Relational Memory-Augmented Language Models

Qi Liu[2*], Dani Yogatama[1], and Phil Blunsom[1,2]
[1]DeepMind, United Kingdom, [2]University of Oxford, United Kingdom
{qi.liu,phil.blunsom}@cs.ox.ac.uk
dyogatama@deepmind.com

Abstract

We present a memory-augmented approach to condition an autoregressive language model on a knowledge graph. We represent the graph as a collection of relation triples and retrieve relevant relations for a given context to improve text generation. Experiments on WikiText-103, WMT19, and enwik8 English datasets demonstrate that our approach produces a better language model in terms of perplexity and bits per character. We also show that relational memory improves coherence, is complementary to token-based memory, and enables causal interventions. Our model provides a simple yet effective way to combine an autoregressive language model and a knowledge graph for more coherent and logical generation.

1 Introduction

A core function of language is to communicate propositions (e.g., who did what to whom). As such, language models need to be able to generate this information reliably and coherently. Existing language models (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020) do not have explicit representations for such information and rely on it being implicitly encoded in their parameters (Liu et al., 2019; Petroni et al., 2019; Wang et al., 2020). This encoding mechanism makes it difficult to interpret what the language models know and often leads to generating illogical and contradictory contents. For example, Logan et al. (2019) observe that existing language models rely heavily on word correlation and fall short of logical reasoning. This causes the model to hallucinate—for example, that Barack Obama's wife is Hillary Clinton based on the high co-occurrence of the two entities. In another example, Lake and Murphy (2020) notice that GPT-2 (Radford et al., 2019) states that unicorns have four horns, directly after speaking that unicorns have one horn.

In this work, we explore ways to combine an autoregressive language model with a knowledge graph. We design a memory-augmented architecture that stores relations from a knowledge graph and investigate the effect of conditioning on this relational memory in an autoregressive language model. In contrast to existing token-based memory-augmented language models that store context-target pairs (Khandelwal et al., 2020b; Yogatama et al., 2021), our memory stores relation triples (head entity, relation, tail entity). Relation triples form the basis of knowledge bases, empowering a wide range of applications such as question answering (Yasunaga et al., 2021), machine reading (Yang and Mitchell, 2019), and reasoning (Minervini et al., 2020). From a cognitive science perspective, we can consider the neural language model to be an instance of System 1, which performs fast inference and the symbolic relational memory as a world model to support slow and logical reasoning of System 2 (Kahneman, 2011).[1] We hypothesize that relational memory can improve performance and coherence of an autoregressive language model.

Given an observed context, we first run an entity tagger to identify entities in the context. We then use tf-idf (Ramos et al., 2003) to select salient entities. We retrieve relations (from a knowledge base) for the selected entities and design a gating function that allows the language model to adaptively combine information from extracted relations and observed textual context to predict the next token. Existing knowledge bases such as Freebase and Wikidata can be used as a source of information from which to retrieve relations. However, they are often incomplete and do not contain relations that are suitable for the particular

[*] Work completed during an internship at DeepMind.

[1] This view is also advocated in a parallel work by Nye et al. (2021), which presents a model for story generation and instruction following.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 555–572, 2022. https://doi.org/10.1162/tacl_a_00476
Action Editor: Xavier Carreras. Submission batch: 7/2021; Revision batch: 12/2021; Published 5/2022.

dataset that we want to work with. Instead of using these predefined knowledge bases, we choose to perform open information extraction (OpenIE) on each language modeling dataset to get relations. As a result, our model is able to move beyond simple co-occurrence statistics and generate text that is more grounded on real-world relations observed in a particular corpus.

Our main contributions are as follows:

- We evaluate the model on three English language modeling datasets. We show that our model outperforms a strong transformer-XL baseline (Dai et al., 2019) on both word-level (WikiText-103 and WMT19) and character-level (enwik8) language modeling in terms of perplexity and bits per character respectively (§3.3).
- We conduct comprehensive ablation and design choice studies to understand contributions of different components of our models (§4.1).
- We measure coherence with human evaluation and two automatic metrics (knowledge perplexity and knowledge F_1) and demonstrate that relational memory improves coherence (§4.2).
- We study the relationship between our method and a typical memory-augmented language model that stores word tokens in its memory (Yogatama et al., 2021). We show that relational memory is complementary to token-based memory and combining them improves performance further (§3.3).
- We perform qualitative analysis by examining gate values and retrieved relations. In line with our main motivation, we find that the relational memory is particularly useful for predicting entities. Further, we demonstrate that such explicit propositional representations allow causal interventions and increase interpretability of language models (§4.3).

2 Model

An autoregressive language model defines the probability of a sequence of tokens $p(\boldsymbol{x}) = p(x_1, \ldots, x_T)$. It is common to factorize this joint probability as a product of conditional probabilities with the chain rule (Jelinek, 1980; Bengio et al., 2003):

$$p(x_1, \ldots, x_T) = \prod_{t=1}^{T} p(x_t | x_0, \ldots, x_{t-1}), \quad (1)$$

where x_0 is a special start token.

Our language model is based on transformer-XL (§2.1) which is augmented with a relational memory (§2.2). We discuss them in detail below.

2.1 Transformer-XL

We use transformer-XL (Dai et al., 2019)—which is based on transformer (Vaswani et al., 2017)—to parametrize the conditional probabilities in Eq. 1. Transformer stacks multiple self-attention layers to obtain contextualized representations.

Language modeling datasets usually consist of articles of different lengths. It is impractical to apply transformer to encode long articles, as its computational complexity is quadratic in the sequence length. In practice, each article is usually truncated into fixed-length text segments $\{x_{t-N+1}, \ldots, x_t\}$ of length N to train and evaluate the model. However, this approximation prevents transformer from capturing long-term dependency beyond text segments. Transformer-XL reuses hidden states from previous text segments to extend the context window.

More specifically, denote the hidden state of x_t at layer ℓ as $\mathbf{h}_t^\ell$. Given a text segment $\{x_{t-N+1}, \ldots, x_t\}$ and its extended context $\{x_{t-N-M+1}, \ldots, x_{t-N}\}$ of length M, both the hidden states of the text segment $\{\mathbf{h}_{t-N+1}^\ell, \ldots, \mathbf{h}_t^\ell\}$ and the hidden states of the extended context $\{\mathbf{h}_{t-N-M+1}^\ell, \ldots, \mathbf{h}_{t-N}^\ell\}$ are used. When performing self-attention, each token in the text segment can attend to the preceding tokens in the text segment and all the tokens in the extended context, enabling longer-term dependency compared to a vanilla transformer. Importantly, transformer-XL does not backpropagate through the hidden states of the extended context during training (by adding stop gradient operators to all the hidden states in the extended context).

2.2 Relational Memory

In this section, we first introduce how we obtain relation triples using OpenIE (§2.2.1). We then use tf-idf to score entities in the observed context and retrieve relation triples related to these entities (§2.2.2) to construct relational memory. Finally,

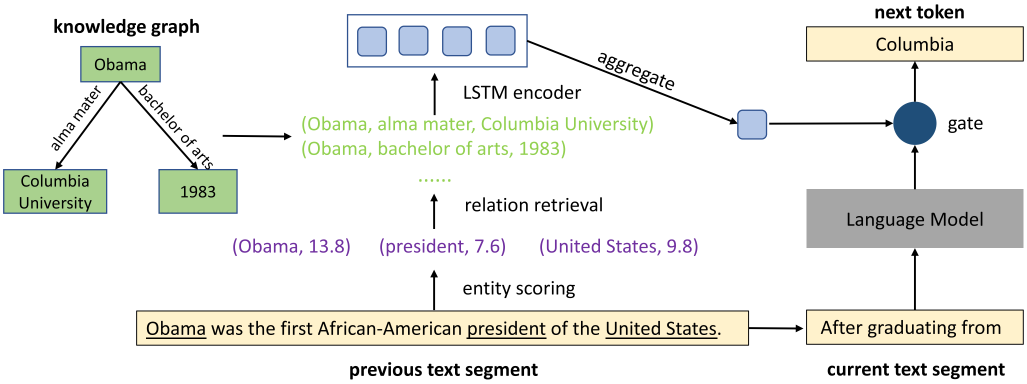

Figure 1: We identify salient entities in the previous text segment and extract relations to build our relational memory. We encode each relation with an LSTM encoder, aggregate the resulting representations into a vector, and use a gate mechanism that allows our language model to adaptively take advantage of relational information for predicting the next token.

we show an integrated architecture that allows transformer-XL to incorporate the relational memory for predicting the next token (§2.2.3). We show our architecture in Figure 1. The pseudocode of training or evaluating with the relational memory is demonstrated in Algorithm 1. In the pseudocode, we use TRAIN($\boldsymbol{x}_c$, $\mathcal{M}$) and EVAL($\boldsymbol{x}_c$, $\mathcal{M}$ to refer to training with the cross entropy loss and evaluating (e.g., calculating perplexity) on the text segment $\boldsymbol{x}_c$ conditioned on the relational memory $\mathcal{M}$, respectively.

2.2.1 Open Information Extraction

A key challenge of utilizing relational information for language modeling is obtaining high-quality relation triples. There are several well-established knowledge bases, such as Freebase (Bollacker et al., 2007) and YAGO (Rebele et al., 2016). However, existing knowledge bases suffer from missing relations and often do not contain relation triples related to observed contexts in a target corpus, even though research on knowledge base completion has resulted in significant advances (Bordes et al., 2013; Trouillon et al., 2016; Zhang et al., 2019).

In this work, we use OpenIE (Angeli et al., 2015; Etzioni et al., 2008) to obtain relation triples. Since OpenIE directly extracts relation triples from each dataset $\mathcal{D}$, it provides a structured way to represent knowledge in $\mathcal{D}$.[2] Specifically, we perform OpenIE on the *training set* of $\mathcal{D}$. Given an entity e, we retrieve a set of relation triples $\mathcal{R}_e = \{r_1, \ldots, r_O\}$, where e is either the head entity or the tail entity in these relation triples. Conceptually, $\mathcal{R}_e$ consists of all the relation triples from the one-hop subgraph centred at the entity e in the knowledge graph constructed from $\mathcal{D}$. Therefore, $\mathcal{R}_e$ can provide ''global'' information about the entity.

Dynamic OpenIE. Dynamic OpenIE takes advantage of the autoregressive nature of language modeling, where text segments are sequentially processed. In addition to extracting relations from the training set of $\mathcal{D}$, we can also extract relations from *previously seen* text segments of our evaluation set. We refer to this extraction mechanism as dynamic OpenIE. After a text segment $\{x_{t-N+1}, \ldots, x_t\}$ has been evaluated, for example, after calculating perplexity on this text segment, we perform OpenIE on it to obtain new relation triples to be added to our knowledge graph. Note that we only perform OpenIE on previously seen text segments and do not use unseen text. We expect that the relation triples extracted from seen text segments are potentially useful for predicting the next tokens. This extraction mechanism will not violate the autoregressive nature of language modeling. Metrics such as perplexity and bits per character are calculated as usual. The idea of using seen text segments during evaluation to improve language modeling is related to dynamic evaluation (Krause et al., 2018, 2019). In dynamic evaluation, the model is adapted based on recent history during evaluation via gradient descent so that it can assign higher probabilities to re-occurring patterns. In contrast to dynamic evaluation, we do not update model parameters and

[2] We provide a comparison of using relations extracted from OpenIE and Freebase in §4.1.

Algorithm 1 Train/Eval w/ Relational Memory

```
procedure TRAIN/EVAL SPLIT($\mathcal{S}$)
    for each article $\mathcal{A}$ in $\mathcal{S}$ do
        Initialise $\mathcal{M}$ to empty
        for each text segment $\boldsymbol{x}_c$ in $\mathcal{A}$ do
            if $\mathcal{S}$ is train set then
                TRAIN($\boldsymbol{x}_c$, $\mathcal{M}$)
            else
                EVAL($\boldsymbol{x}_c$, $\mathcal{M}$)
                Run dynamic OpenIE on $\boldsymbol{x}_c$
            end if
            Perform relation retrieval with $\boldsymbol{x}_c$
            Update $\mathcal{M}$ with retrieved triples
        end for
    end for
end procedure
```

only extract new relations from seen text segments to enrich our corpus-specific knowledge graph.

Mismatch between Training and Evaluation. As shown in Algorithm 1, because we do not use dynamic OpenIE during training due to its additional efficiency overhead (see speed comparison in §4.1), this results in a mismatch between training and evaluation. We extract all the relation triples from the training set of each dataset $\mathcal{D}$ before training on D. As a result, during training we may retrieve relation triples extracted from *unseen* text of the training set when performing relation retrieval (§2.2.2). We do not suffer from this issue during evaluation, as we extract relations from previously seen text of our evaluation set. We believe this mismatch is minor given the superior performance of our model in the experiments.

2.2.2 Relation Retrieval

Given a knowledge graph (represented as a collection of triples), an ideal relational memory consists of a set of triples that are relevant to the observed context. There are many choices to measure the relatedness between the observed context and relation triples in our knowledge graph—for example, based on keyword search or dense retrieval (Karpukhin et al., 2020; Guu et al., 2020; Yogatama et al., 2021).

In this work, we use keyword search because of its simplicity and leave methods based on dense retrieval to future work. Specifically, given the observed context, we perform entity recognition (Ratinov and Roth, 2009; Nadeau and Sekine, 2007) on this context and score the tagged entities with tf-idf (Ramos et al., 2003). The top-K scored entities (K is set to 5 in our experiments) are used to retrieve relations $\{\mathcal{R}_{e_1}, \ldots, \mathcal{R}_{e_K}\}$. These retrieved relations are used to construct the relational memory $\mathcal{M}$. Note that the entities are selected from the observed context, so that unseen text is not utilized. We limit the capacity of $\mathcal{M}$ to P. If the number of newly retrieved triples is larger than P, we randomly drop relations and only select P of them to be inserted into $\mathcal{M}$. Otherwise, the relational memory operates with a first-in-first-out principle. When $\mathcal{M}$ is full, older relations retrieved will be overwritten by newly retrieved relations. The relational memory is re-initialized to empty when an article ends.

As shown in Algorithm 1, since we update $\mathcal{M}$ only after processing an entire text segment, all the tokens in the same text segment will be conditioned on the same relational memory. This approach is more efficient compared to updating $\mathcal{M}$ each time a new entity is encountered and is more amenable for batch training.

2.2.3 Integration with Transformer-XL

We now show how we can integrate relational memory with transformer-XL. We refer to our model as RelationLM.

Relation Triple Encoding. We first discuss how we encode relation triples in the relational memory $\mathcal{M}$. We treat relation triples as text and serialize each relation triple into a sequence, for example, (Barack Obama, president of, United States) is converted into a sequence ''Barack Obama, president of, United States''. This sequential representation can well capture the order of head entities and tail entities and is also adopted by KG-BERT (Yao et al., 2019) and Kepler (Wang et al., 2021b). Because each example in a batch corresponds to P retrieved relations, we obtain $B \cdot P$ relation sequences for each batch, where B and P denote batch size and relational memory length, respectively. In the order of hundreds of relation triples, this prevents us from using large models (e.g., a multi-layer transformer) to encode these sequences due to memory constraints. In our preliminary experiments, we compare LSTM (Hochreiter and Schmidhuber, 1997), GRU (Cho et al., 2014), and a one-layer transformer and find that LSTM performs marginally better. Therefore, for each relation triple r_p, we reuse the

Dataset	# Train	# Valid	# Test	# Articles	# Vocab	# Entities	# Relations	# Relations/Entity
WikiText	103M	0.2M	0.2M	28,595	267,735	980K	8.9M	9.03
WMT19	151M	0.3M	0.3M	169,180	50,259	976K	7.8M	7.97
enwik8	94M	5M	5M	12,350	256	361K	2.4M	6.66

Table 1: Statistics of datasets used in our experiments. For each subset, we show the number of (sub)words for WikiText-103 and WMT19 or the number of characters for enwik8.

transformer-XL word embedding matrix $\mathbf{W}_e$ to map each token in the sequence to its embedding vector. We then run LSTM to encode the sequence and use the hidden representation of the last token as the relation representation $\mathbf{r}_p$.

There are other approaches to encode relation triples, for example, embedding-based (Bordes et al., 2013; Trouillon et al., 2016) and graph-based (Schlichtkrull et al., 2018; Zhang and Chen, 2018) methods. We leave a comparison of these approaches to future work.

Integration. Given a text segment $\boldsymbol{x}_c = \{x_{t-N+1}, \ldots, x_t\}$, after L self-attention layers with transformer-XL, we obtain contextualized representations $\{\mathbf{h}^L_{t-N+1}, \ldots, \mathbf{h}^L_t\}$. At each timestep t, we use its hidden representation $\mathbf{h}^L_t$ as the query vector to attend over the P encoded contents of $\mathcal{M}$, i.e., $\{\mathbf{r}_1, \ldots, \mathbf{r}_P\}$. We use a standard scaled dot-product attention (Vaswani et al., 2017) to aggregate all triples into a single vector:

$$\mathbf{m}_t = \sum_{p=1}^{P} \frac{\exp(\mathbf{h}^L_t \cdot \mathbf{r}_p / \sqrt{d})}{\sum\limits_{j=1}^{P} \exp(\mathbf{h}^L_t \cdot \mathbf{r}_j / \sqrt{d})} \mathbf{r}_p,$$

where d denotes the hidden size of our transformer-XL. Finally, we combine $\mathbf{m}_t$ and transformer-XL representation $\mathbf{h}^L_t$ via a gate:

$$\begin{aligned}
\mathbf{g}_t &= \sigma(\mathbf{W}_g[\mathbf{h}^L_t, \mathbf{m}_t]) \\
\mathbf{z}_t &= \mathbf{g}_t \odot \mathbf{h}^L_t + (1 - \mathbf{g}_t) \odot \mathbf{m}_t \\
p(x_{t+1} \mid \boldsymbol{x}_{\leq t}) &= \mathrm{softmax}(\mathbf{W}_e \mathbf{z}_t),
\end{aligned}$$

where σ is the sigmoid function, $[,]$ denotes concatenation of two vectors, $\odot$ is element-wise multiplication, and $\mathbf{W}_e$ is the embedding matrix shared by both input and output embeddings (Inan et al., 2016). The only new parameters introduced by our method are an LSTM relation encoder and the gate matrix $\mathbf{W}_g$. This gating mechanism allows our model to adaptively take advantage of relational information for predicting the next token.

3 Experiments

Our experiments seek to evaluate the effect of augmenting language models with a relational memory. We introduce datasets used for evaluation (§3.1), discuss implementation details (§3.2), and present our main results (§3.3). We then show ablation studies and further analysis of our model (§4).

3.1 Datasets and OpenIE

We use three English language modeling datasets: WikiText-103 (Merity et al., 2017), WMT19 (Barrault et al., 2019), and enwik8 (Hutter, 2012). Descriptive statistics of these datasets are shown in Table 1. WikiText-103 and WMT19 are (sub) word-level datasets, while enwik8 is a character-level dataset.

WikiText-103 is a knowledge-driven dataset consisting of featured articles from English Wikipedia. WMT19 contains English news from the WMT19 workshop.[3] The news are segmented into months. We use the news from January to October for training, and news in November and December for development and test, respectively. Compared to Wikipedia articles, news contains more dynamic and temporal information, exposing new challenges for utilizing relational information. We reuse the vocabulary of GPT-2 (Radford et al., 2019) with 50,259 tokens to tokenize this dataset. enwik8 contains more than 100M bytes of Wikipedia text. Character-level language modeling has a much smaller vocabulary size than (sub)word-level language modeling.

We perform OpenIE on each dataset. For enwik8, OpenIE is performed after detokenizing its text into words. Statistics of extracted relations are also included in Table 1. Each entity from WikiText-103, WMT19, and enwik8 has 9.03, 7.97, and 6.66 relation triples on average.

[3] `http://www.statmt.org/wmt19/`.

3.2 Implementation Details

All models are implemented with JAX[4] (Bradbury et al., 2018) and Haiku[5] (Hennigan et al., 2020). We set the hidden size to 512 and the number of layers to 16 for all models. In (sub)word-level language modeling, we use adaptive softmax (Grave et al., 2017) for efficiency. We use GELU (Hendrycks and Gimpel, 2016) as our activation function and Adam (Kingma and Ba, 2015) as the optimizer. For training, we use batch size 128 and train the models on 64 16GB TPUs. We apply 4,000 warmup steps, before utilizing cosine annealing to decay the learning rate. Dropout (Srivastava et al., 2014) is applied during training with a rate of 0.25.

We set the lengths of text segment N, extended context M, and the relational memory P to (512, 512, 300), (384, 384, 800), and (768, 1536, 400) for WikiText-103, WMT19, and enwik8, respectively. These are determined by grid searches on development sets.

3.3 Main Results

We compare with a strong transformer-XL baseline trained under the same setting as our model. Our main results are shown in Table 2. We obtain three observations comparing transformer-XL and RelationLM. First, RelationLM consistently outperforms transformer-XL on all three datasets, demonstrating the effectiveness of relational memory. Note that a decrease of 0.01 is considerable on enwik8 with the bits per character metric. Second, relational memory not only improves language modeling on knowledge-driven articles (WikiText-103), but also generalizes to the challenging news domain (WMT19), where information is more dynamic and temporal. Last, the results indicate that relational memory improves both (sub)word-level and character-level language modeling.

Complementarity to Spalm. Spalm (Yogatama et al., 2021) is a state-of-the-art memory-augmented language model. Instead of retrieving relation triples, it retrieves a set of related tokens at each timestep. Specifically, it first stores (context, the next token) pairs from training data. It then uses a pre-trained transformer language model to measure the similarities between the stored contexts and the observed context during training/evaluation. The next tokens of similar contexts are retrieved and are integrated with the observed context via a gating mechanism for generation.

	Model	# Params	Dev	Test
WikiText	Transformer-XL	122M	19.0	19.9
	RelationLM	124M	18.5	19.2
	Spalm	122M	18.1	19.0
	↪ + RelationLM	124M	**17.7**	**18.6**
WMT19	Transformer-XL	114M	21.7	21.5
	RelationLM	116M	21.0	20.7
	Spalm	114M	20.4	20.3
	↪ + RelationLM	116M	**19.8**	**19.6**
enwik8	Transformer-XL	93M	1.05	1.03
	RelationLM	95M	1.04	1.02
	Spalm	93M	1.04	1.02
	↪ + RelationLM	95M	**1.03**	**1.01**

Table 2: We use perplexity (↓) on WikiText-103 and WMT19 and bits per character (↓) on enwik8 for evaluation.

We investigate whether RelationLM is complementary to Spalm. Because Spalm also uses a gating mechanism for integrating the retrieved tokens, we first apply RelationLM to combine transformer-XL output $\mathbf{h}_t^L$ with relational information to obtain $\mathbf{z}_t$ (as shown in §2.2.3), before using Spalm to integrate $\mathbf{z}_t$ with retrieved tokens. The results are shown in Table 2. Spalm outperforms transformer-XL and even performs comparably or better compared to RelationLM on three datasets, demonstrating the effectiveness of retrieving related tokens. However, integrating RelationLM and Spalm can further improve the performance, indicating that these two models are not mutually exclusive. Therefore, retrieving relation triples brings complementary benefits to retrieving tokens.

4 Analysis

In this section, we study several design choices of relational memory, including its knowledge source, input component, capacity, dynamic OpenIE, entity scoring method used, and speed comparison. We then show quantitative and qualitative analysis results to better understand our model.

4.1 Ablations and Design Choice Studies

For the ablation studies, we use the development set of WikiText-103.

[4] https://github.com/google/jax.

[5] https://github.com/deepmind/dm-haiku.

Model	Dev
Transformer-XL	19.0
RelationLM + Freebase	19.0
RelationLM + OpenIE	**18.5**

Table 3: RelationLM with OpenIE or Freebase triples.

Model	Dev
Transformer-XL	19.0
Triple - Relation - Tail	19.0
Triple - Relation	18.7
Triple	**18.5**

Table 4: Ablating relation and/or tail entity from a relation triple.

Source of Relation Triples. We compare relation triples extracted from Freebase or using OpenIE. In the Freebase case, we use the Freebase API[6] to obtain relation triples for each entity. For WikiText-103, there are 10.74 relations per entity on average, which is comparable to OpenIE relations (9.03 relations/entity). The results are shown in Table 3. Although Freebase relations have been observed to improve the performance on smaller datasets (e.g., WikiText-2; Logan et al., 2019) and particular domains (e.g., movies and actors; Ahn et al., 2016), we find that RelationLM with Freebase relations does not improve over transformer-XL on a much larger WikiText-103 dataset. We observe that a large portion of Freebase relations is from infoboxes of Wikipedia pages, which only cover information such as occupation, birth place, and religion. We believe these triples are too general to be useful for most contexts. The result of RelationLM with OpenIE shows the advantages of extracting relations from each dataset compared to using Freebase relations.

Ablating Relation Triples. We ablate relation and/or tail entity from a relation triple (head entity, relation, tail entity) to study the contribution brought by each component. The results are shown in Table 4. We find that ablating both relation and tail entity performs comparably to transformer-XL. As head entities are extracted from the observed context, we believe the extended memory of transformer-XL can offset the effect brought by conditioning on head entities. Ablating relation performs better than transformer-XL. This shows the advantage of introducing tail entities. Using complete relation triples performs the best, demonstrating the effectiveness of this triple representation of knowledge.

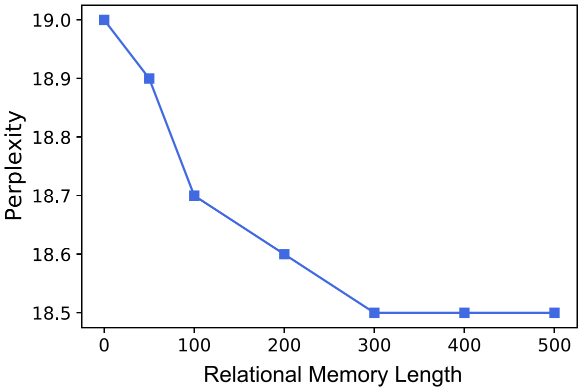

Figure 2: Perplexity on WikiText-103 with different number of relation triples.

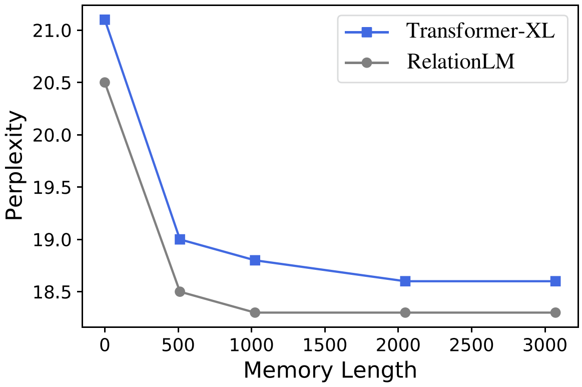

Figure 3: Increasing extended memory length.

Length of Relational Memory. We study how many relation triples need to be stored in the relational memory. As shown in Figure 2, we can see that the perplexity improves with more relation triples. However, the curve becomes flat with more than 300 relation triples.

Length of Transformer-XL Memory. As increasing the length of context window can capture longer dependency, we study whether increasing the length of extended (transformer-XL) memory removes the performance gap between RelationLM and transformer-XL. As shown in Figure 3, the performance of both RelationLM and transformer-XL improves with larger extended memory. However, RelationLM still outperforms transformer-XL even with extended

[6] https://developers.google.com/freebase.

Model	Wiki	WMT	ew8
Transformer-XL	19.0	21.7	1.05
w/o Dynamic OpenIE	18.6	21.4	**1.04**
w/ Dynamic OpenIE	**18.5**	**21.0**	**1.04**

Table 5: Perplexity with and without dynamic OpenIE.

Model	Dev
Random	19.1
Frequency	18.7
tf-idf	**18.5**

Table 6: Perplexity with different entity scoring methods.

Model	Train	Eval
Transformer-XL	**0.51**	**0.31**
RelationLM	0.76	0.65

Table 7: The unit is second/step. We use batch size 128 and 1 per step for training and evaluation, respectively.

memory length 3072. We conclude that relational memory brings complementary benefits to simply expanding extended memory, since it provides global information about entities on each dataset.

Dynamic OpenIE. All our main results use dynamic OpenIE. We show results without dynamic OpenIE in Table 5. We include the results on three datasets for a comparison. We can see that RelationLM with dynamic OpenIE performs comparably to RelationLM without dynamic OpenIE on WikiText-103 and enwik8, while larger improvements are obtained on WMT19. This indicates that dynamic OpenIE is more helpful for the news domain, which is more dynamic and temporal compared to knowledge-driven articles.

Entity Scoring. We study different entity scoring mechanisms for relation retrieval. We consider random selection (where entities extracted from the observed context are randomly selected), frequency-based scoring, and tf-idf scoring. As shown in Table 6, tf-idf performs the best.

Speed Comparison. The wall clock time for both training and evaluation is shown in Table 7. RelationLM is 1.5 and 2.1 times slower during training and evaluation, respectively. Evaluation slows down some more due to dynamic OpenIE as shown in Algorithm 1.

Dataset	Subset	# Entity	# Non-Entity
WikiText	Dev	61.6K	155.9K
	Test	65.8K	179.7K
WMT	Dev	84.9K	262.2K
	Test	81.0K	256.6K
enwik8	Dev	1.7M	3.3M
	Test	1.7M	3.3M

Table 8: Statistics of entity and non-entity tokens.

	Metric	Model	Dev	Test
Knowledge PPX	WikiText	Transformer-XL	47.3	52.3
		RelationLM	**45.6**	**50.9**
	WMT	Transformer-XL	77.2	77.0
		RelationLM	**73.2**	**73.1**
	enwik8	Transformer-XL	2.25	2.21
		RelationLM	**2.22**	**2.19**
Non-entity PPX	WikiText	Transformer-XL	13.3	13.8
		RelationLM	**13.0**	**13.4**
	WMT	Transformer-XL	14.4	14.4
		RelationLM	**14.2**	**14.3**
	enwik8	Transformer-XL	**1.98**	**1.95**
		RelationLM	**1.98**	**1.95**

Table 9: Knowledge perplexity (↓) and non-entity perplexity (↓).

4.2 Does Relational Memory Improve Coherence?

For evaluating coherence, we use two automatic metrics—knowledge perplexity and knowledge F_1—to investigate whether the models can faithfully use entities. We further perform a human evaluation to study whether language models can generate coherent and knowledgeable sequences. We believe the human evaluation is a reliable way of evaluating coherence. This claim is advocated in Barzilay and Lapata (2005). We note that question answering is also often used to evaluate coherence (Guu et al., 2020; Lin et al., 2021). We leave this to future work.

Knowledge Perplexity. While vanilla perplexity considers all words in an evaluation set, knowledge perplexity only considers entities for calculating perplexity. We use it to evaluate whether the model can assign higher probabilities for the correct entities under different contexts. Table 8 shows the numbers of entity words and non-entity words in our corpora. We show the results in Table 9. We observe that the gap between RelationLM and transformer-XL is larger on

Metric	Model	Dev	Test
WikiText	Transformer-XL	9.9	9.4
	RelationLM	**11.4**	**11.2**
WMT	Transformer-XL	11.4	11.0
	RelationLM	**12.6**	**12.3**
enwik8	Transformer-XL	16.0	18.9
	RelationLM	**16.6**	**19.4**

Table 10: Knowledge F_1 (↑).

knowledge perplexity. RelationLM only performs comparably or slightly better compared to transformer-XL on non-entity perplexity. This shows that relational memory is helpful for predicting entity words. Note that knowledge perplexity tends to be much higher than perplexity on non-entity words, indicating the difficulty of predicting entity words. This collection of results indicates that relational memory helps the model use entities coherently and consistently under different contexts.

Knowledge F_1. We use knowledge F_1 to explore whether our model generates tokens that are grounded to its contexts. Given a context as input, we sequentially generate 32 words (or 128 characters) for word-(character-)level language modeling by sampling from the distribution of the next word (character). To reduce variance, we generate 100 continuations for each context. We then perform entity recognition for both the generated sequences and their corresponding ground-truth sequences and calculate an F_1 score based on these two sets of entities. For example, given the context ‘‘...Ayola was nominated and shortlisted for the ‘Female Performance in TV’ award’’, we compare the generated text and the ground truth ‘‘in the 2006 Screen Nation Awards, for her role as Kyla Tyson in Holby City...’’ to calculate F_1. The results are shown in Table 10. We notice that RelationLM performs better compared to transformer-XL. We conclude that models with relational memory can generate more coherent and logical text.

Human Evaluation. We conduct a human evaluation to study whether language models can generate coherent and knowledgeable sequences. We take 1,000 contexts from the test set of WikiText-103. We show the contexts, ground-truth sequences, and continuations generated by RelationLM and transformer-XL to five annotators. We use greedy decoding for both models. We shuffle the order of the continuations generated by RelationLM and transformer-XL so that the annotators are unaware of the sources of sequences. We then pose the following questions to the annotators:

Model	Coherent	Knowledgeable
Transformer-XL	388	416
RelationLM	**612**	**584**

Table 11: We show the number of contexts in which a continuation from a particular model is chosen by human evaluators for each evaluation criterion. Recall that the total number of contexts used for human evaluation is 1,000. Because we have five annotators, we use majority voting to decide the favored model for each continuation. We use the Kappa statistic to measure inter-annotator agreement. The statistic is 0.64, which shows substantial agreement among the annotators.

1. **Coherent.** Given the context and its ground-truth continuation for reference, which generated sequence is more logical and coherent?
2. **Knowledgeable.** Given the context and its ground-truth continuation, which generated sequence provides more insights and is more knowledgeable?

We show the results in Table 11. We find that RelationLM outperforms transformer-XL in the human evaluation. These results are consistent with the two automatic metrics, knowledge perplexity and knowledge F_1. This corroborates our claim that relational memory improves coherence in language modeling.

4.3 Qualitative Analysis

Gate Values. As we use a gating function to integrate transformer-XL with relational information, we study gate values in this section. The histogram of gate values is shown in Figure 5. We notice that the histogram concentrates around 0.9. This is expected because non-entity words, which account for a large portion of text (according to Table 8), benefit less from the relational memory and mainly rely on the observed context for prediction as shown in §4.2. We further calculate the average gate values for entity words and non-entity words. The average gate value for entity

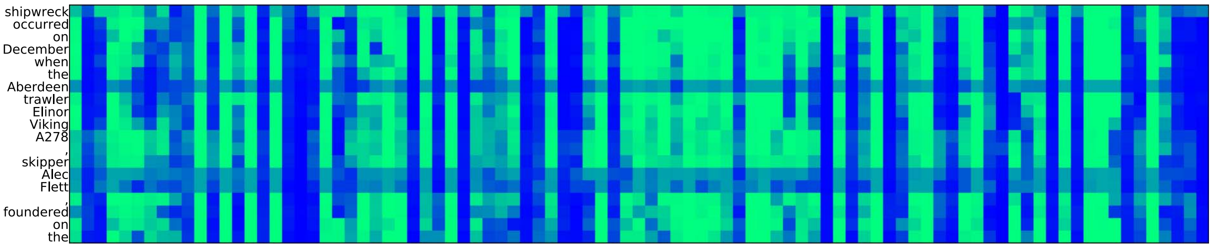

Figure 4: Heatmap of gate values.

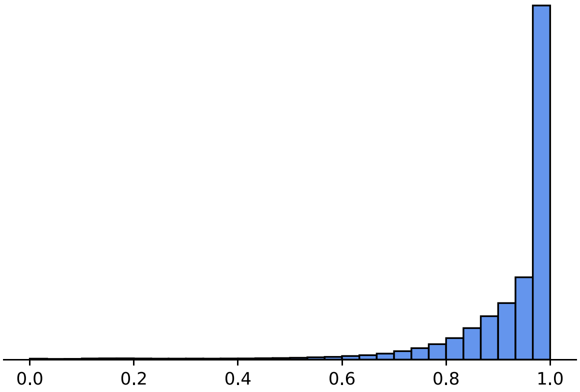

Figure 5: Histogram of gate values $\mathbf{g}_t$.

words is 0.87, while the average value is 0.92 for non-entity words. This confirms that entity words rely more on relational information for prediction compared to non-entity words. We also plot the heatmap of gate values and a cherry-picked example is shown in Figure 4. Note that we randomly select 100 dimensions from 512 dimensions for readability. We notice that the entities, Aberdeen and Alec Flett, use more relational information than other positions (as shown by the horizontal blue lines). These results demonstrate that RELATIONLM can adaptively incorporate relational information for prediction.

Example. We show three cherry-picked examples in Table 12. We take the first for illustration, which shows a text segment from the article, Joe Biden 2008 presidential campaign[7] and some retrieved relations. We find that the first two relations, (Joe Biden, senior Senator, Delaware) and (Joe Biden presidential campaign, began, January 7 2007), are extracted from previous text segments, while (Joe Biden, was nominated, vice president) and (Biden, withdrew nomination, 1987) are extracted from the other articles, Joe Biden[8] and Joe Biden 1988 presidential campaign,[9] respectively. We notice that the relation (Joe Biden, was nominated, vice president) is highly predictive of the sequence, ‘‘Biden was selected to be Democratic presidential nominee Barack Obama’s vice presidential running mate’’. From the observed context, the model also identifies a closely related entity, Barack Obama, and retrieves the relation (Barack Obama, president of, United States). Therefore, we conclude that the relational memory can give a global picture of

Seven months after conclusion of his campaign, Biden was selected to be Democratic presidential nominee Barack Obama's vice presidential running mate. The pair won in the general election, and were sworn in on January 20, 2009 ... (Joe Biden, senior Senator, Delaware) (Joe Biden presidential campaign, began, January 7 2007) (Joe Biden, was nominated, vice president) (Biden, withdrew nomination, 1987) (Barack Obama, president of, United States)
From 7 February 2006 to 9 December 2008, Ayola starred in BBC medical drama Holby City as nurse Kyla Tyson. She had previously appeared in Holby City 's sister show Casualty … (Holby City, is, BBC medical drama) (Rakie Ayola, played the role, Kyla Tyson)
Independiente became Arjona's fourth number one album on the Billboard Top Latin Albums where it debuted for the week ending 22 October 2011. For thirteen non-consecutive weeks it topped the Latin Pop Albums chart ... (Independiente, number one on, Top Latin Albums chart) (Independiente, became number one on, 22 October 2011)

Table 12: Three examples of text segment and retrieved relations (based on previous text segments).

[7] https://en.wikipedia.org/wiki/Joe_Biden_2008_presidential_campaign.

[8] https://en.wikipedia.org/wiki/Joe_Biden.

[9] https://en.wikipedia.org/wiki/Joe_Biden_1988_presidential_campaign.

related entities and provide relevant information for language modeling.

Causal Intervention. We use causal intervention to study whether changing the contents in the relational memory will affect language model prediction. Given the relation (Obama, born in, Hawaii) along with other relations about Barack Obama, we let the model complete the sequence, ‘‘Obama was born in’’. RelationLM outputs ‘‘Obama was born in and raised in Hawaii.’’ with greedy decoding. However, after modifying the relation to (Obama, born in, Kenya), we obtain ‘‘Obama was born in Kenya and was the first African-American president.’’ We further change to (Obama, born in, Paris) and the model outputs ‘‘Obama was born in Paris, France.’’ This indicates that RelationLM can take advantage of relation triples for making prediction. While we can also use prompts as intervention for vanilla language models, it remains challenging about selecting the appropriate prompts in different applications (Liu et al., 2021a).

5 Related Work

Knowledge-enhanced Architectures. Injecting symbolic knowledge to machine learning models is widely adopted to improve the performance of natural language understanding (Annervaz et al., 2018; Ostendorff et al., 2019), question answering (Zhang et al., 2018; Huang et al., 2019; Hixon et al., 2015), dialogue systems (Zhang et al., 2018; Moon et al., 2019; Guo et al., 2018; Liu et al., 2021b), and recommendation systems (Zhang et al., 2016; Wang et al., 2018a, 2019). Different from these models, we focus on using symbolic knowledge for language modeling. Existing language models are prone to generating illogical and contradictory contents. We believe that connecting language modeling and knowledge graphs is a promising direction to overcome the problem. Next we review previous knowledge-enhanced language models.

Knowledge-enhanced Language Models. Our model is closely related to previous work on grounding autoregressive language models with knowledge graphs (Ahn et al., 2016; Logan et al., 2019; Hayashi et al., 2020; Wang et al., 2021a). However, these models rely on complex and adhoc preprocessing or rules to link text with knowledge bases (e.g., Freebase and Wikidata). As a result, previous work is more aligned with conditional language modeling, for example, graph-to-text generation $p(\mathbf{x}|\mathcal{G})$ in Wang et al. (2021a), which contrasts with unconditional language modeling $p(\mathbf{x})$ considered in this work. As the graph $\mathcal{G}$ is constructed with the *unseen* text $\mathbf{x}$, predicting $\mathbf{x}$ given $\mathcal{G}$ is easier due to this information leakage for Wang et al. (2021a). Also in Hayashi et al. (2020), topic entities are required for language modeling, which may not be available in most datasets, for example, the news domain. We do not compare with these previous models due to the different settings. In contrast, we adopt OpenIE relations and use a tf-idf search to retrieve relation triples for connecting language models and knowledge graphs. In the experiments, we demonstrate the effectiveness of our approach on three datasets, WikiText-103, WMT19, and enwik8.

There are language models incorporating entity information, such as entity coreference annotations (Ji et al., 2017; Clark et al., 2018), surface forms of entities (Kiddon et al., 2016; Yang et al., 2017; Cao et al., 2021), entity types (Parvez et al., 2018; Wang et al., 2018b), and entity descriptions (Bahdanau et al., 2017). Different from these models, we augment language models with a relational memory consisting of relation triples. We demonstrate the effectiveness of using relation triples by ablating tail entities and relations in §4.1.

Knowledge-enhanced Pretraining. Using knowledge information for pretraining language models (Peters et al., 2019; Sun et al., 2019; Liu et al., 2020; Guu et al., 2020; Wang et al., 2021b; Agarwal et al., 2021; Verga et al., 2021) has recently grown in popularity and has achieved substantial improvements on knowledge-driven tasks such as question answering and named entity recognition. Instead of using knowledge information for improving downstream knowledge-driven tasks, we focus on using knowledge information for improving the generation capability of the language model itself.

Retrieval-augmented Models. Retrieval-augmented models are now widely adopted in open-domain question answering (Chen et al., 2017; Lewis et al., 2020; de Masson d’Autume et al., 2019; Izacard and Grave, 2021), dialogue (Dinan et al., 2019; Fan et al., 2021; Thulke et al., 2021), and machine translation (Bapna and Firat, 2019;

Khandelwal et al., 2020a). We focus on retrieval augmentation for language modeling (Merity et al., 2017; Grave et al., 2016; Khandelwal et al., 2020b; Yogatama et al., 2021). These algorithms are specifically tailored for language modeling, where related tokens are retrieved to help predict the next token. In this work, we move beyond token augmentation and show the benefits of retrieving relation triples. We also demonstrate that our model is complementary to a token augmentation model, SPALM (Yogatama et al., 2021), in the experiments.

6 Conclusion

We presented RELATIONLM, a language model that is augmented with relational memory. We showed how to obtain relevant knowledge graphs for a given corpus and how to combine them with a state-of-the-art language model such as transformer-XL. We demonstrated that our model improves performance and coherence on WikiText-103, WMT19, and enwik8. We also performed a comprehensive analysis to better understand how our model works. Our model provides a way to combine an autoregressive language model with general knowledge graphs.

Acknowledgments

We would like to thank our action editor (Xavier Carreras) and three anonymous reviewers for their insightful comments. We also thank Angeliki Lazaridou, Cyprien de Masson d'Autume, Lingpeng Kong, Laura Rimell, Aida Nematzadeh, and the DeepMind language team for their helpful discussions.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565. https://doi.org/10.18653/v1/2021.naacl-main.278

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *arXiv preprint arXiv: 1608.00318*.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 344–354. The Association for Computer Linguistics. https://doi.org/10.3115/v1/P15-1034

K. M. Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. 2018. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 313–322. Association for Computational Linguistics.

Dzmitry Bahdanau, Tom Bosc, Stanislaw Jastrzebski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. 2017. Learning to compute word embeddings on the fly. *CoRR*, abs/1706.00286.

Ankur Bapna and Orhan Firat. 2019. Nonparametric adaptation for neural machine translation. 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)*, pages 1921–1931. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-5301

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25-30 June 2005, University of Michigan, USA*, pages 141–148. The Association for Computer Linguistics. https://doi.org/10.3115/1219840.1219858

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Kurt D. Bollacker, Robert P. Cook, and Patrick Tufts. 2007. Freebase: A shared database of structured general human knowledge. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 1962–1963. AAAI Press.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. 2018. JAX: Composable transformations of Python+NumPy programs.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1171

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259. https://doi.org/10.3115/v1/W14-4012

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2250–2260. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1204

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *7th Interna-*

tional Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74. https://doi.org/10.1145/1409360.1409378

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2021. Augmenting transformers with KNN-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99. https://doi.org/10.1162/tacl_a_00356

Édouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. 2017. Efficient softmax approximation for GPUs. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1302–1310. PMLR.

Edouard Grave, Armand Joulin, and Nicolas Usunier. 2016. Improving neural language models with a continuous cache. *CoRR*, abs/1612.04426.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2946–2955.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pretraining. *CoRR*, abs/2002.08909.

Hiroaki Hayashi, Zecong Hu, Chenyan Xiong, and Graham Neubig. 2020. Latent relation language models. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7911–7918. AAAI Press. https://doi.org/10.1609/aaai.v34i05.6298

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. 2020. Haiku: Sonnet for JAX.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 851–861. The Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735, PubMed: 9377276

Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, pages 105–113. ACM. https://doi.org/10.1145/3289600.3290956

Marcus Hutter. 2012. The human knowledge compression contest. *http://prize. hutter1. net*, 6.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *CoRR*, abs/1611.01462.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 – 23, 2021*, pages 874–880. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.74

Frederick Jelinek. 1980. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of Workshop on Pattern Recognition in Practice, 1980*.

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A. Smith. 2017. Dynamic entity representations in neural language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1830–1839. Association for Computational Linguistics.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020a. Nearest neighbor machine translation. *CoRR*, abs/2010.00710.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020b. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 329–339. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1032

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

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2018. Dynamic evaluation of neural sequence models. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2771–2780. PMLR.

Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2019. Dynamic evaluation of transformer language models. *CoRR*, abs/1904.08378.

Brenden M. Lake and Gregory L. Murphy. 2020. Word meaning in minds and machines. *CoRR*, abs/2008.01766.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *CoRR*, abs/2109.07958.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Qi Liu, Lei Yu, Laura Rimell, and Phil Blunsom, 2021b. Pretraining the noisy channel model for task-oriented dialogue. *Transactions of the Association for Computational Linguistics*, 9:657–674. https://doi.org/10.1162/tacl_a_00390

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling language representation with knowledge graph. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational*

Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 2901–2908. AAAI Press. https://doi.org/10.1609/aaai.v34i03.5681

Robert L. Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5962–5971. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1598

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Pasquale Minervini, Matko Bosnjak, Tim Rocktäschel, Sebastian Riedel, and Edward Grefenstette. 2020. Differentiable reasoning on large knowledge bases and natural language. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5182–5190. AAAI Press. https://doi.org/10.1609/aaai.v34i04.5962

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26. https://doi.org/10.1075/li.30.1.03nad

Maxwell Nye, Michael Henry Tessler, Joshua B. Tenenbaum, and Brenden M. Lake. 2021. Improving coherence and consistency in neural sequence models with dual-system, neurosymbolic reasoning. *CoRR*, abs/2107.02794.

Malte Ostendorff, Peter Bourgonje, Maria Berger, Julián Moreno Schneider, Georg Rehm, and Bela Gipp. 2019. Enriching BERT with knowledge graph embeddings for document classification. In *Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019*.

Md. Rizwan Parvez, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2018. Building language models for text with named entities. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers*, pages 2373–2383. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1221

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. 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*, pages 43–54. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1005

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? 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*, pages 2463–2473. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1250

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

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 242, pages 29–48. Citeseer.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155. https://doi.org/10.3115/1596374.1596399

Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, volume 9982 of *Lecture Notes in Computer Science*, pages 177–185. https://doi.org/10.1007/978-3-319-46547-0_19

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer. https://doi.org/10.1007/978-3-319-93417-4_38

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *CoRR*, abs/1904.09223.

David Thulke, Nico Daheim, Christian Dugast, and Hermann Ney. 2021. Efficient retrieval augmented generation from unstructured knowledge for task-oriented dialog. *arXiv preprint arXiv:2102.04643*.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2021. Adaptable and interpretable neural memoryover symbolic knowledge. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3678–3691. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.288

Chenguang Wang, Xiao Liu, and Dawn Song. 2020. Language models are open knowledge graphs. *CoRR*, abs/2010.11967.

Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018a. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1835–1844. ACM. https://doi.org/10.1145/3178876.3186175

Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 2000–2010. ACM.

Luyu Wang, Yujia Li, Ozlem Aslan, and Oriol Vinyals. 2021a. WikiGraphs: A Wikipedia text - knowledge graph paired dataset. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language*

Processing (TextGraphs-15), pages 67–82, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.textgraphs-1.7

Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018b. Describing a knowledge base. In *Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018*, pages 10–21. Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-6502

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021b. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194. https://doi.org/10.1162/tacl_a_00360

Bishan Yang and Tom M. Mitchell. 2019. Leveraging knowledge bases in lstms for improving machine reading. *CoRR*, abs/1902.09091.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017. Reference-aware language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1850–1859. Association for Computational Linguistics. https://doi.org/10.18653/v1/D17-1197

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. KG-BERT: BERT for knowledge graph completion. *CoRR*, abs/1909.03193.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. *CoRR*, abs/2104.06378. https://doi.org/10.18653/v1/2021.naacl-main.45

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373. https://doi.org/10.1162/tacl_a_00371

Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 353–362. ACM. https://doi.org/10.1145/2939672.2939673

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5171–5181.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 2731–2741.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018. Variational reasoning for question answering with knowledge graph. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 6069–6076. AAAI Press.

Xiangyang Zhou, Lu Li, Daxiang Dong, Yi Liu, Ying Chen, Wayne Xin Zhao, Dianhai Yu, and Hua Wu. 2018. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1118–1127. Association for Computational Linguistics.