



Injecting Knowledge Graph Triples into Large Language Models for Passage & Question Generation

Charles Foster, Jesse Hamer,
Joel Walsh, Geoff Converse
and Claudio Souza

Contents

1. Introduction	4
2. Related Work	5
3. Pipeline Background and Motivation	7
3.1 <i>Why triples?</i>	7
3.2 <i>Description of Pipeline</i>	8
4. Results.....	10
4.1 <i>Multiple choice questions</i>	10
4.2 <i>Constructed response</i>	12
4.3 <i>Passage generation</i>	12
5. Discussion	13
6. Conclusion	14
7. References.....	14
8. Appendix	16
8.1 <i>Appendix A: Knowledge Graph Triples in GPT-3</i>	16
8.2 <i>Appendix B: Multiple Choice</i>	16
8.3 <i>Appendix C: Multiple facts, multiple choice</i>	17
8.4 <i>Appendix D: Constructed response</i>	19
8.5 <i>Appendix E: Passage generation</i>	20

Executive Summary

The performance of Large Language Models (LLMs) has progressed in leaps and bounds in the last decade, especially with respect to tasks like passage and question generation. However, the specter of confabulation (“hallucinated” claims) and incorrectness still haunts most language model outputs. This makes their use in our industry difficult, as factual accuracy is a critical feature for content that is meant to measure competency. Assessment developers cannot afford to now worry about confabulation on top of the myriad of other requirements a high-quality assessment must meet.

This paper details one proposed solution for language model confabulation; the injection of structured knowledge via knowledge graph triples. Methods from the past few years achieve gains in factualness by fusing knowledge graph entity embeddings with semantic and positional embeddings.

Wikidata2LLM bypasses these complicated architectures by querying an entity’s associated knowledge graph triples directly from the Wikidata knowledge base. The triples are then entered into a language model via structured natural language prompts. Early results indicate that this method produces multiple choice, constructed response, and short passages that utilize these facts, a potential remedy to the problem of confabulation.

1. Introduction

Large Language Models have achieved great success at generating text, as measured through benchmarks datasets that measure perplexity like SuperGLUE (Wang et al., 2020). This success has extended to question answering, as LLMs have had great success in achieving state of the art in many popular question answering (QA) datasets. The designers of LLMs mainly achieve this through scaling up model parameters and training data (Brown et al., 2020). However, further studies have shown that many popular QA dataset questions were simply memorized by large models. This tendency of LLMs to fabricate claims, especially in specialized content domains like health, law, finance, and politics (Lin, Hilton, and Evans 2021), has come to be known as confabulation or “hallucination”. This phenomenon affects the performance of LLMs on a range of tasks, to include machine translation, question answering and generation, and abstractive summarization. Confabulation can appear due to a multitude of design factors in both training and inference (Ji et al., 2022).

For LLMs to answer domain specific questions accurately, the models must have some grounding in verified facts. For the related task of question generation, it is imperative for models to be able to generate questions that are grounded in factual accuracy and can be answered truthfully. One method of ensuring that language models are producing truthful outputs is to inject structured knowledge. Several methods incorporate some sort of fusion layer, whereby entity level knowledge embeddings are fused with semantic, contextual, and positional embeddings. However, the few and no-shot capabilities of LLMs open up new possibilities for

injecting structured knowledge into language models directly via natural language prompts, which doesn't require entity embeddings or any changes to either the computational graphs or sampling procedures of a language model. The Wikidata2LLM method proposes something much simpler: to pipe structured knowledge directly into LLM prompts. There are many reasons to believe that this will help to systematically create ground truth questions. For one, LLMs appear to have the capability of turning structured knowledge into coherent sentences, and draw on this knowledge to form question items. Second, large LLMs like GPT-3 seem to contain structured knowledge within the existing model parameters, as evidenced by experiments that elicit this knowledge (see appendix A).

2. Related Work

The task of question generation is of particular utility in educational technology, as automated or AI-assisted test generation can lead to savings of time and labor within organizations. In NLP, one parameterization of this task is to take a passage or set of facts in natural language as input, and produce a question as output. While earlier NLP approaches to question generation relied on rules and other heuristics, later approaches used seq2seq neural architectures and copy mechanisms (Qiu and Xion, 2019). As with many NLP tasks, LLMs have made it possible to eliminate many of these heuristics and copy mechanisms (Qui and Xion, 2019). LLMs have enabled new techniques which avoid these mechanisms, such as Generate, a content generation system which uses LLMs to create assessment content in a wide variety of domains and formats (Khan, Hamer, and Almeida, 2021).

Yet still, LLMs by themselves have important technical limitations, related to the phenomenon of confabulation.

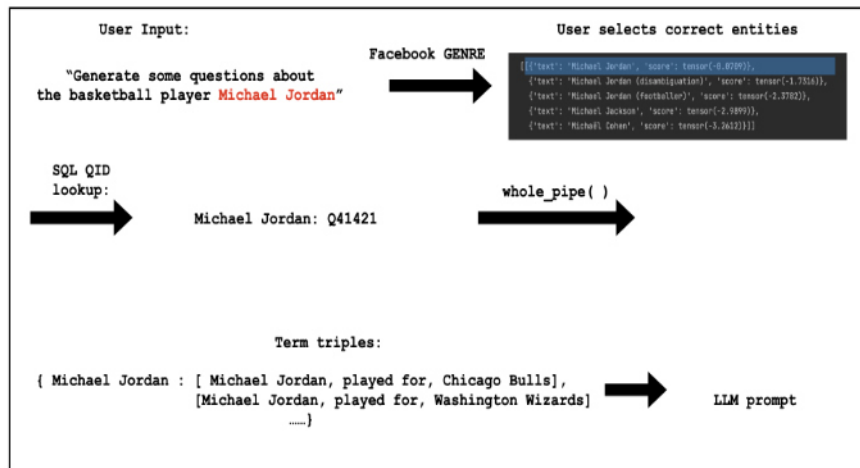


Figure 1: Description of pipeline

While LLMs have generally excelled at creating text that is stylistically and grammatically similar to training corpora, the models are purely autoregressive. This leads to a phenomenon of confabulated content also known as “hallucination” or, “fluent but unsupported text” (Filippova, 2020, p.1). Consider the prompt “Barack Obama was born on”. The LLM GPT-2 will complete this text with “January”, when it is actually “August” (Logan IV et al., 2019). This tendency of LLMs has implications for a number of applications; such as neural machine translation, abstractive summarization, and generative question answering (Ji et al., 2022).

Several methods exist to combat confabulation. One method involves combining structured knowledge into the autoregressive output of language models. These can be achieved using a number of approaches from fusing entity embeddings to existing semantic embeddings (Zhang et al., 2019; Peters et al., 2019) to using external knowledge bases directly (Logan IV et al., 2019). These techniques generally involve some form of fine-tuning or compute-heavy training regiment.

3. Pipeline Background and Motivation

Wikidata2LLM is a method that seeks to eliminate complicated fusion or external memory techniques that require additional training, instead feeding structured knowledge directly into an LLM for question generation. The atomic unit of structured knowledge in Wikidata, and all knowledge graphs, is the triple. A triple consists of two entities, e_1 and e_2 , and their relation to each other. Wikidata is a knowledge graph project that draws on Wikipedia information (Canon et al., 2016). While Wikidata hardly represents all of the known structured knowledge about a given topic, it is perhaps the most comprehensive open source knowledge graph. While Google’s knowledge graph is most likely larger and more comprehensive (it builds on top of Wikidata), it is not as accessible as Wikidata.

3.1 Why triples?

Structured knowledge can come in many forms, but knowledge graph triples have proven to be superior to older forms like semantic networks. Knowledge graph relations are intentionally

kept to a minimum in a knowledge base, which allows for easier comparisons between entities. It would also appear as if LLMs like GPT-3 have enough exposure to knowledge graph triples in training to generate triples without examples (See figure x). However, there is a definite limit to how much structured knowledge exists within the parameters of an LLM like GTP-3. Simply prompting, “Write all of the facts about Type 2 diabetes in knowledge graph triples”, or , “Write all of the facts about Type 2 diabetes ”, will result in sets of facts that are relatively shallow compared to the set of facts within a Wikidata entry for Type 2 diabetes.

3.2 Description of Pipeline

Modern LLMs require only natural language prompts in order to produce outputs, a process known as zero or few shot learning (Wei et al., 2021; Brown et al., 2020). These prompts can be entered through APIs. In the case of Open AI’s GPT-3, the playground function was used for all experiments. The particular GPT-3 model used for this analysis was text-davinci-003, a model fine tuned to receive human instructions. For the purposes of comparison the temperature parameter was set to 0, in order to create deterministic outputs that can be reproduced. To show the efficacy of this approach, consider the LLM prompt: “Write a multiple choice/open ended question using the fact(s): [triples] ”.

The Wikidata to GPT-3 method requires a user’s request to be linked to the appropriate Wikipedia entity (by QID number) containing that information. Consider the query, “Create a multiple choice question about the business mogul and former basketball

player Michael Jordan". This would necessarily need to refer to the Chicago Bulls' Michael Jordan, just as "Create a multiple choice question about the pioneer of expectation maximization Michael Jordan", would need to link to the computer science professor's QID in Wikipedia. There are a number of neural models that take in sentences with mentions of entities and output the top-k Wikipedia QID matches and their associated confidence scores. The best performing models for short prompts were from Facebook's GENRE Autoregressive Entity Retrieval suite (De Cal, et al., 2021).

Once the correct entity is identified by Wikipedia QID, all of the associated triples can be generated through the JSON file hosted by Wikidata. These triples can also be gathered through SPARQL queries, but using the entire JSON entity dump allows for a faster and more flexible parsing process in Python. The raw JSONs from Wikidata, and the triples extracted from these files are entirely in Q (entities) and P (relations) ID form. In order to be useful for GPT-3 they need to be translated back into natural language. This is done via two methods: a SQLite database and a python dictionary.

As there are relatively few relations/ PIDs (10,000 in the latest Wikidata release), a native python dictionary is a decent method of looking up natural language values for P ID keys. A key, value lookup is $O(1)$ to $O(n)$ worst case, so it makes sense to segregate the relations from the entity lookup. For entity lookups, a 1.92 GB SQL database holds all of the English key, value pairs compiled from pickled dictionaries provided by Facebook's GENRE team. This is pared down from an 3GB database that originally included several other widely used languages. In order to avoid re-querying

entities while processing the triples associated within a user-defined entity, recurring queries are memoized in a Python dictionary as the triples are processed. Queries to this SQL database could possibly be sped up more by adopting alternative indexing system. The end result is the list of triples associated with a given entity, in natural language form. This list can be randomly sampled and appended to question prompts (See Figure 1).

4. Results

For all prompts, the language model was GPT-3 text-davinci003, temperature set to 0, and maximum length is set to 256 tokens. This could theoretically be replaced with any large language model. Stop tokens were generated automatically. These examples highlight question and passage generation for one particular entity, Type II Diabetes. The narrow scope of this analysis is mainly due to the difficult and time consuming nature of prompt/factual accuracy evaluation. However, these initial results have been promising, and point to necessity for much more thorough research on the effects of injecting knowledge into LLM prompts.

4.1 Multiple choice questions

As a baseline, consider the prompt : Write a multiple choice question about Type 2 diabetes . The first modification to this prompt was to add one random fact in triple form, so that the prompt read: Write a multiple choice question using the fact (Type 2 diabetes, drug or therapy, ACE inhibitor). The baseline question resulted in an “All of the above” option, which is the correct answer

(see Appendix B). The addition of a fact triple added greater specificity to the question, but with one caveat. The question becomes worded as “Which of the following”, implying that there could be more than one correct answer. It appears that three of the answers are correct, as Metformin and Insulin are also used for the treatment of Type 2 diabetes (See appendix B). So while injecting a single fact does lead to greater specificity in model output, it might not always mean that only one answer will be correct. It should be noted that adding the entire generated question (essentially re-prompting the language model with its own output) did not result in the model answering the question, but instead generated stop tokens for this prompt.

Adding additional related facts into this multiple choice prompt yielded some unexpected results, such as the model eventually providing answers and injecting the extra facts into wrong answer choices (See Appendix C). There are two ways that additional facts could be related, by relation and entity or just by entity. An example of a related fact by relation and entity would be (Type 2 diabetes, drug or therapy used for treatment, healthy diet). An example of a fact with the same entity but different relation is (Type 2 diabetes, risk factor, smoking). Adding a fact with the same relation led to an item that allowed for both or neither healthy diet and ACE inhibitors as answer options. Using any additional fact with a different relation inserted smoking as a distractor answer as a drug or therapy used for treating Type 2 diabetes (See appendix C). While this is a somewhat absurd distractor, it does hint that additional facts could possibly be leveraged to create informative distractors.

4.2 Constructed response

For constructed response question types, adding fact triples is less useful. To analyze free response, the term “multiple choice” was changed to free response, with all of the previous perturbations (no facts, one fact, two facts with the same relation and entities, two facts with different relations and the same entities). While adding no facts or one fact tended to generate somewhat relevant questions, adding more than one fact led to some questions that clumsily combined the facts. For example, the prompt [Write a constructed response question using the facts (Type 2 diabetes, drug or therapy, ACE inhibitor), (Type 2 diabetes, risk factor smoking)]. led to the question, “ What are the potential benefits of using an ACE inhibitor as a drug or therapy to treat Type 2 diabetes in individuals who smoke? ” (See appendix D).

4.3 Passage generation

Experiments with passage generation generally found that the injection of facts led to more specific engagement. As a baseline, consider the prompt “Write a short passage about ACE inhibitors”. Curiously, the addition of more facts led to shorter and shorter passages. If some of these facts contained the same relations, the language model tended to put them into the same sentences in a list format, which probably explains the generation of shorter passages. Adding one fact to the prompt also added some increased specificity with respect to domain-specific terminology (See Appendix E). The baseline prompt referred to the causal

mechanisms of ACE inhibitors in more general terms, using “chemical” to refer to Angiotensin II, whereas the one fact prompt used the more appropriate “protein” terminology.

5. Discussion

Wikidata2LLM-style pipelines have engineering and enduser advantages, but there is much additional research that can help calibrate future approaches. While the (entity1, relation, entity2) triple format was used in this analysis, LLMs tend to have seen, and can reproduce other knowledge graph formats like Resource Description Framework (RDF). It may be helpful to see if feeding in knowledge via alternate formats creates questions that better match an end user’s specifications. While many of the facts in Wikidata follow a simpler (entity1, relation, entity2) format, many facts contain the slightly more complicated qualifier . Initial attempts to prompt LLMs to create coherent sentences and questions from these prompts were not consistent, yet some degree of few shot learning and prompting may unlock the potential of these fact types. Lastly, some Wikidata triples contain two common relationships; subclass and instance of. These relations enable a number of ontological/hierarchical possibilities, which can lead to more nuanced and rigorous question generation if appropriately leveraged.

Initial tests appear to show that inserting knowledge graph triples into language model prompts leads to desirable outcomes in question generation, like factualness and specificity. If the triples are grounded in real world knowledge, using them in LLM prompts will generally result in questions that are grounded real world

knowledge. This is not to say that using knowledge graph triples will always result in passages and questions that contain only facts that are grounded in the supplied triples, and additional research would be helpful in order to explore this tendency of individual LLMs to bring in either outside facts or untrue assertions.

6. Conclusion

The prompts used in this analysis are rather simple and can be seen as a baseline approach to the incorporation of structured knowledge to LLM inputs. That being said, early results on generating factual items have been encouraging, both along dimensions of efficiency and factuality. While subject matter experts and curriculum experts are still the ideal source for evaluating factfulness and appropriateness, knowledge based methods like Wikidata2LLM can help them quickly author potential test items with increased confidence that flaws in AI assistance will not undermine their work. With mitigations in place for language model confabulations, the focus of research and development can turn towards other important aspects of item quality, including fit for purpose, fairness, appropriate cognitive complexity, and reliability.

7. References

De Cao, N., Izacard, G., Riedel, S., Petroni, F. (2021). Autoregressive Entity Retrieval (arXiv:2010.00904). arXiv. <https://doi.org/10.48550/arXiv.2010.00904>

Filippova, K. (2020). *Controlled Hallucinations: Learning to Generate Faithfully from Noisy Data*. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 864–870. <https://doi.org/10.18653/v1/2020.findings-emnlp.76>

Khan, S., Hamer, J., Almeida, T. (2021, June). *Generate: A NLG system for educational content creation*. *Proceedings of the 14th Educational Data Mining Conference*. *Educational Data Mining*.

Logan IV, R. L., Liu, N. F., Peters, M. E., Gardner, M., Singh, S. (2019). *Barack’s Wife Hillary: Using KnowledgeGraphs for Fact-Aware Language Modelling (arXiv:1906.07241)*. *arXiv*. <https://doi.org/10.48550/arXiv.1906.07241>

Pellissier Tanon, T., Vrandečić, D., Schaffert, S., Steiner, T., Pintscher, L. (2016). *From Freebase to Wikidata: The Great Migration*. *Proceedings of the 25th International Conference on World Wide Web*, 1419–1428.

Peters, M. E., Neumann, M., Logan IV, R. L., Schwartz, R., Joshi, V., Singh, S., Smith, N. A. (2019). *Knowledge Enhanced Contextual Word Representations (arXiv:1909.04164)*. *arXiv*. <https://doi.org/10.48550/arXiv.1909.04164>

Qiu, J., Xiong, D. (2019). *Generating Highly Relevant Questions (arXiv:1910.03401)*. *arXiv*. <https://doi.org/10.48550/arXiv.1910.03401><https://doi.org/10.48550/arXiv.2109.01652>

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q. (2019). *ERNIE: Enhanced Language Representation with Informative Entities*.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 1441–1451. <https://doi.org/10.18653/v1/P19-1139>

8. Appendix

All prompts were submitted to GPT-3: text-davinci-003, temperature = 0, Maximum Length = 256.

8.1 Appendix A: Knowledge Graph Triples in GPT-3

Prompt: Write all of the facts about Type 2 diabetes in knowledge graph triple (entity, relation, entity) format.

Output: Type 2 Diabetes, is a type of, Diabetes)
(Type 2 Diabetes, is caused by, Insulin Resistance)
(Type 2 Diabetes, is characterized by, High Blood Sugar)
(Type 2 Diabetes, is associated with, Obesity)
(Type 2 Diabetes, is treated with, Diet and Exercise)
(Type 2 Diabetes, is managed with, Medication)
(Type 2 Diabetes, can lead to, Complications)

8.2 Appendix B: Multiple Choice

8.2.1 Baseline no fact

Prompt: Write a multiple choice question about Type 2 diabetes.

Output: Type 2 diabetes is caused by:

- A. Genetics
- B. Poor diet
- C. Lack of exercise
- D. All of the above

8.2.2 One fact

Prompt: Write a multiple choice question using the fact (Type 2 diabetes, drug or therapy used for treatment, ACE inhibitor).

Output: Which of the following is a drug or therapy used for the treatment of Type 2 diabetes?

- A. Metformin
- B. Insulin
- C. ACE inhibitor
- D. Glucagon

8.3 Appendix C: Multiple facts, multiple choice

8.3.1 Multiple facts, same relation

Prompt: Write a multiple choice question using the facts [(Type 2 diabetes, drug or therapy used for treatment, ACE inhibitor), (Type 2 diabetes, drug or therapy used for treatment, healthy diet)]

Output: Which of the following is a drug or therapy used for treatment of Type 2 diabetes?

- A. ACE inhibitor
- B. Healthy diet
- C.Both A and B
- D. Neither A nor B

8.3.2 Multiple facts, different relation

Prompt: Write a multiple choice question using the facts [(Type 2 diabetes, drug or therapy used for treatment, ACE inhibitor), (Type 2 diabetes, risk factor, smoking)]

Output: Which of the following is a drug or therapy used for the treatment of Type 2 diabetes?

- A. Smoking
- B. ACE inhibitor

C. Insulin

D. Diet

8.4 Appendix D: Constructed response

Prompt: Write a free response question about Type 2 diabetes.

Output: What lifestyle changes can someone with Type 2 diabetes make to help manage their condition?

Prompt: Write a free response question using the fact (Type 2 diabetes, drug or therapy, ACE inhibitor).

Output: What are the potential benefits and risks of using an ACE inhibitor to treat Type 2 diabetes?

Prompt: Write a free response question using the facts [(Type 2 diabetes, drug or therapy used for treatment, ACE inhibitor), (Type 2 diabetes, risk factor, smoking)]

Output: What are the potential benefits of using an ACE inhibitor as a drug or therapy to treat Type 2 diabetes in individuals who smoke?

Prompt: Write a free response question using the fact [(Type 2 diabetes, drug or therapy used for treatment, ACE inhibitor), (Type 2 diabetes, drug or therapy used for treatment, healthy diet)].

Output: What are the benefits of combining an ACE inhibitor with a healthy diet for the treatment of Type 2 diabetes?

8.5 Appendix E: Passage generation

ACE inhibitors are a type of medication used to treat high blood pressure and other cardiovascular conditions. They work by blocking the action of a chemical called angiotensin II, which causes blood vessels to constrict. This makes it easier for the heart to pump blood and helps to lower blood pressure. ACE inhibitors are also used to treat heart failure and prevent strokes. Prompt: Write a short passage about ACE inhibitors

Output: ACE inhibitors are a type of medication used to treat high blood pressure and other cardiovascular conditions. They work by blocking the action of a chemical called angiotensin II, which causes blood vessels to constrict. This makes it easier for the heart to pump blood and helps to lower blood pressure. ACE inhibitors are also used to treat heart failure and prevent strokes.

Prompt: Write a short passage using the fact (Type 2 diabetes, drug or therapy used for treatment, ACE Inhibitor)

Output: ACE inhibitors are a type of drug used to treat high blood pressure and other conditions. They work by blocking the action of a protein called angiotensin-

converting enzyme (ACE). This protein is involved in a number of processes in the body, including narrowing of blood vessels. By blocking the action of ACE, ACE inhibitors help to relax blood vessels and reduce blood pressure. ACE inhibitors are also used to treat other conditions, such as heart failure and diabetes.

Prompt: Write a short passage using the following facts: (Diabetes, cause of, Cataract), (Diabetes, cause of, Arsenic neuritis), (Diabetes, cause of, Diabetic retinopathy)

Output: Diabetes has been linked to a number of serious health conditions, including cataracts, arsenic neuritis, and diabetic retinopathy. Each of these conditions can lead to serious vision problems, and in some cases, blindness. If you have diabetes, it is important to monitor your vision closely and see your doctor regularly to help prevent these complications.

Prompt: Write a short passage using the following facts: (Diabetes, cause of, Cataract), (Diabetes, cause of, Arsenic neuritis), (Diabetes, cause of, Diabetic retinopathy) (Type 2 diabetes, symptoms, Excessive thirst), (Type 2 diabetes, symptoms, Polyphagia)

Output: There are many causes of diabetes, including cataract, arsenic neuritis, and diabetic retinopathy. Type 2 diabetes is characterized by symptoms such as excessive thirst and polyphagia.