Memory-augmented generative adversarial transformers

Raaijmakers, S.; Bakker, R.; Cremers, S.; Kleijn, R.E. de

MEMORY-AUGMENTED GENERATIVE ADVERSARIAL TRANSFORMERS

Stephan Raaijmakers[1,2], **Roos Bakker**[2], **Anita Cremers**[3], **Roy de Kleijn**[4], **Tom Kouwenhoven**[5], **and Tessa Verhoef**[5]

[1]Leiden University Centre for Linguistics (LUCL)
[2]TNO, The Netherlands
[3]University of Applied Sciences, Utrecht
[4]Institute of Psychology, Leiden University
[5]Leiden Institute of Advanced Computer Science (LIACS)

ABSTRACT

Conversational AI systems that rely on Large Language Models, like Transformers, have difficulty interweaving external data (like facts) with the language they generate. Vanilla Transformer architectures are not designed for answering factual questions with high accuracy. This paper investigates a possible route for addressing this problem. We propose to extend the standard Transformer architecture with an additional memory bank holding extra information (such as facts drawn from a knowledge base), and an extra attention layer for addressing this memory. We add this augmented memory to a Generative Adversarial Network-inspired Transformer architecture. This setup allows for implementing arbitrary felicity conditions on the generated language of the Transformer. We first demonstrate how this machinery can be deployed for handling factual questions in goal-oriented dialogues. Secondly, we demonstrate that our approach can be useful for applications like *style adaptation* as well: the adaptation of utterances according to certain stylistic (external) constraints, like social properties of human interlocutors in dialogues.

1 Introduction

Transformers ([1]) are capable of producing natural, well-formed language with high degrees of fluency. They are responsible for the latest commercial and open source large language models (LLMs) like BLOOM ([2]) and the GPT-models powering ChatGPT. Transformer architectures produce such language models either through a full encoder-decoder combination (e.g. T5, [3]), just the encoder (BERT, [4]) or the decoder (e.g. the GPT models, [5]). In general, probabilistic language models decompose the probability of word sequences $w_1, \ldots, w_n$ into a product of conditional probabilities, estimated from raw textual data:

$$P(w_t \mid w_1 \ldots w_{t-1}) = \prod_{t=1}^{n} P(w_t \mid w_{<t}) \tag{1}$$

The size of neural network-driven Large Language Models is determined by the amount of training data, the amount of model parameters, and the amount of GPU flops for training the models. LLMs treat words as points in a high-dimensional vector space (*encoding*) and complete (encoded) word sequences with generated, subsequent words (*decoding*). Such probabilistic neural models can be summarized succinctly for a left-to-right situation (generating words on the basis of left context) as

$$P(w_t \mid w_1 \ldots w_{t-1}) = \exp\left(Emb_{w_t}^{\top} f_\theta(w_1 \ldots w_{t-1})\right) \tag{2}$$

with $Emb_{w_t}^{\top}$ the embedding of token w_t, and $f_\theta(\cdot)$ a parameterized function that, based on the learned parameter settings θ of the neural network architecture used, encodes the preceding word sequence $w_1 \ldots w_{t-1}$ into a combined

vector representation. A more general form is

$$P(w_t \mid w_1 \dots w_{t-1}) = \prod_{t=1}^{n} P_\theta(w_t \mid w_{<t}) \tag{3}$$

LLMs minimize during training the negative log-likelihood over a training corpus (a collection of documents D, $D[d]$ being the d-th document in D, and $|D[d]|$ its document length in terms of tokens):

$$\mathcal{L}(D) = -\sum_{d=1}^{|D|} \sum_{t=1}^{|D[d]|} \log p_\theta(w_t^{D[d]} \mid w_{<t}^{D[d]}). \tag{4}$$

In conversational AI, and specifically in goal-oriented dialogues, additional conditions on generated utterances typically apply. In such dialogues, conversational agents need to respond to users with appropriate utterances that are not only linguistically correct but also provide correct information, and are stylistically acceptable in the current dialogue context (e.g. [6]). Unfortunately, vanilla Transformers are not equipped for such tasks. Their purpose is to generate language from language, conditioned on attention patterns (*self-attention* and *intra-attention*). A Transformer typically learns to convert input sequences into output sequences, encoding their input with attention values, and generating output from the encoded representations. Standard Transformers cannot readily produce factual information in response to user input, e.g. for answering questions ([7]): all knowledge they contain is derived from their underlying language model. In addition, they appear to be prone to *lexical hallucinations* ([8]), generating random output words based on small, unexpected perturbations in their input data. These innate deficiencies make them unsuitable for accurate question answering. This paper addresses the question of how to improve this situation for conversational systems based on Transformer-produced LLMs ("conversational LLMs"). Augmenting the memory of Transformers with external (e.g. factual) data is a well-known tactic (see Section 3 below). We argue for adding another control facility to such memory-augmented Transformers: a generative adversarial training tactic that allows for exerting additional conditions (such as factual compliance) on the utterances generated by Transformers. We will first discuss such an adversarial control mechanism, and then present the combination of that mechanism with a memory augmented Transformer architecture.

2 Generative Adversarial Transformers

A generative adversarial network (GAN, [9]) is an implementation of a *zero-sum* game, where two parties interact with each other: a *discriminator* and a *generator*. In zero-sum games, the loss of one party benefits the other party. Zero-sum games are commonly expressed in terms of a *value function* $V(D, G)$ which, for a discriminator D and a generator G, expresses the quantities to be minimized and maximized during training. In particular, the discriminator D seeks to *maximize* the objective of discriminating between a reference data distribution and the 'fake' data generated by the generator, which initially will be noisy, but will gradually start approximating the true data distribution. The generator attempts to *minimize* the dispersion of its own data distribution and the reference data. In this context, zero-sum implies that the amount to which the discriminator discerns successfully between fake and real data points translates to a penalty for the generator: the better the discriminator performance (i.e. the lower its error when discriminating between true and faked data), the more severe the penalty for the generator: it must work harder to mislead the discriminator, and thereby increase the discriminator error. Conditional GANs are a specific type of networks that are conditioned on desired output labels ([10]). We propose in this paper a type of conditional Generative Adversarial Transformer (GAT). Both generator and discriminator are implemented as Transformers, trained through the adversarial zero-sum game of GANs. The generator can be conditioned on additional loss functions $\mathcal{L}_\mathcal{G}$ and external data $\mathcal{M}$. The GAT has the following constrained value function V:

$$\min_D \max_G V(D, G) = \mathbb{E}_{x \sim p(x} \left[\log(D(x, m_x)\right] + \mathbb{E}_{z \sim p_z(z)} \left[\log(1 - D(G(z, m_z, \lambda^*))\right] \tag{5}$$

with p_x the true data distribution, p_z the noisy distribution from the generator, and

$$\lambda^* = \lambda_1^n \in \mathcal{L}_G = \lambda_1(\cdot) + \ldots + \lambda_n(\cdot). \tag{6}$$

with the default that λ^* is a zero function $f(\cdot) = 0$.

D and G are parameterized functions (m_x the external data associated with data point x):

$$\begin{gathered} D(x, m_x; \theta_D) \\ \\ G(x, m_x, \lambda^*; \theta_G) \end{gathered} \tag{7}$$

The expected values are defined as:

$$\mathbb{E}_{x\sim p(x)}\left[\log(D(x, m_x)\right] = \int \log(D(x, m_x))p(x)dx \tag{8}$$

and

$$\mathbb{E}_{z\sim p_z(z)}\left[\log(1 - D(G(z, m_z, \lambda^*)), m_z)\right] = \int \log(1 - D(G(z, m_z, \lambda^*), m_z))p(z)dz \tag{9}$$

This expresses that, for a given data point x, both the generator G and the discriminator D take the corresponding external data m_x into account, and that the generator may be subjected to additive, generator-specific loss functions. According to this design, a GAT can be conditioned in three manners: (1) through the external data that accompanies the normal stream of input data; (2) through additional felicity conditions imposed on the generator; (3) by combining (1) and (2). This opens up possibilities for checking factual adherence of these models. Practically, we implement such conditions as loss functions on the generator, and we add their results through summation to the default Transformer loss function scores (sparse categorical cross entropy, measuring overlap between predicted utterance and ground truth utterance). In terms of language modeling, this means we redefine negative log-likelihood as follows:

$$\mathcal{L}(D) = -\sum_{d=1}^{|D|}\sum_{t=1}^{|D[d]|} \log p_\theta(w_t^{D[d]} \mid w_{<t}^{D[d]}, m_d, \mathcal{L}_G). \tag{10}$$

with m_d the external data linked to "document" d (a training data item).

Next, we need to make the extra, external data available to the conditional Generative Adversarial Transformer through *memory-augmentation*.

2.1 Memory-augmented Conditional Generative Adversarial Transformers

We equip the GAT generator and discriminator with an additional attention layer addressing a separate stream of external data, creating a *memory-augmented* Transformer. The external data is aligned with the original input data on an item-by-item basis, and can be optionally empty or even address data that is part of the original input, as a means of emphasis. Specifically, in both encoder and decoder components of the Transformer, we add an additional multihead attention layer for external data. Every head in this layer computes attention using the standard Transformer *Query/Key/Value* mechanism ([1]). For handling external data, we let *Query*=inputs, *Key*=external data and *Value*=external data. Attention for both original input data and external data is summed in both the encoder and decoder blocks of the Transformer, similar to the approach of [11]. Figure 1 illustrates the encoder and decoder blocks of the memory-augmented Transformer. The controlling Generative Adversarial Transformer is illustrated in Figure 2

As discussed, the generators in GANs generate initially noisy data that becomes scrutinized or rewarded by the discriminators. The worse the discriminator discerns the generated data from real data, the higher the reward for the generator. In the case of text-based Transformers, the output of the generators, per training epoch, is treated as noisy data, all of which is labeled as 'fake'. The generator can be conditioned with arbitrary loss functions, measuring the quality of the generated answers given the corresponding questions. These loss functions are supplementary to the normal loss function imposed on the Transformer, which measures the dispersion of the generated answers with respect to the ground truth training data using sparse categorical entropy over the integer-encoded words in inputs and predictions. In subsection 4.2, we outline a sample loss function used in our experiments.

3 Related work

There are different perspectives on conditioning conversational Large Language Models on external data. Our work can be related to - and contrasted with- five major types of approaches.

Black-box LLMs and fine-tuning One obvious bottleneck for fine-tuning and memory augmentation approaches to conditional language generation is the necessity to invest in (re)training LLMs. Recent approaches like [12] attempt to circumvent this restriction by connecting pre-trained large language models ("black-box LLMs") to external data sources directly. This differs from our approach, where we effectively train LLMs from the ground up. Alternatively, many approaches focus on *fine-tuning* pre-trained Large Language Models for conditional conversation generation. An early example is [11], who propose to combine conversational and non-conversational, factual data with Memory Networks through multi-task learning, where conversational and factual text generations tasks are learned in conjunction.

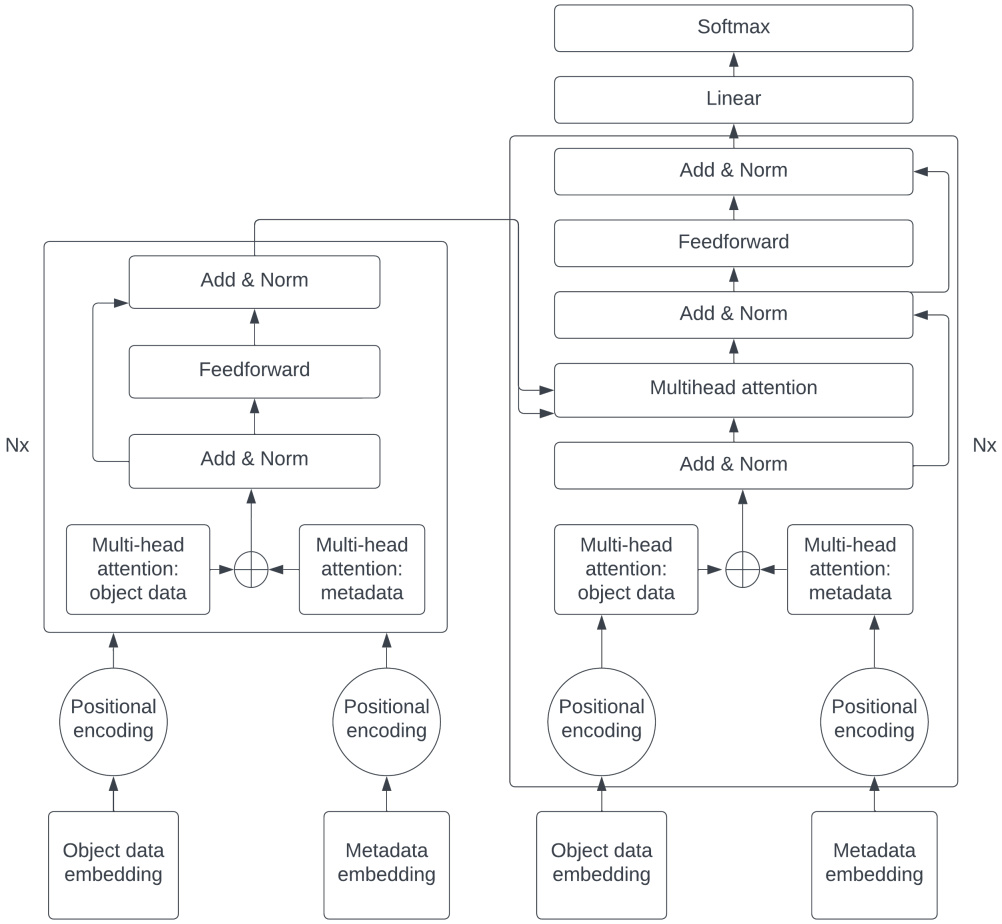

Figure 1: The memory-augmented Transformer with its encoder (left block) and decoder (right block).

While their approach differs from the approach we outline in this paper from an architectural point of view, it is related in combining encoded factual and non-factual information into one representation, as we will point out below.

Memory and retrieval augmentation Adding external memory banks to LLMs is known as *memory augmentation*. In [13], Large Language Models are interpolated with k-nearest neighbor (k-NN) models, using textual similarity of current LLM context with retrieved context(and their completions) for generating interpolated completions. Since k-NN models are memory-based models that store exceptions very well ([14], nearest neighbor LLMs can handle rare patterns and factual information. In a similar vein, *memory augmentation* approaches aim to augment the memory capacities of the neural architectures that produce Large Language Models, by adding extra memory facilities that address external information, like facts from a knowledge base, e.g. [15], where Large Language Models are conditioned on structured external information organized in knowledge graphs. Retrieval-Augmented Generation (RAG, [16]) aims to connect an LLM to external data sources, by mapping user queries or prompts to vectorized representations used as queries for external data sources (like databases). Retrieval results are handled by the LLM to formulate replies that depend on both the user prompt data and the retrieval result. Unlike RAG, we attempt to internalize such external knowledge in the LLM itself. In [17], a separate entity memory ("Entities as Experts") is linked to a Transformer model, that is used for filling in contextual entity mentionings with learned entity embeddings, using contextual similarity matching. Our approach omits multi-step involvement of entity data - we do not explicitly encode separate entity information, but rather force Transformers to focus on external information (not restricted to entities) through the native attention mechanism of Transformers. In [18], the Entities as Experts approach is extended with additional knowledge base facts pertaining to (again) entities. The work of [19] augments the Transformer memory with short term token (i.e. lexical) memory, long term token memory and external token memory. Their approach differs from ours in the type of

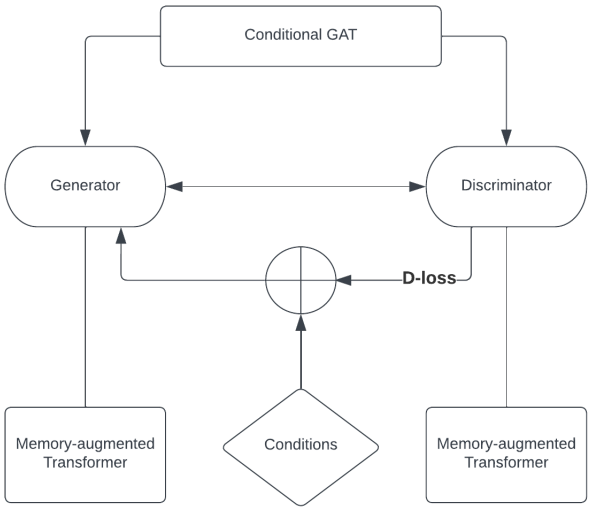

Figure 2: The conditional Generative Adversarial Transformer (GAT), built up from two memory-augmented Transformers. The generator is equipped with additional loss functions conditioning its output. Notice how the generator receives summed losses from the additional loss functions ("Conditions") and the discriminator ("D-loss").

memory data (lexical tokens), and the absence of additional constraint facilities. In contrast, our approach is agnostic to the type of information in the external data memory bank, and allows for arbitrary constraints on outputs. In [20], a separate entity-based external model is proposed, the entirety to which the Transformer pays attention during utterance generation. Our work relates to this approach, in proposing a memory bank for external information, but is more general in allowing for any type of information, including emphasizing existing, internal information from the original input memory, and in allowing for the expression of felicity conditions on the Transformer that can address the extra data memory and the produced responses. Finally, the recent models of [21] propose verbal feedback memory buffers for self-reflection by LLMs.

Prompting [22] and [23] propose to leverage *prompting* (or *in-context learning*, [24, 25]) facilities of current Large Language Models: inserting factual information into a short-term memory buffer of an existing Large Language Model, forcing it to take such information in account when generating new utterances. Our work differs from this work in not explicitly mixing input data with controlling external data, but instead keeping external apart in a separate memory buffer, and by allowing for explicit conditions on outputs through the adversarial organization of the Transformer model.

Generative Adversarial Transformers. Finally, in the image analysis field, Generative Adversarial Transformer models have been proposed, e.g. by [26]. In [27], a Generative Adversarial Transformer architecture has been proposed that exclusively focuses on textual style transfer. Our work can be seen as a generalization of this idea, in being agnostic to the task at hand.

4 Experiments

In a number of experiments we assess the benefits of the external data memory and conditioning the generator of the adversarial Transformers. First, we describe our data, and subsequently we outline our experimental setup.

4.1 Data

In our experiments, we use two datasets:

- The data from ([28]), referred to below as CAR. This is multi-turn conversational data gathered through a *Wizard-of-Oz* experiment for the automotive domain, describing goal-oriented dialogues between humans and a car navigation system.

- The Personalized bAbI data ([29]). This dataset builds upon the well-known bAbI dialogue datasets ([30], and consists of goal-oriented (restaurant table booking) dialogues extended with gender and age information about the human interlocutor. The overarching task is to train dialogue systems to adapt their style to human interlocutor characteristics.

These datasets are described in more detail below.

Factual Question-Answering: CAR

The CAR data contains interactions of humans with a (fictive, i.e. human-imitated) car voice assistant. Drivers can ask the assistant about weather conditions, nearby points of interests ("poi's"), and express navigation instructions. The data is grounded with a knowledge base containing destinations, addresses, weather information and points of interest (POIs). It consists of 3,031 dialogues in three different domain types, all dealing with in-car personal assistants. Uniquely sorting the preprocessed data for the three stages resulted in 5,958 single turn cases for stage 1, 3,864 single turn cases for stage 2, and 3,041 single turn cases for stage 3. Examples are listed in Table 1.

Question/Instruction	**Response**
Where's the nearest parking garage?	The nearest parking garage is Dish Parking at 550 Alester Ave. Would you like directions there?
Yes, please set directions via a route that avoids all heavy traffic if possible.	It looks like there is a road block being reported on the route but I will still find the quickest route to 550 Alester Ave.
Thanks so much for your help.	You're very welcome!
Show me the closest location where i can get chinese food.	The closest chinese restaurant is PF Changs, located 5 miles away.

Table 1: Examples from the data from [28].

In order to obtain fine-grained control over external data for answer generation, we carve up this dialogue process into three stages:

Stage 1: Slot detection. Using the knowledge base (KB) information associated with the dialogues in this dataset, we reverse engineer a slot-tagged version of the dataset, mapping words and phrases to the associated slot names in the KB information that accompanies each turn in a dialogue. We train an adversarial Transformer on the detection of entity slots (names, locations, distances, etc.) from raw data. This Transformer has an empty external data memory bank, since external data do not play a role here. The output of this stage consists of entity-attenuated utterances (Table 1)[1].

Stage 2: Slot mapping. We train a memory-augmented adversarial Transformer on mapping entity-attenuated source utterances (questions, instructions) to entity-attenuated target utterances (responses). In this stage, external data consists of the entity slots specified in the source utterances, which serves to *emphasize* this information for response generation.

Stage 3: Slot filling. We train a memory-augmented adversarial Transformer on mapping the entity-attenuated responses from stage 2 to instantiated utterances, with actual values substituted for slot names. In this stage, external data consists of factual database values.

This setup is illustrated in Figure 3. Sample processed data from the CAR dataset is listed in Table 2.

In order to produce the external data for stage 2 and 3 (which merges this data with the template generated by stage 2), stage 1 delivers crucial information that needs to be converted into a database query. By aligning the entities assigned to words in stage 0 (in the example in Table 1: *fastest* $\mapsto$ *poidistance*; *parking garage*$\mapsto$ *poitype*), we can construct a database query with restrictions, to be resolved against a knowledge base. For instance, for the listed example, such a

[1]The reverse data engineering process was not fault-proof and led in a number of cases to ill-formed data (missing entities, or mistagged entities).

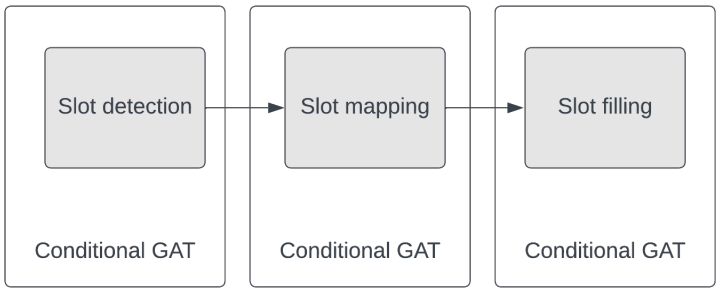

Figure 3: Setup for factual question answering. Three separate memory-augmented Transformers, each trained with their own conditional GAT, address different stages in the process.

Stage 1: slot detection	Stage 2: slot mapping	Stage 3: slot filling
Input: can you find me the fastest route to a parking garage	Input: can you find me the poidistance route to a poitype	Input: poi is poidistance away
External data: none	External data: poitype poidistance	External data: poitype:parking garage poi:webster garage poidistance:4 miles
Output: can you find me the poidistance route to a poitype	Output: poi is poidistance away	Output: webster garage is 4 miles away

Table 2: Examples from our attenuated version of the data from [28], across the three stages.

query submitted to a retrieval engine like Elasticsearch[2] could look like a `GET` operation (Figure 4). The result of this

```
GET /_search
{
  "query": {
      "filter": [
        { "poitype": "parking garage" },
        { "poidistance": { "fastest": { "current-location": "..." }}}
      ]
  }
}
```

Figure 4: Sample Elastic query.

query should be the external data for stage 2, in our example[3]

- `poitype:parking garage, poi:webster garage poidistance:4 miles.`

The external data for stage 2 is essentially a repetition of the extracted slots in the input buffer, produced by stage 1. By representing these slots without context, explicitly in a separate memory buffer and subjecting them to separate attention, we instruct the Transformer to focus on this information during response generation. In summary, for the CAR data we split up dialogue utterance generation into three separate generative steps: an initial slot extraction step, generating attenuated utterances with entities (like dates, destinations, locations) replaced by slot names; a subsequent answer generation step that maps slot-attenuated utterances (questions, commands) to slot-attenuated responses, and a

[2]https://www.elastic.co

[3]In the CAR data we used, such queries have been resolved already.

final step that fills in attenuated slots with values coming from a knowledge base. Under this perspective –which can trivially be generalized to other datasets– question-answering becomes a slot extraction and slot filling problem. In our current experiments, we did not train the three Conditional GAT models in an end-to-end fashion, they were trained independently.

Style adaptation: Personalized bAbI

For the Personalized bAbI data, we focus on different external data: the gender and age information associated with the human interlocutor. The data contains 5 tasks, in different sizes (large and small). We used the dataset for task 3, a small dataset comprising 1,369 single question/answer turns [4].The human-provided initial question and the age/gender information about the interlocutor serve as input for the answer generation process. Typically, age and gender are reflected in the style of the response (higher age leading to more formal, polite answers, for instance). Some examples are listed in Table 3. This data is processed in one step: our Transformer maps input and external data directly to output.

Question	External data	Answer
bombay please	female elderly	would you mind telling me how many guests shall be at your table
bombay please	female young	how many are you
bombay please	male middle-aged	ok sir i'm looking for options for you
can you book a table	female elderly	thank you madam i shall start the reservation now

Table 3: Examples from the Personalized bAbI data from [29].

4.2 Experimental conditions

We carried out three experiments for our two datasets, listed in Table 4.

Experiment	Data	Description
EXP1	CAR data	For stages 1 and 3 for the CAR setup (Table 2), this experiment verifies the linguistic capability of the memory-augmented Transformers. Stage 1 does not use external data, and stage 3 uses external data per definition, to fill in database values in slots detected by stage 1.
EXP2	CAR data	This experiment investigates the use of external data for stage 2 of the CAR setup and the benefits of the POI loss function as a condition on the generator. Total loss is computed by summing standard loss and POI loss, and performance is compared to memory-augmented Transformers with emptied external data.
EXP3	Personalized bAbI data	The benefit of accessing gender and age-related external data. Performance is compared to memory-augmented Transformers with emptied external data. No specific loss function is imposed on the generator.

Table 4: Our experiments.

As mentioned, for the CAR dataset, stage 1 does not use external data, and stage 3 uses external data per definition for filling in slots with values. We therefore only report on the effect of external data or no external data for stage 2 in our

[4]See https://github.com/chaitjo/personalized-dialog for further details of this data.

experiments. Notice that the three models for stages 1-3 were not trained end-to-end; they were trained independently (see Section 6). Additionally, we investigate the effect of a loss function that specifically addresses the overlap in external data between query and generated answer: the *point of interest (POI)* loss function (λ_{poi}), which we define as a function based on recall and precision for POI slot names:

$$
\begin{aligned}
Precision^{poi} &= \frac{TP^{poi}}{TP^{poi}+FP^{poi}} \\
Recall^{poi} &= \frac{TP^{poi}}{TP^{poi}+FN^{poi}} \\
F_1^{poi} &= 2 \cdot \frac{Precision^{poi} Recall^{poi}}{Precision^{poi}+Recall^{poi}} \\
\lambda_{poi} &= 1 - F_1^{poi}
\end{aligned}
\tag{11}
$$

This loss function measures the amount to which POI slot names in the extra data memory appear in the generated predictions. It only checks the presence of slot names, not specific values.

As for the Personalized bAbI data, we test for the benefit of exposing Transformers to the gender and age-related external data (see Table 3).

In our experiments, we uniformly split off a randomized, fixed (over runs) portion of 20% of training data for testing purposes and trained on the remaining 80% of data[5]. All Transformers were trained for 1,000 epochs on a single Tesla T4 16GB GPU. We measure results with the `sacrebleu` ([31]) toolkit[6]. This toolkit computes a cumulative sentence BLEU scores based on word 4-grams (BLEU-4), and additionally computes chrF2 (a character n-gram-based F-score for the match between two utterances, [32]) and TER (Translation Edit Error, [33]), which measures the amount of edit steps needed to convert a machine-generated utterance into a reference utterance (lower TER is better). The `sacrebleu` toolkit also compares results across multiple conditions, with one of them being a baseline, to a reference dataset and reports eventual statistical significance. Additionally, we measure outcomes with the following set of metrics, using the `nlg-eval` toolkit (see [34] for details):

- METEOR: computes an F-score based on a fuzzy alignment of unigrams between source and target utterances, and has been shown to align better with human judgment ([35]).
- ROUGE-L: computes an F-score addressing the *longest common subsequence* between source and target utterances.
- Embedding similarities: SkipThoughtsCosineSimilarity (computes cosine similarity defined on skip-thought sentence embeddings, see [36]), EmbeddingAverageCosineSimilarity (cosine similarity based on averaged word embeddings), VectorExtremaCosineSimilarity (computes cosine similarity between sentence embeddings using extreme values of the constituting word embeddings).
- GreedyMatchingScore: computes cosine similarity based on different pairings of the words in source and target sentences.

4.3 Results

Below we present the results for our three experiments[7]. Note that BLEU scores in the range of 50-60 are generally seen as high (see e.g. [37]).

EXP1 - Slot detection and filling for the CAR data. Tables 5 and 6 list results of stages 1 (slot detection) and 3 (slot filling) for the CAR dataset.

[5]The Transformers parameters were: batch size=8, embedding dimension for input (words) and external data=256, number of attention heads=8.

[6]We ran `sacrebleu` with the signature `-l en-en -m bleu chrf ter -chrf-word-order 4 -b -w 4 -paired-bs`.

[7]The Appendix lists sample responses for the CAR and Personalized bAbI tasks.

System	BLEU-4 (μ ± 95% CI)	chrF2++++ (μ ±95% CI)	TER (μ ± 95% CI)
Stage 1	81.7 (81.6311 ± 1.8414)	86.5 (86.4899 ± 1.3697)	12.5 (12.5455 ± 1.3463)
Stage 3	52.0 (52.0076 ± 2.2496)	61.6 (61.5831 ± 1.7601)	32.5 (32.5215 ± 1.7969)

Table 5: EXP1: the `sacrebleu` results produced for the CAR data for stages 1 and 3.

Metric	Stage 1	Stage 3
BLEU-1	90.5	71.6
BLEU-2	87.5	64.4
BLEU-3	84.8	58.2
BLUE-4	82.1	52.5
METEOR	56.4	36.6
ROUGE-L	92.7	75
SkipThoughtsCosineSimilarity	91.9	78.3
EmbeddingAverageCosineSimilarity	97.9	94.2
VectorExtremaCosineSimilarity	94.9	75.2
GreedyMatchingScore	97.9	88.4

Table 6: EXP1: metric results produced by `nlg-eval` for the CAR data for stages 1 (slot extraction) and stage 3 (slot filling).

While displaying relatively strong BLEU (and other) scores, stage 3 results show room for improvement in terms of factual adherence. Frequently, we witnessed factual hallucinations like:

Input: the poidistance is poi poidistance away at poiaddress
External data: poidistance:3 miles poiaddress:9981 archuleta ave poitype:coffee or tea place poitrafficinfo:moderate traffic poi:peets coffee
Ground truth: the closest is peets coffee 3 miles away at 9981 archuleta ave
Prediction: the nearest is peets coffee 1 miles away at 9981 archuleta ave

It is clear we need to impose additional value-checking loss functions here, which should come as no surprise, since the POI loss function only checks for the restoration of slot names in the generated output, not their values.

EXP2 - Slot mapping for the CAR data. In Table 7, the `sacrebleu` scores for stage 2 for the CAR dataset are displayed. First, we compare using external data with the standard loss to using external data where POI loss is added to standard loss. Second, we compare the basic setting of using no external data and just the standard loss to using external data and POI loss added to standard loss. Third, we compare the use of external data plus the combined POI and standard loss to not using external data but still using the combined POI loss and standard loss. Here, the POI loss is based on the overlap of input buffer POIs with predicted POIs in the generated utterances. This latter comparison specifically assesses the importance of the memory buffer.

System	BLEU-4 (μ ± 95% CI)	chrF2++++ (μ ±95% CI)	TER (μ ± 95% CI)
External data, standard loss	9.9 (9.9140 ± 1.0474)	29.9 (29.8409 ± 1.2403)	90.1 (90.1385 ± 2.3366)
External data, POI loss+standard loss	**11.2** (11.1627 ± 1.1571) (p = 0.0230)*	**31.2** (31.1684 ± 1.2127) (p = 0.0090)*	90.8 (90.9146 ± 2.7216) (p = 0.2208)
No external data, standard loss	10.3 (10.2779 ± 1.1844)	29.9 (29.8587 ± 1.2475)	89.2 (89.2030 ± 2.4749)
External data, POI loss+standard loss	11.2 (11.1627 ± 1.1571) (p = 0.0819)	**31.2** (31.1684 ± 1.2127) (p = 0.0140)*	90.8 (90.9146 ± 2.7216) (p = 0.1169)
External data, POI loss+standard loss	**11.2** (11.1627 ± 1.1571)	**31.2** (31.1684 ± 1.2127)	90.8 (90.9146 ± 2.7216)
No external data, POI loss+standard loss	9.3 (9.2560 ± 1.0709) (p = 0.0040)*	29.2 (29.1112 ± 1.2121) (p = 0.0020)*	89.4 (89.4502 ± 2.2439) (p = 0.1299)

Table 7: EXP2: `sacrebleu` results for the CAR data for stage 2, comparing the standard Transformer loss function with the POI loss function constraining the generator. Boldface and * indicate significantly better results according to `sacrebleu`.

Table 8 displays the `nlg-eval` scores for the conditions of Table 7.

Metric	External data, standard loss	External data, POI loss + standard loss	No external data, standard loss	No external data, POI loss + standard loss
BLEU-1	34.3	**36.0**	34.3	32.3
BLEU-2	22.5	**23.8**	22.4	20.7
BLEU-3	14.7	**15.9**	14.9	13.8
BLEU-4	9.9	**11.2**	10.3	9.3
METEOR	17.5	**18.2**	17.5	16.6
ROUGE-L	32.6	**32.9**	**32.9**	31.0
SkipThoughtsCosineSimilarity	53.6	**54.2**	54.0	52.7
EmbeddingAverageCosineSimilarity	74.3	**74.4**	74.3	71.4
VectorExtremaCosineSimilarity	55.8	**55.9**	55.5	52.9
GreedyMatchingScore	76.6	**78.0**	77.2	76.0

Table 8: EXP2: metric results produced by `nlg-eval` for the CAR data for stage 2, comparing the standard Transformer loss function with the POI loss function constraining the generator. Boldface indicates best results.

According to `sacrebleu`, using external data with the POI loss function combined with standard loss in stage 2 is significantly better than using external data with just the standard loss function. Further, using the memory buffer in conjunction with POI loss and standard loss significantly outperforms using no external data and POI loss combined with standard loss. This confirms the benefit of the extra memory buffer. The results for using no external data, POI loss + standard loss are on a par with using no external data with just the standard loss.

While BLEU scores decrease significantly for stage 2 compared to stages 1 and 3, strict string similarity appears to be not a suitable metric for evaluating this stage. Recall that in stage 2, answer patterns are produced for the attenuated input patterns that are produced by stage 1. A manual inspection for accuracy revealed that the stage 2 patterns produced by Experiment 1 displayed an answer accuracy of 77.7% with accuracy meaning here: leading to a correct answer to the original question, i.e. having the correct slots, but not necessarily matching formulations. Some examples are:

Input: when is poitypetennis activity
External data: poiparty poidate poievent poiagenda poitime
Ground truth: you have a poievent activity on poidate at poitime

*Prediction:*your poievent activity is on poidate at poitime

Input:: who is going
External data: poiparty poidate poievent poiagenda poitime
Ground truth: poiparty will be at the poievent activity on poidate at poitime
Prediction: poiparty will be attending your poievent activity on poidate at poitime

Input: find some places downtown where i can poitype
External data: poiaddress poidistance poitype poi
Ground truth: the poi is poidistance away
Prediction: you will be able to poitype located at poiaddress it is poidistance away

EXP3 - Personalized bAbI. Table 9 lists the BLEU-4 results for the Personalized bAbI task. While this dataset is small and has repetitive answers, using external data yields significantly better performance. Table 10 lists the metric results.

System	BLEU-4 ($\mu \pm 95\%$ CI)	chrF2++++ ($\mu \pm 95\%$ CI)	TER ($\mu \pm 95\%$ CI)
External data	**61.1** (60.9185 ± 5.7854)	**67.7** (67.6811 ± 5.0516)	**37.4** (37.3672 ± 5.6361)
No external data	8.8 (8.7159 ± 3.1313) (p = 0.0010)*	20.8 (20.7812 ± 2.6477) (p = 0.0010)*	97.5 (97.5206 ± 4.4556) (p = 0.0010)*

Table 9: EXP3: the `sacrebleu` results for the Personalized bAbI data. Boldface and * indicate significantly better results according to `sacrebleu`.

Metric	External data	No external data
BLEU-1	**67.2**	21.8
BLEU-2	**64.5**	14.6
BLEU-3	**62.8**	10.7
BLEU-4	**61.1**	8.8
METEOR	**39.5**	11.1
ROUGE-L	**65.6**	22.5
SkipThoughtsCosineSimilarity	**78.6**	46.8
EmbeddingAverageCosineSimilarity	**89.0**	80.1
VectorExtremaCosineSimilarity	**73.6**	50.0
GreedyMatchingScore	**82.6**	63.4

Table 10: EXP3: metric results produced by `nlg-eval` for the Personalized bAbI data. Boldface indicates best results.

5 Discussion

In our experiments, we have found initial indications that informing adversarially trained Transformer models with additional external data and constrained generators can be helpful. Using just external data with the standard loss function appears not to outperform the standard setting where no external data is used. However, we found that using external data with a tailored loss function (POI loss) compared to the standard loss function improves stage 2 of the CAR data experiment. Using POI loss as a condition on the Transformer generator also produces better absolute results across all metrics minus 1 for stage 2 of the CAR data. The use of the extra memory buffer plus the POI loss function outperformed using no external data and POI loss applied to the input buffer and predictions. This indicates that the extra memory is indeed useful for this task. Our results can be interpreted as a within-system form of ablation, since we tested for using no external data versus using external data, effectively shutting the extra memory buffer off and on. For Stage 2 of the CAR dataset, we effectively used the extra memory buffer to emphasize the data already in the input buffer - the buffer basically repeats that information. In stage 3 for the CAR dataset, it is clear that the external data (which, in that stage, consists of non-repeated database values) is crucial for filling in the slots names with specific

values. For the Personalized bAbI dataset, the situation is actually similar: the external data is crucial for setting the right tone of voice, which is dependent on the age and gender of the human interlocutor. Such data is not redundantly available in the input buffer. The Personalized bAbI task involves only the extra conditioning of answer generation on two external variables (the gender and age of the human interlocutor). This may explain the effectiveness of external data we observed for this dataset.

Limitations of our research consist of a partial assessment of the use of additional loss functions, and the relatively small sample size for our training and test datasets. Also, for practical (computing) reasons, we did not perform a systematic grid search over the various Transformer parameters, including the number of training epochs. We have not addressed the full potential of conditioning the adversarial aspect of our approach yet. For instance, coding aspects of past utterances in dialogues (like sentiment, style, or topic) may lead to better regulation of future utterances, in conjunction to the general in-context learning capabilities of large language models. Finally, we are aware that our evaluation metrics address only string and semantic overlap between ground truth and predicted data. Follow-up research should extend these metrics with more fine-grained evaluation metrics that target factual adherence to external data.

6 Future work

As for future work, we see a number of potential topics:

End-to-end training In our current setup for factual question-answering, we have trained three memory-augmented Transformers in isolation. An end-to end schema would imply joint optimization with potential performance benefits, and will be investigated in our follow-up research.

Quality improvement for factual adherence Additional loss functions will need to be applied in order to tighten the adherence of the GAT models to factual data, as witnessed by the frequent hallucination in stage 3 for the CAR dataset. Factual evaluation metrics will be used for a better estimation of factual adherence.

Structured external data Structured external data in the form of knowledge graphs or ontologies can improve the answers in stage 2 and 3 for the CAR dataset. Previous work has shown the value of using knowledge graphs as an external knowledge base to question answering with language models [38, 39]. The CAR dataset currently contains a knowledge base per scenario, structured in a triple format [28]. These knowledge bases are manually crafted per scenario and have limited information; useful in a limited experiment setting but not scalable for real-life applications. Large graph databases such as Neo4j [40] provide a large factbase and allow for discovery of patterns and flexible extensions, as [41] demonstrate for the medical domain. For follow-up research, it would be interesting to explore the benefits of different types of knowledge graphs as external fact databases.

Comparison with Retrieval-Augmented Generation (RAG) models In future work, we intend to compare the GAT architecture to RAG architectures, on shared datasets.

Reinforcement learning from explicit human feedback Reinforcement learning from human feedback (RLHF) [42, 5] can potentially further condition the GAT by providing a mechanism for improving performance based on human-generated rewards or evaluations. This presupposes the detection of appropriate reward signals that capture the desired behavior of the system. For example, in the case of factual question answering, the reward signal could be based on the accuracy of the answer provided by the system compared to a human-provided answer, but does not have to involve a direct human-in-the-loop. A reward model can be trained using human labelers, which in turn can be used to provide a reward signal, making the process time- and cost-efficient. This technique can be applied to all three stages of the dialogue process described above. Sometimes, based on new evidence or research methods, factual knowledge can change. By incorporating RLHF, the memory can be updated based on the reward policy. This allows the system to dynamically adapt its memory content and attention focus based on the feedback received through reinforcement learning.

Reinforcement learning from implicit human feedback As human-system dialogues become more natural, human feedback to system output is better reflecting the grounding process that takes place in human-human conversations. This means that during a conversation partners are continuously making sure their utterances are mutually understood ([43, 44, 45]). After the presentation phase, in which the first partner presents an utterance, the acceptance phase follows, in which the second partner provides evidence of understanding. Grounding is often achieved through the use of back-channel responses, such as “uh huh” or “mm” ([46]). However, the acceptance phase may take several turns, including sequences of clarification requests and repairs. Once both phases are complete, it will be common ground between both partners that the second partner has understood what the first partner meant. In addition, this

whole process may involve dialog acts indicating the second partner is processing the utterance of the first partner (autofeedback), or the partner needs some time to formulate his contribution the dialog (time management). Problems with understanding or formulating may also become apparent in delays in responding ([47]). The response to the grounded utterance of the first partner depends on the dialog act of this utterance and the corresponding expected dialog act of the second partner. These expected pairs of dialog acts are called adjacency pairs, of which typical examples are: question–answer, greeting–greeting and offer–acceptance ([48]). The response may turn out to adhere to this expectation (e.g., offer-acceptance), contradict it (e.g., offer–refusal) or leave it undecided (e.g. offer-hesitation). Both length and nature of the process of achieving common ground, occurrence of autofeedback and time management dialog acts or delays in answering, and the adherence of the ultimate response of the human to the system's dialog act may provide valuable input to reinforcement learning.

The exact interplay of memory augmentation and feedback will be an interesting new research topic.

Fine-tuning Our experiments started *ab ovo* with a proprietary dataset and an untrained Transformer. In future work, we will attempt to reconcile our approach with pre-trained large language models (*base* or *foundation* models) by interpreting our memory augmentation as a form of fine-tuning.

7 Conclusion

This paper has presented a new type of memory-augmented Transformers: adversarially trained generative Transformers that can be conditioned on arbitrary loss functions imposed on their generators. We have demonstrated the efficacy of an adversarial training scheme for textual Transformers, and found indications that adding external information through memory augmentation leads to performance improvement of our models, for two divergent tasks: factual question-answering and style adaptation. Our approach is completely agnostic with respect to the type of external data, and may lead to insights which types of external data are beneficiary for a given task at hand. For handling factual question-answering, we outlined an approach based on attenuation and de-attenuation (slot detection, slot mapping, and slot filling), using memory augmentation for emphasizing certain parts of the input data. While we did not arrive at high quality factual adherence in the slot filling stage of these latter experiments yet, we hypothesize that additional value-checking loss functions will be effective for raising performance. Memory augmentation with truly external data (i.e. data that is not explicitly repeated in the input buffer) appeared useful for a small scale style adaptation dataset. While our results are based on limited data and therefore should not be taken as conclusive, we feel encouraged to further explore the proposed approach across different tasks and on larger datasets.

Acknowledgments

This research was partially supported by the TNO Research Programme APPL.AI (https://appl-ai-tno.nl).

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, page 5998–6008. Curran Associates, Inc., 2017.

[2] Le Scao et al. Bloom: A 176b-parameter open-access multilingual language model. In *BigScience Workshop et al.*, volume abs/2211.05100, 2023.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020.

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

[5] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. cite arXiv:2203.02155.

[6] Christine Liebrecht, Lena Sander, and Charlotte van Hooijdonk. Too informal? How a chatbot's communication style affects brand attitude and quality of interaction. In Asbjørn Følstad, Theo Araujo, Symeon Papadopoulos, Effie L.-C. Law, Ewa Luger, Morten Goodwin, and Petter Bae Brandtzaeg, editors, *Chatbot Research and Design*, pages 16–31, Cham, 2021. Springer International Publishing.

[7] Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. Factuality enhanced language models for open-ended text generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 34586–34599. Curran Associates, Inc., 2022.

[8] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023.

[9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv*, 2014. cite arXiv:1406.2661.

[10] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv*, abs/1411.1784, 2014.

[11] Marjan Ghazvininejad, Chris Brockett, Ming-Wei Chang, Bill Dolan, Jianfeng Gao, Wen tau Yih, and Michel Galley. A knowledge-grounded neural conversation model. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*, pages 5110–5117. AAAI Press, 2018.

[12] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv*, abs/2302.12813, 2023.

[13] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*, 2020.

[14] Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. Forgetting exceptions is harmful in language learning. *Mach. Learn.*, 34(1-3):11–41, 1999.

[15] Qi Liu, Dani Yogatama, and Phil Blunsom. Relational Memory-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 10:555–572, 05 2022.

[16] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, and N. Goyal et al. Retrieval-augmented generation for knowledge-intensive nlp tasks.

[17] Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. Entities as experts: Sparse memory access with entity supervision, 2020.

[18] Pat Verga, Haitian Sun, Livio Baldini Soares, and William Cohen. 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*, pages 3678–3691, Online, June 2021. Association for Computational Linguistics.

[19] Zexuan Zhong, Tao Lei, and Danqi Chen. Training language models with memory augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5657–5673, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[20] Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William Cohen. Mention memory: incorporating textual knowledge into transformers through entity mention attention. *arXiv*, abs/2110.06176, 2021.

[21] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

[22] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *arXiv*, abs/2302.00083, 2023.

[23] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. Replug: Retrieval-augmented black-box language models, 2023.

[24] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv*, abs/2202.12837, 2022.

[25] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. *arXiv*, abs/2211.01910, 2023.

[26] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4487–4499. PMLR, 18–24 Jul 2021.

[27] Kuo-Hao Zeng, Mohammad Shoeybi, and Ming-Yu Liu. Style example-guided text generation using generative adversarial transformers. *arXiv*, abs/2003.00674, 2020.

[28] Mihail Eric, Lakshmi Krishnan, Francois Charette, and Christopher D. Manning. Key-value retrieval networks for task-oriented dialogue. In Kristiina Jokinen, Manfred Stede, David DeVault, and Annie Louis, editors, *SIGDIAL Conference*, pages 37–49. Association for Computational Linguistics, 2017.

[29] Chaitanya K. Joshi, Fei Mi, and Boi Faltings. Personalization in goal-oriented dialog. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.*, 2017.

[30] Antoine Bordes, Y-Lan Boureau, and Jason Weston. Learning end-to-end goal-oriented dialog. In *ICLR 2017*, 2017.

[31] Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October 2018. Association for Computational Linguistics.

[32] Maja Popović. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[33] Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA, August 8-12 2006. Association for Machine Translation in the Americas.

[34] Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799, 2017.

[35] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[36] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

[37] Google. Evaluating models. `https://cloud.google.com/translate/automl/docs/evaluate`, 2023. [Online; accessed 25-April-2023].

[38] Lihui Liu, Boxin Du, Jiejun Xu, Yinglong Xia, and Hanghang Tong. Joint knowledge graph completion and question answering. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1098–1108, 2022.

[39] Linyao Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. *arXiv*, abs/2306.11489, 2023.

[40] Neo4j. Neo4j - the world's leading graph database, 2012.

[41] Zhixue Jiang, Chengying Chi, and Yunyun Zhan. Research on medical question answering system based on knowledge graph. *IEEE Access*, 9:21094–21101, 2021.

[42] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[43] H.H. Clark. *Using language*. Cambridge University Press, Cambridge, 1996.

[44] Herbert H. Clark and Ed Schaefer. Contributing to discourse. *Cogn. Sci.*, 13:259–294, 1989.

[45] Herbert H. Clark and Susan Brennan. Grounding in communication. In *Perspectives on socially shared cognition*, 1991.

[46] Emanuel A. Schegloff. Discourse as an interactional achievement: some uses of 'uh huh' and other things that come between sentences. In Deborah Tannen, editor, *Analyzing Discourse: Text and Talk*, page 71–93. Georgetown University Press, Washington, D.C., 1982.

[47] Harry Bunt. Multifunctionality in dialogue. *Computer Speech & Language*, 25:222–245, 04 2011.

[48] Emanuel A. Schegloff and Harvey Sacks. Opening up closings. *Semiotica*, 8(4):289–327, 1973.