2022-03-16

Symbolic Semantic Memory in Transformer Language Models

Robert Kenneth Morain
Brigham Young University

Symbolic Semantic Memory in Transformer Language Models

Robert Kenneth Morain

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Dan Ventura, Chair
Nancy Fulda
Steven Luke

Department of Computer Science

Brigham Young University

ABSTRACT

Symbolic Semantic Memory in Transformer Language Models

Robert Kenneth Morain
Department of Computer Science, BYU
Master of Science

This paper demonstrates how transformer language models can be improved by giving them access to relevant structured data extracted from a knowledge base. The knowledge base preparation process and modifications to transformer models are explained. We evaluate these methods on language modeling and question answering tasks. These results show that even simple additional knowledge augmentation leads to a reduction in validation loss by 73%. These methods also significantly outperform common ways of improving language models such as increasing the model size or adding more data.

Keywords: natural language processing, knowledge base, semantics

ACKNOWLEDGMENTS

Thank you to my wife, Emi, for your constant love and support; to my advisor, Dr. Ventura, for believing in me and bringing out my best work; and to Jack Demke for the incessant brain picking.

Table of Contents

Chapter 1

In Preparation: Symbolic Semantic Memory in Transformer Language Models

This manuscript has not yet been accepted for publication.

Symbolic Semantic Memory in Transformer Language Models

Robert Morain , **Kenneth Vargas** and **Dan Ventura**
Computer Science Department
Brigham Young University
rmorain@byu.edu, kenneth.vargas.rivas@gmail.com, ventura@cs.byu.edu

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 knowledge base preparation process and modifications to transformer models are explained. We evaluate these methods on language modeling and question answering tasks. These results show that even simple additional knowledge augmentation leads to a reduction in validation loss by 73%. These methods also significantly outperform common ways of improving language models such as increasing the model size or adding more data.

1 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 [Brown *et al.*, 2020] as well as making changes to the model architecture [Devlin *et al.*, 2019]. There has also been work put into prompt engineering [Zhang *et al.*, 2021] to help guide the model to produce some desired output. However, this work demonstrates a different approach aimed to improve 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" [Bender *et al.*, 2021]. 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.

To proceed, we draw inspiration from research on semantic memory as a cognitive process [Tulving, 1972]. 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 [Jones *et al.*, 2015], 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 [Mao *et al.*, 2019].

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 [Kahneman, 2011]. 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 [Laird, 2012].

The idea of combining connectionist and symbolic models has recently become more popular. Bosselut *et al.* introduced Commonsense Transformers (COMET) [2019], a GPT-2 model trained on ATOMIC [Sap *et al.*, 2019] and ConceptNet [Speer *et al.*, 2017] to automatically construct knowledge graphs. Miller *et al.* introduces key-value memory networks [2016] which are trained on structured data from Wikipedia to improve performance on question answering tasks. The CLEVR dataset [Johnson *et al.*, 2017] 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.
- Modification to GPT-2's causal mask to attend to additional knowledge data.

[1] The top 10 models on SuperGLUE's [2019] leaderboard are all transformer-based models

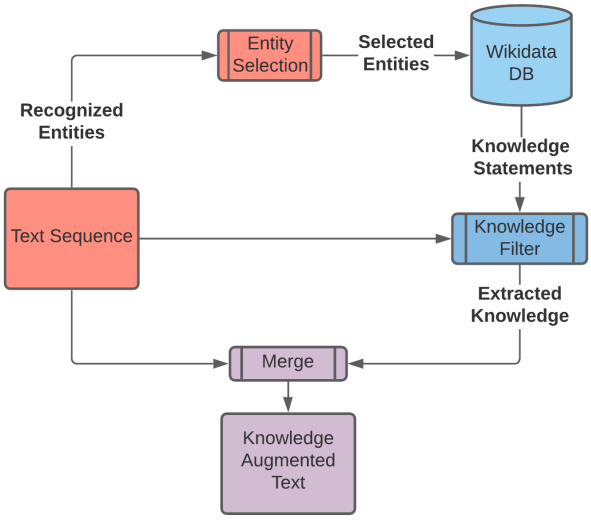

Figure 1: For each sequence in a language dataset, spaCy is used to identify the entities. Based on the entity selection criteria, a single entity is used to query the Wikidata database to acquire a set of knowledge statements. These statements are then filtered based on their relevance to the corresponding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

- Demonstrating the effectiveness of knowledge augmented data on language modeling, which reduces loss by 73%.
- Demonstrating the effectiveness of knowledge augmented data on a multiple-choice question answering task.

2 Methods

In order to test the efficacy of providing semantic knowledge to a transformer language model, there must be a way to augment the standard linguistic training data with the semantic knowledge. The basic steps for doing so are outlined below. While this process is general to any dataset or knowledge base, a description of the specific implementation details are provided. Details regarding the creation of an original question answering dataset are provided as well.

2.1 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 dataset [Merity *et al.*, 2017] 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 [Radford *et al.*, 2019].

Knowledge base: Wikidata [Vrande, 2014] is a free and open knowledge base of structured [Cafarella *et al.*, 2011] 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 SQLite 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) [Miller, 1998]. Generally, *values* are other *items* but can also be unknown or quantitative values.

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 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 average attention score to easily 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

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

[3]https://spacy.io/

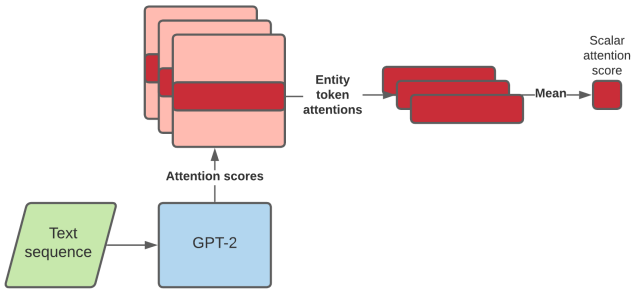

Figure 2: To compute an entity-specific score: a sequence is input to a pretrained GPT-2 model and the token attention scores are returned in an $n \times n \times h$ tensor, where n is the sequence length and h is the number of heads; the scores of each entity are isolated by selecting only the rows 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, which is representative of their relative importance.

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.

2.2 Knowledge Question Answering Dataset

In order to evaluate whether knowledge augmentation improves a model's knowledge of the real world, models are trained on an original question answering dataset. This dataset is created by generating multiple-choice questions of the form shown in Table 1. The data for these questions comes directly from the Wikidata knowledge base, which is queried to get descriptions of an entity. One question is generated for each entity extracted from the WikiText dataset. The distractor choices are randomly selected from other known entities.

The dataset is augmented by adding knowledge tokens between the question and the choices. For this dataset, the added knowledge is the description of the entity in the question. Since the correct answer and the added knowledge are exactly the same, a certain percentage of the knowledge tokens are randomly masked. This is done to determine the limit to which noisy or incomplete knowledge data continues to prove useful.

3 Experiments

The knowledge augmentation process is evaluated on two tasks: language modeling and multiple-choice question answering. Since a language model is often indirectly asked to exploit the semantic knowledge it has acquired through pretraining, it stands to reason that language modeling would benefit from adding supplementary knowledge to the dataset. On the other hand, multiple-choice question answering directly tests a model's real-world knowledge.

3.1 Knowledge-augmented Language Modeling

The term knowledge-augmented language modeling refers to performing 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.

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 [Wolf *et al.*, 2019] 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 considers the benefit of added semantic knowledge vs. the benefit of additional textual context.

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 $1e{-}4$, and the ADAM optimizer [Kingma and Ba, 2015]. The early stopping criteria terminates training after three epochs of no improvement to the validation loss. The length of the text sequence is limited to 128 tokens while leaving a 64 token buffer available for knowledge tokens.

Entity Selection Criteria

These experiments focus on three primary variations of entity selection. As discussed previously, the average attention

Question	Correct Answer	Distractor 1	Distractor 2	Distractor 3
What is {Entity 1 label}?	{Entity 1 description}	{Entity 2 description}	{Entity 3 description}	{Entity 4 description}
What is Stephen Curry?	American basketball player	star in the constellation Ophiuchus	scientific article published on 01 January 1980	species of crustaceans

Table 1: This question answering dataset is generated using the features shown in the table columns. Distractor choices are randomly selected from other entities seen in the WikiText dataset.

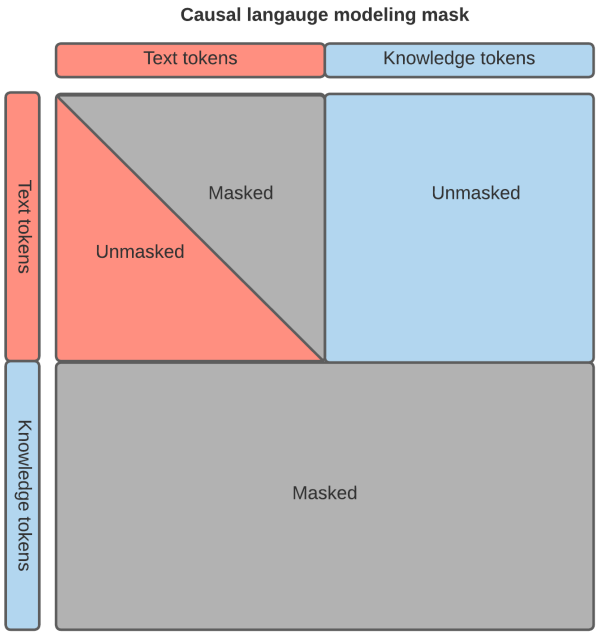

Figure 3: Modifications to the causal mask of Huggingface's GPT-2 to allow the text tokens to attend to knowledge tokens for each prediction. Recall that attention scores are calculated from each token to every token in the sequence, resulting in an $n \times n$ attention matrix. The text tokens are 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.

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.

3.2 Question Answering

In this task, the knowledge model is trained on a custom multiple-choice question answering task. Each question in the dataset has four possible choices, so a random baseline would achieve 25% accuracy. Huggingface's double heads model, which is the recommended model for multiple-choice question answering, is used as a base for the knowledge model. While the model has the option to train on the question answering and language modeling loss, this experiment relies exclusively on the question answering loss for training.

The baseline for this task uses the plain question answering dataset without knowledge augmentation. When the knowledge tokens are added, a certain percentage of knowledge tokens are randomly masked. The percentage of masked knowledge tokens is incrementally reduced until all of the tokens are visible.

4 Results

These results show that even simple knowledge augmentation can dramatically improve performance on language modeling tasks. At its best, the validation loss decreases by 73%. These improvements persist even as the number of model parameters increases and with a different model architecture (BERT) on a masked language modeling task.

4.1 Knowledge Language Modeling

GPT-2 Small

Figures 4 and 5 illustrate how the best knowledge augmentation strategy (min attention) causes the baseline validation loss to decrease by 53.4%. Of the three knowledge augmentation strategies tested, min attention performed the best, followed closely by the median attention (0.26% worse), and then max attention (5.4% 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

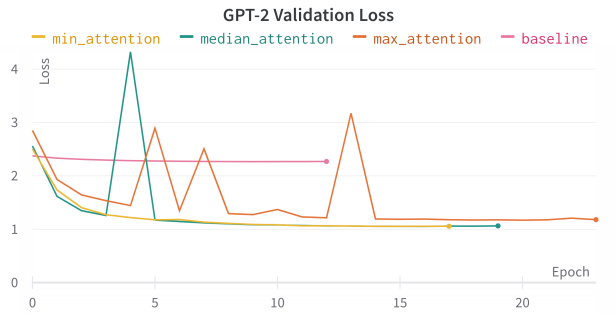

Figure 4: Validation loss curves for language modeling on GPT-2 small. The min attention knowledge augmentation strategy decreases the loss of the baseline by 53.4%.

Figure 5: The minimum validation loss for each augmentation strategy when language modeling on GPT-2 small.

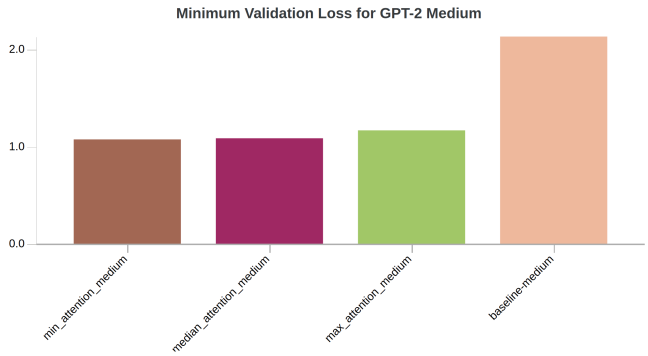

Figure 6: Minimum validation loss for each knowledge augmentation strategy when language modeling on GPT-2 medium. Knowledge augmentation methods continue to improve performance even on larger language models.

modeling fine-tuning tasks. This increased performance is achieved without optimizing the knowledge augmentation process—instead, simply adding a *label:description* relation to knowledge buffer. It stands to reason that the loss could be reduced further by adding more relevant data to the knowledge buffer.

While the difference in minimum validation loss 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.

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 loss for GPT-2 medium only decreases by 5.86% 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 loss by 52.71% 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 [Brown *et al.*, 2020] 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 47.54%.

Higher Order Combinations of Knowledge

Figure 7 shows the validation loss for each combination of knowledge augmentation strategies. As discussed previously, the best first order reduction, min attention, decreases the validation loss by by 53.4%. The best second order combination of min and max attention improves on that by another 12.65%. Finally, the combination of min, median, and max

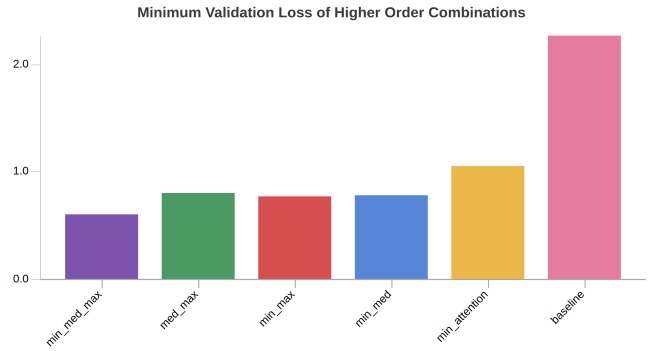

Figure 7: The minimum validation loss for higher-order combinations of knowledge data when language modeling on GPT-2 small. Augmenting with minimum, median, and maximum knowledge entities yields the best results reducing the baseline validation loss by 73.26%.

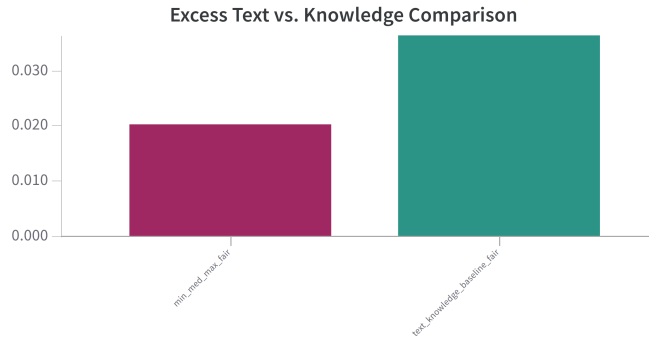

Figure 8: Minimum validation loss when language modeling on GPT-2 small with comparison between filling the knowledge buffer with excess text tokens and filling it with min_med_max knowledge tokens for a dataset where each row has excess tokens.

attention knowledge improves on the second order by an additional 7.21% for a total of 73.26% improvement over the baseline. This demonstrates that a higher quantity of added semantic information consistently results in better generalization.

Excess Text Tokens Versus Knowledge Tokens

Figure 8 compares the validation loss 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 44.28%. This suggests that structured semantic data has a distinct advantage over additional textual context tokens.

BERT

Figure 9 once again shows the minimum validation loss for the primary knowledge augmentation strategies. All of the knowledge augmentation runs outperform the baseline, with the max attention variation resulting in the greatest percentage decrease of the baseline loss (8.01%). While this reduction in loss 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 [Devlin *et al.*, 2019] while GPT-2 excludes it. This means the pre-

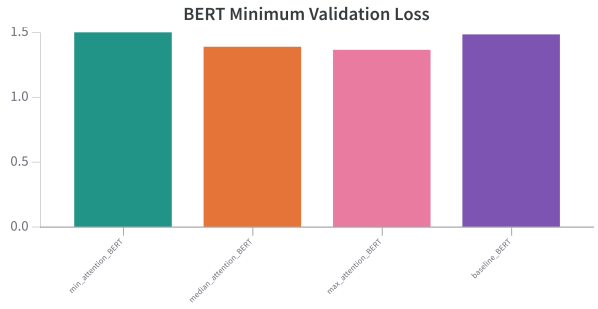

Figure 9: Minimum validation loss when language modeling on BERT. Knowledge augmentation continues to improve results on BERT despite its being pretrained on Wikipedia.

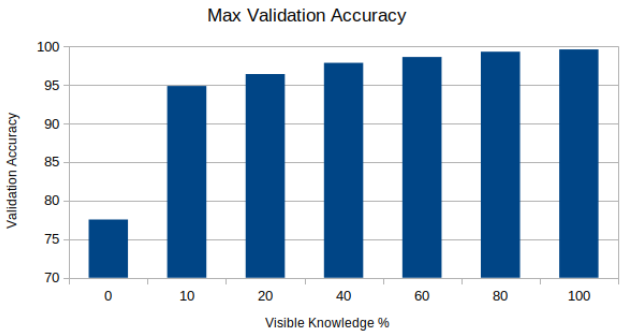

Figure 10: Question answering validation accuracy across a range of knowledge token visibility percentages. The leftmost column with 0% visible knowledge is the baseline dataset that does not contain knowledge augmentation. Even with only 10% of the knowledge tokens visible, the knowledge augmented model improves significantly over this baseline.

trained 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 loss values are not directly comparable. Taking all this into account, these results continue to validate the efficacy of these methods even across model architectures.

4.2 Question Answering

Figure 10 shows the validation accuracy across a range of knowledge token visibility percentages for the question answering task. Even when only 10% of the knowledge tokens are visible, the validation accuracy still increases by 22.3%. This demonstrates that even noisy (a proxy for less relevant data) knowledge data may significantly improve performance on question answering tasks. Of course, due to the fact that on this simple question answering dataset (enough) knowledge tokens (eventually) give away the answer, these results are less reflective of a model's general question answering ability

and instead demonstrate that the model is able to identify the correct answer despite noisy knowledge data. Question answering tasks are a promising next step for these knowledge augmentation methods, even if the knowledge is not perfectly relevant to the question at hand.

5 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 and question answering 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 [Bender *et al.*, 2021]. 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.

References

[Bender *et al.*, 2021] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret 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*, pages 610–623, 2021.

[Bosselut *et al.*, 2019] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, volume 1, pages 4762–4779, 2019.

[Brown *et al.*, 2020] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[Cafarella *et al.*, 2011] Michael J. Cafarella, Alon Halevy, and Jayant Madhavan. Structured data on the web. *Communications of the ACM*, 54(2):72–79, 2011.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina 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*, volume 1, pages 4171–4186, 2019.

[Johnson *et al.*, 2017] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017.

[Jones *et al.*, 2015] Michael N Jones, Jon Willits, Simon Dennis, and Michael Jones. Models of semantic memory. In Jerome R. Busemeyer, Zheng Wang, James T. Townsend, and Ami Eidels, editors, *Oxford Handbook of Mathematical and Computational Psychology*, pages 232–254. Oxford University Press, New York, 2015.

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

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

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

[Mao *et al.*, 2019] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *Proceedings of the International Conference on Learning Representations*, 2019.

[Merity *et al.*, 2017] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations*, 2017.

[Miller *et al.*, 2016] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason

Weston. Key-value memory networks for directly reading documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, 2016.

[Miller, 1998] Eric Miller. An introduction to the resource description framework. *Bulletin of the American Society for Information Science and Technology*, 25(1):15–19, 1998.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[Sap *et al.*, 2019] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035, 2019.

[Speer *et al.*, 2017] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4444–4451, 2017.

[Tulving, 1972] Endel Tulving. *Episodic and Semantic Memory*. Academic Press, Oxford, England, 1972.

[Vrande, 2014] Denny Vrande. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[Wang *et al.*, 2019] 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 *Advances in Neural Information Processing Systems 32*, pages 3261–3275, 2019.

[Wolf *et al.*, 2019] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019.

[Zhang *et al.*, 2021] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. *arXiv preprint arXiv:2108.13161*, 2021.