation). Comparing this dataset with an identical dataset where knowledge tokens fill the knowledge buffer considers the benefit of added semantic knowledge vs. the benefit of additional textual context. We also include results using an attention mask which eliminates entity leakage from the added knowledge.

2) Baseline: The baseline for this experiment is an unmodified pretrained GPT-2 small model fine-tuned on the WikiText dataset. This is sufficient to observe the effect that additional knowledge has on language modeling. All of the models are trained using a batch size of 32, a learning rate of $1\mathrm{e}-4$, and the ADAM optimizer [26]. The early stopping criteria terminates training after three epochs of no improvement to the validation perplexity. The length of the text sequence is limited to 128 tokens while leaving a 64 token buffer available for knowledge tokens.

[3]https://spacy.io/

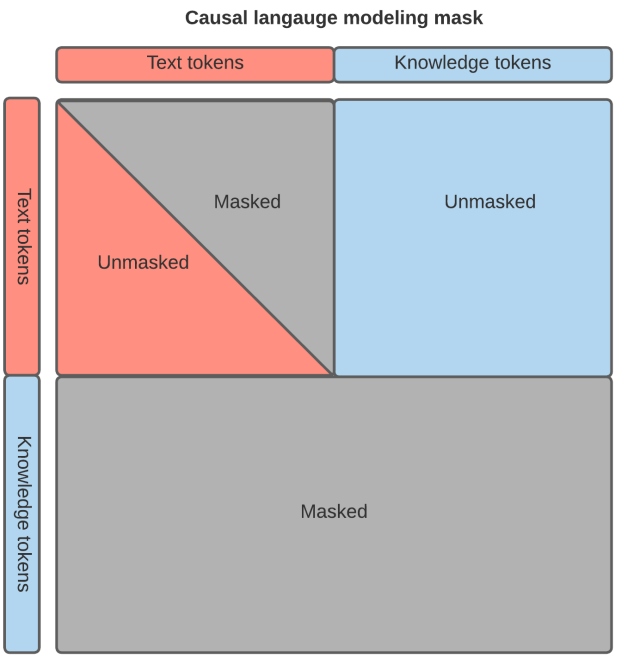

Fig. 3. Modifications to the causal mask of Huggingface's GPT-2 to allow the text tokens to attend to knowledge tokens for each prediction. Recall that attention scores are calculated from each token to every token in the sequence, resulting in an $n \times n$ attention matrix. The text tokens are masked normally, with one less token being masked for each row until all but one of the tokens is unmasked (upper left quadrant). In the upper right quadrant, all of the knowledge tokens are unmasked to allow each unmasked text token to attend to the knowledge tokens. Since the model does not predict on the knowledge tokens, all tokens are masked on the lower half of the causal mask.

3) Entity Selection Criteria: These experiments focus on three primary variations of entity selection. As discussed previously, the average attention score for each entity is calculated using the attention scores output by a pretrained GPT-2 model (Figure 2). Given a list of entities ordered by attention score, the performance when using the maximum-, median-, or minimum-attention score as the entity selection criterion is compared. Based on which entity is selected, the corresponding description is retrieved from the knowledge base to form a knowledge statement. This knowledge statement populates the knowledge buffer as described previously. The four possible combinations of these primary variations (max/median, max/min, median/min, max/median/min) are also tested. The differences in performance of each of these variations provides insight into whether the entity selection criteria has an effect on the performance of a task.

V. Results

These results show that even simple knowledge augmentation can dramatically improve performance on language modeling tasks. At its best, the validation perplexity decreases by 81.04%. These improvements persist even as the number of model parameters increases and with a different model architecture (BERT) on a masked language modeling task.

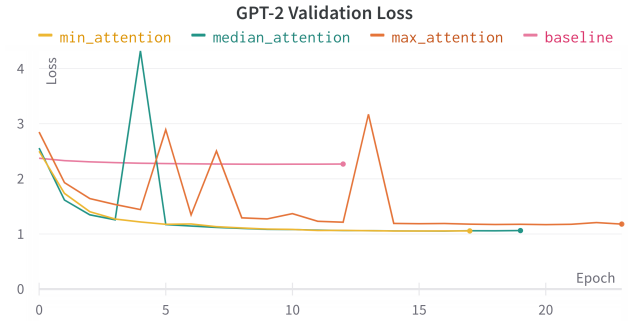

Fig. 4. Validation loss curves for language modeling on GPT-2 small. The min attention knowledge augmentation strategy decreases perplexity by 70.29%.

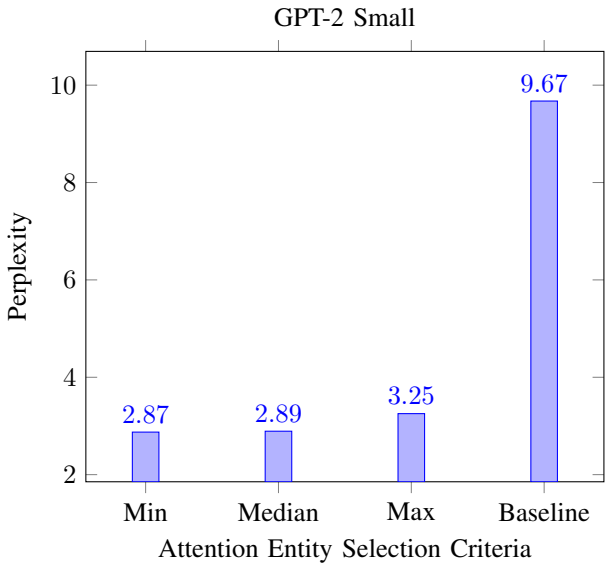

Fig. 5. The validation perplexity for each augmentation strategy when language modeling on GPT-2 small.

A. Knowledge Language Modeling

1) GPT-2 Small: Figures 4 and 5 illustrate how the best knowledge augmentation strategy (min attention) causes the baseline validation perplexity to decrease by 70.29%. Of the three knowledge augmentation strategies tested, min attention performed the best, followed closely by the median attention (0.19% worse), and then max attention (3.92% worse than min attention).

Based on these results, all of these knowledge augmentation strategies are effective in improving the performance of pretrained transformer language models on causal language modeling fine-tuning tasks. This increased performance is achieved without optimizing the knowledge augmentation process—instead, simply adding a *label:description* relation to the knowledge buffer. It stands to reason that the perplexity could be reduced further by adding more relevant data to the knowledge buffer.

While the difference in minimum validation perplexity

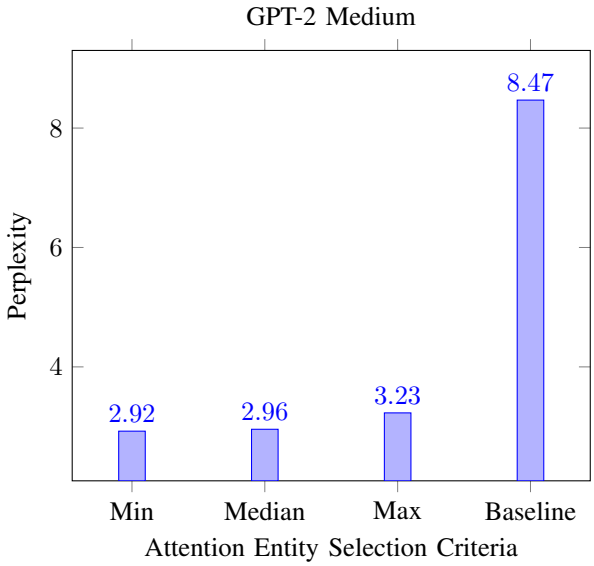

Fig. 6. Validation perplexity for each knowledge augmentation strategy when language modeling on GPT-2 medium. Knowledge augmentation methods continue to improve performance even on larger language models.

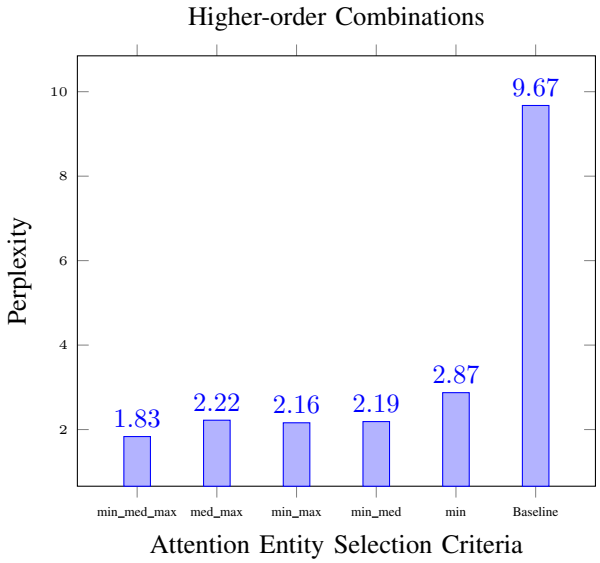

Fig. 7. The validation perplexity for higher-order combinations of knowledge data when language modeling on GPT-2 small. Augmenting with minimum, median, and maximum knowledge entities yields the best results reducing the baseline validation perplexity by 81.04%.

between the min and median attention runs is small, the min attention runs consistently outperform the max attention runs. This may be caused by GPT-2 assigning lower attention scores to entities it does not know very well and higher attention scores to entities it recognizes. By this logic, the additional data about an already common entity could be seen as redundant while the description of an unknown entity might be vital information.

2) GPT-2 Medium: While GPT-2 small has 85M parameters, GPT-2 medium increases this by 313% to 354M. Despite these added parameters, the validation perplexity for GPT-2 medium only decreases by 12.46% when compared to GPT-2 small (compare baselines from Figures 5 and 6). However, adding min attention knowledge to GPT-2 medium decreases the validation perplexity by 69.78% from the GPT-2 small baseline. The difference between min, median, and max is comparable to the experiments with GPT-2 small.

The knowledge augmentation approach remains effective even as models get larger. This suggests that even very large models such as GPT-3 [2] could benefit from additional knowledge. In fact, the knowledge augmentation is more effective than increasing the model size since just adding min attention knowledge to GPT-2 small outperforms GPT-2 medium by 66.07%.

3) Higher Order Combinations of Knowledge: Figure 7 shows the validation perplexity for each combination of knowledge augmentation strategies. As discussed previously, the best first order reduction, min attention, decreases the validation perplexity by by 70.29%. The best second order combination of min and max attention improves over min attention by another 7.37%. Finally, the combination of min, median, and max attention knowledge improves over min_max attention by an additional 3.37% for a total of 81.04% improvement over the baseline. This demonstrates that a higher quantity of added semantic information consistently results in better generalization.

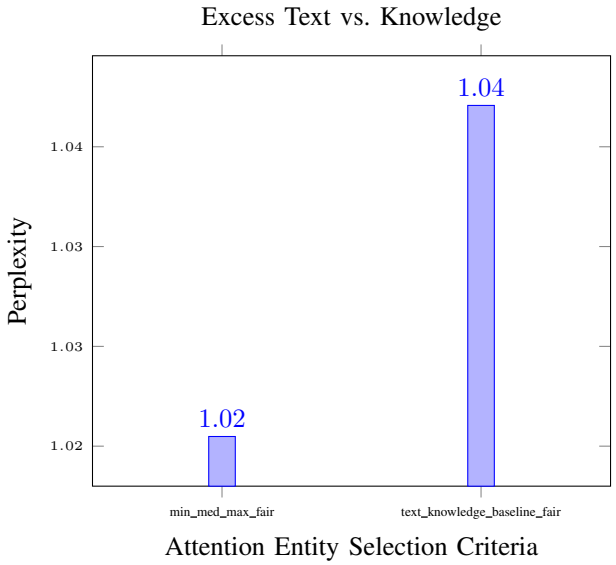

Fig. 8. Validation perplexity when language modeling on GPT-2 small with comparison between filling the knowledge buffer with excess text tokens and filling it with min_med_max knowledge tokens for a dataset where each row has excess tokens.

4) Excess Text Tokens Versus Knowledge Tokens: Figure 8 compares the validation perplexity between filling the knowledge buffer with additional contextual text tokens or min_med_max knowledge tokens. For this dataset, knowledge tokens outperform the text tokens by 1.60%. This suggests that structured semantic data has a distinct advantage over

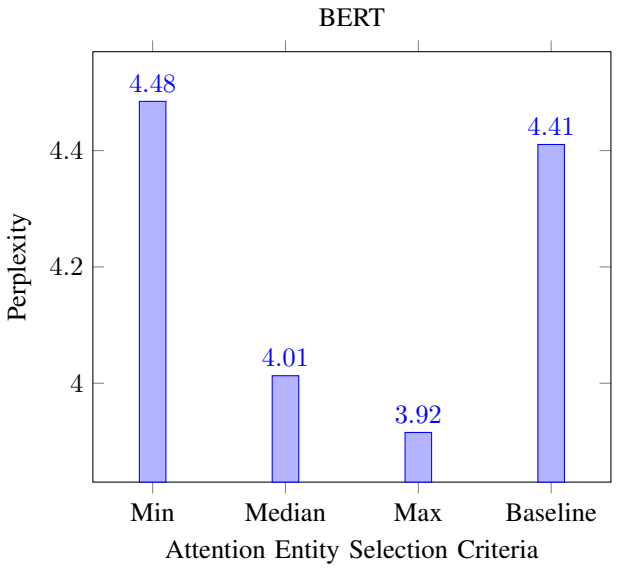

Fig. 9. Validation perplexity when language modeling on BERT. Knowledge augmentation continues to improve results on BERT despite its being pretrained on Wikipedia.

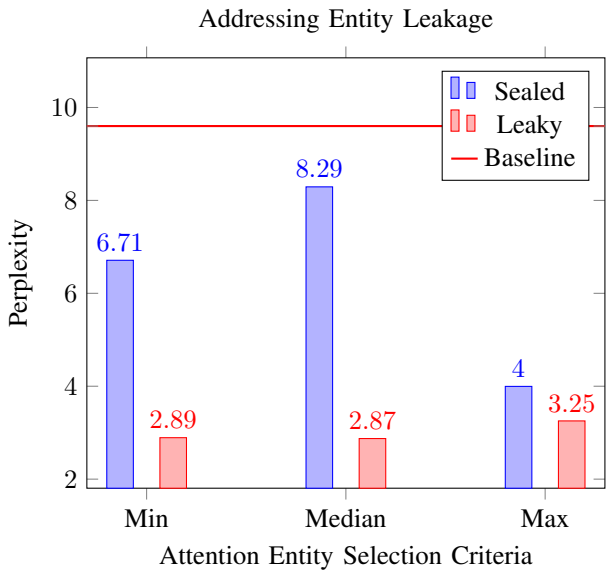

Fig. 10. Validation perplexity when language modeling on GPT-2 small with and without entity leaking.

additional textual context tokens.

5) BERT: Figure 9 once again shows the minimum validation perplexity for the primary knowledge augmentation strategies. In this experiment, only the median and max attention knowledge augmentation runs outperform the baseline, with the max attention variation resulting in the greatest percentage decrease of the baseline perplexity (11.23%). While this reduction in perplexity is not as large as with GPT-2, there are several contributing factors which may explain this. First of all, BERT is pretrained on the English Wikipedia corpus [3] while GPT-2 excludes it. This means the pretrained BERT is already relatively close to it's training limit when fine tuning begins. Secondly, because BERT is a bidirectional model, it can attend to all unmasked tokens in the sequence for each prediction. This makes the added knowledge less useful in disambiguating the subject of the sentence when compared with causal language modeling—especially early on in the sequence. Also, since BERT only predicts on 15% of tokens, these perplexity values are not directly comparable. Taking all this into account, these results continue to validate the efficacy of these methods even across model architectures.

6) Entity Leakage: As shown in Figure 2, the knowledge buffer is always visible for each step in the decoding process. However, since the knowledge tokens contain at least one entity label that occurs in the sequence, this inherently leads to information about upcoming tokens being leaked to the model. Therefore, the knowledge model has an unfair advantage compared to the baseline model in that it is easier to predict at least one entity in the sequence. In order to evaluate the extent to which the knowledge model benefits from entity leakage, additional experiments use an alternative masking approach which seals any potential leaked information. This dynamic knowledge mask causes the knowledge tokens to remain hidden until the last token of the entity label is exposed in the input sequence. All other parts of the dataset augmentation and training processes remain the same.

Results in Figure 10 show each of the entity selection criteria still reduce validation perplexity from the baseline even without entity leakage. Despite the expected drop off in performance from removing entity leakage, max attention beats the baseline by 58.69%. It should also be expected that min and median attention suffer slightly more than max attention due to the fact that the min and median entities are more likely to be positioned toward the end of a sequence. Overall, these results verify the efficacy of these knowledge-augmentation strategies independent of any entity leakage.

VI. Discussion

The purpose of these experiments was to determine whether knowledge-augmented datasets are effective in improving a model's semantic memory. The significant improvement demonstrated on both language modeling shows that knowledge augmentation does make a difference. One possible reason for this could be that statistical language models rely so heavily on the context around the word that the definition of the word itself remains a little too obscure. This may get to the point where words used in an unfamiliar context, possess little meaning and confuse the model. This would explain why including additional semantic information provides a useful bias that keeps unfamiliar words in context. This is further demonstrated by the fact that the min attention entity generally outperformed max and median attention entities. Assuming that the min attention entity is deemed least important by GPT-2, the added knowledge for this entity appears to give new relevance to overlooked or unfamiliar data.

Another way to think about knowledge augmentation is as a form of prompt engineering—where a prompt is automatically generated to help the model with the task at hand.

In these experiments, the entity identification, selection, and knowledge extraction process is relatively simple. In future work, this entire process would be self-directed where a single model learns to use the knowledge base to best suit the task at hand.

As language models becomes more important to society, challenges will arise where language modeling will fall short if it is solely reliant on statistics. Ethical concerns regarding transformer models spreading misinformation and bias have already harmed the reputation of popular language models such as GPT-3 [5]. The principles discussed in this work will directly enable ways of biasing language models away from the dubious ideas present in "wild" training data and towards a shared reality present in curated knowledge bases. Code for this work is available at https://github.com/rmorain/symbolic_semantic_mem.

REFERENCES

[1] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "SuperGLUE: A stickier benchmark for general-purpose language understanding systems," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 3261–3275.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2019, pp. 4171–4186.

[4] N. Zhang, L. Li, X. Chen, S. Deng, Z. Bi, C. Tan, F. Huang, and H. Chen, "Differentiable prompt makes pre-trained language models better few-shot learners," *arXiv preprint arXiv:2108.13161*, 2021.

[5] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 610–623.

[6] E. Tulving, *Episodic and Semantic Memory*. Oxford, England: Academic Press, 1972.

[7] M. N. Jones, J. Willits, S. Dennis, and M. Jones, "Models of semantic memory," in *Oxford Handbook of Mathematical and Computational Psychology*, J. R. Busemeyer, Z. Wang, J. T. Townsend, and A. Eidels, Eds. New York: Oxford University Press, 2015, pp. 232–254.

[8] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *Proceedings of the International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rJgMlhRctm

[9] D. Kahneman, *Thinking, Fast and Slow*. Macmillan, 2011.

[10] J. E. Laird, *The Soar Cognitive Architecture*. MIT press, 2012.

[11] A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Çelikyilmaz, and Y. Choi, "COMET: Commonsense transformers for automatic knowledge graph construction," in *Proceedings of the 57th Conference of the Association for Computational Linguistics*, vol. 1, 2019, pp. 4762–4779.

[12] M. Sap, R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N. A. Smith, and Y. Choi, "Atomic: An atlas of machine commonsense for if-then reasoning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3027–3035.

[13] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4444–4451.

[14] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1400–1409.

[15] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2901–2910.

[16] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M.-W. Chang, "Realm: Retrieval-augmented language model pre-training," *arXiv preprint arXiv:2002.08909*, 2020.

[17] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 176–194, 2021.

[18] A. Liu, J. Du, and V. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 1142–1150. [Online]. Available: https://doi.org/10.18653/v1/n19-1117

[19] M. E. Peters, M. Neumann, R. L. L. IV, R. Schwartz, V. Joshi, S. Singh, and N. 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 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 43–54. [Online]. Available: https://doi.org/10.18653/v1/D19-1005

[20] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *Proceedings of the International Conference on Learning Representations*, 2017.

[21] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[22] D. Vrande, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

[23] M. J. Cafarella, A. Halevy, and J. Madhavan, "Structured data on the web," *Communications of the ACM*, vol. 54, no. 2, pp. 72–79, 2011.

[24] E. Miller, "An introduction to the resource description framework," *Bulletin of the American Society for Information Science and Technology*, vol. 25, no. 1, pp. 15–19, 1998.

[25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *CoRR*, vol. abs/1910.03771, 2019. [Online]. Available: http://arxiv.org/abs/1910.03771

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980