Real-Time Validation of ChatGPT facts using RDF Knowledge Graphs

Michalis Mountantonakis[1,2,*], Yannis Tzitzikas[1,2]

[1]Institute of Computer Science, FORTH, Heraklion, Greece
[2]Department of Computer Science, University of Crete, Heraklion, Greece

Abstract
ChatGPT is an innovative application of Large Language Models (LLMs) that produces detailed and articulate responses across many domains of knowledge. However, it does not provide evidence for its responses, and it returns several erroneous facts, even for popular persons, places and others. For tackling the mentioned limitation, we present the fact checking service of the research prototype GPT•LODS, which can validate ChatGPT facts by using RDF Knowledge Graphs (KGs) containing high quality structured data. Indeed, GPT•LODS is able to generate triples for a question, an entity or a given text using ChatGPT. Afterwards, it can validate at real-time the generated ChatGPT triples through DBpedia or LODsyndesis KG (a KG that has indexed 400 other RDF KGs), by combining SPARQL queries, word embeddings and sentence similarity metrics. We present the functionality and use cases of GPT•LODS, including fact checking, question answering, triples generation from text and comparison of different GPT models.

Demo URL: https://demos.isl.ics.forth.gr/GPToLODS/FactChecking
Demo Video: https://youtu.be/5DW1d37aPMc

Keywords
Fact Checking, Validation, Provenance, ChatGPT, Knowledge Graphs, Embeddings, LODsyndesis

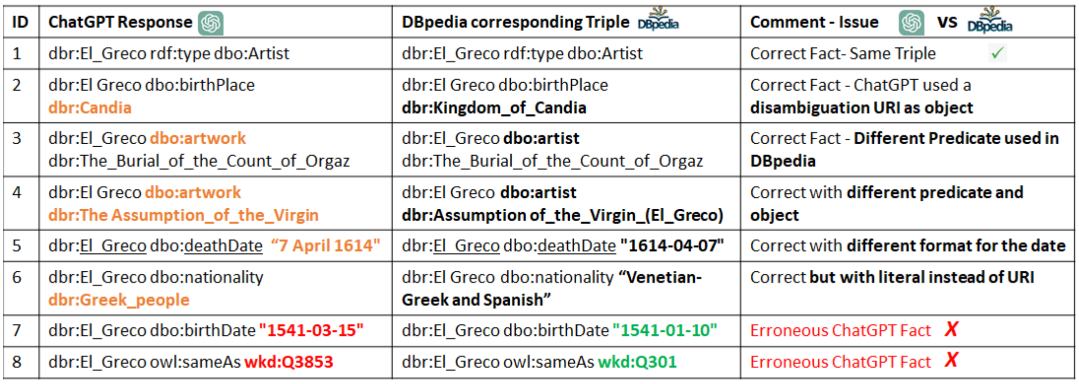
1. Introduction

ChatGPT is a novel Artificial Intelligence (AI) chatbox (https://openai.com/), which is based on Large Language Models (LLMs), and offers detailed responses across many domains. However, it does not provide justifications for the responses, and can return erroneous and outdated facts, even for popular places, persons and other entities. The mentioned limitation can be assisted by using RDF Knowledge Graphs (KGs) containing high quality structured data. Indeed, there have been already proposed approaches that combine RDF KGs with ChatGPT, i.e., for summarizing RDF KGs [1], for entity matching [2], for question answering [3] and for annotation [4]. In this demo we focus on providing a service for validating ChatGPT facts by using multiple RDF KGs.

In particular, it seems that ChatGPT can produce valid RDF N-triples for a given model, e.g., DBpedia ontology [5], i.e., see Fig. 1. In that real example, we asked ChatGPT for facts for the famous greek painter El Greco (in RDF N-triples format using DBpedia model). The left side

ISWC 2023 Posters and Demos: 22nd International Semantic Web Conference, November 6-10, 2023, Athens, Greece
mountant@ics.forth.gr (M. Mountantonakis); tzitzik@ics.forth.gr (Y. Tzitzikas)
https://users.ics.forth.gr/~mountant/ (M. Mountantonakis); https://users.ics.forth.gr/~tzitzik/ (Y. Tzitzikas)
0000-0002-1951-0241 (M. Mountantonakis); 0000-0001-8847-2130 (Y. Tzitzikas)

CEUR Workshop Proceedings (CEUR-WS.org)



ID	ChatGPT Response	DBpedia corresponding Triple	Comment - Issue VS
1	dbr:El_Greco rdf:type dbo:Artist	dbr:El_Greco rdf:type dbo:Artist	Correct Fact- Same Triple ✓
2	dbr:El Greco dbo:birthPlace **dbr:Candia**	dbr:El Greco dbo:birthPlace **dbr:Kingdom_of_Candia**	Correct Fact - ChatGPT used a **disambiguation URI as object**
3	dbr:El_Greco **dbo:artwork** dbr:The_Burial_of_the_Count_of_Orgaz	dbr:El_Greco **dbo:artist** dbr:The_Burial_of_the_Count_of_Orgaz	Correct Fact - **Different Predicate used in DBpedia**
4	dbr:El Greco **dbo:artwork** **dbr:The Assumption_of_the_Virgin**	dbr:El Greco **dbo:artist** **dbr:Assumption of_the_Virgin_(El_Greco)**	Correct with **different predicate and object**
5	dbr:El_Greco dbo:deathDate **"7 April 1614"**	dbr:El_Greco dbo:deathDate **"1614-04-07"**	Correct with **different format for the date**
6	dbr:El_Greco dbo:nationality **dbr:Greek_people**	dbr:El Greco dbo:nationality **"Venetian-Greek and Spanish"**	Correct **but with literal instead of URI**
7	dbr:El_Greco dbo:birthDate **"1541-03-15"**	dbr:El_Greco dbo:birthDate **"1541-01-10"**	Erroneous ChatGPT Fact ✗
8	dbr:El_Greco owl:sameAs **wkd:Q3853**	dbr:El_Greco owl:sameAs **wkd:Q301**	Erroneous ChatGPT Fact ✗

Figure 1: ChatGPT Key Issues related to RDF KGs based on a real ChatGPT response for "El Greco".

shows the ChatGPT response, the middle side the corresponding triple in DBpedia and the right side a comparison between them. The ideal scenario is the same triple to be part of the KG (see ID 1), e.g., DBpedia, however, this is not always the case even for the correct facts. In particular, we can see correct ChatGPT facts with i) wrong/invalid URIs, e.g., disambiguation URIs, wrong predicates, unknown URIs (see IDs 2-4) or/and ii) problems with literals, e.g., different formats (see ID 5) or URIs instead of literals (see ID 6). On the contrary, there can be several erroneous facts even for popular entities, including both wrong literals and URIs, e.g., see IDs 7-8 in Fig. 1.

For enabling the validation of all these cases, we demonstrate the GPT•LODS fact checking service, which generates RDF N-triples from ChatGPT and validates the generated ChatGPT facts by combining SPARQL queries, word embeddings and sentence similarity metrics. For increasing the possibility to find the desired fact in an RDF KG, we also use LODsyndesis KG (https://demos.isl.ics.forth.gr/lodsyndesis), which has integrated data from 400 RDF KGs, by taking into account the transitive closure of their equivalence relationships. To the best of our knowledge, this is the first service that validates ChatGPT facts using one or more RDF KGs.

The rest of this demo paper is organized as follows: §2 introduces the related work, §3 describes the steps and the functionality of the Fact Checking Service of GPT•LODS, §4 presents the use cases and §5 concludes the paper and discusses future work.

2. Related Work

Concerning the related approaches, in [3] the authors used ChatGPT for the Question Answering task, and they concluded that ChatGPT offered high precision answers for popular domains, but low precision for unpopular ones. Moreover, ChatGPT has been used for entity matching [2], for named entity identification tasks over historical data [6] and for summarizing RDF KGs by using a ChatGPT constructed classifier [1]. Moreover, in our previous work [4], we have used multiple RDF KGs for annotating ChatGPT responses by using popular Entity Recognition tools, and each entity is enriched with more statistics and links to LODsyndesis [7].

Concerning the novelty, to the best of our knowledge, this is the first service that tries to validate ChatGPT facts by using RDF KGs and techniques based on word embeddings.

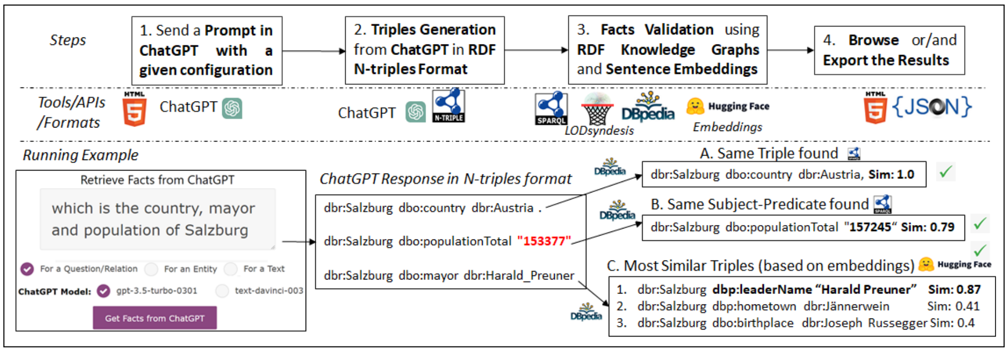

Steps
1. Send a Prompt in ChatGPT with a given configuration
2. Triples Generation from ChatGPT in RDF N-triples Format
3. Facts Validation using RDF Knowledge Graphs and Sentence Embeddings
4. Browse or/and Export the Results
Tools/APIs /Formats
ChatGPT
ChatGPT
LODsyndesis
Embeddings
{JSON}
Running Example
Retrieve Facts from ChatGPT
which is the country, mayor and population of Salzburg
● For a Question/Relation ○ For an Entity ○ For a Text
ChatGPT Model: ● gpt-3.5-turbo-0301 ○ text-davinci-003
Get Facts from ChatGPT
ChatGPT Response in N-triples format
dbr:Salzburg dbo:country dbr:Austria .
dbr:Salzburg dbo:populationTotal "153377".
dbr:Salzburg dbo:mayor dbr:Harald_Preuner
A. Same Triple found
dbr:Salzburg dbo:country dbr:Austria, Sim: 1.0 ✓
B. Same Subject-Predicate found
dbr:Salzburg dbo:populationTotal "157245“ Sim: 0.79 ✓
✓
C. Most Similar Triples (based on embeddings) Hugging Face
1. dbr:Salzburg dbp:leaderName “Harald Preuner” Sim: 0.87
2. dbr:Salzburg dbp:hometown dbr:Jännerwein Sim: 0.41
3. dbr:Salzburg dbo:birthplace dbr:Joseph Russegger Sim: 0.4

Figure 2: The steps of the Fact Checking Service of GPT•LODS and a running example about “Salzburg"

3. Steps and Functionality of the ChatGPT Fact Checking Service

Here, we present the steps and the functionality, which are analyzed below and are depicted in Fig. 2 with a running example about the city of “Salzburg" (see the lower left part of Fig. 2). The steps are generic and can be easily adjusted for performing fact checking for the response of any LLM (e.g., a new version of ChatGPT or any existing/new LLM offering an analogous API).

Step 1. Send a Prompt to ChatGPT. The user can type a i) question, ii) an entity or iii) a text, and the user selects a ChatGPT model to use (i.e., “text-davinci-003" or “gpt-3.5-turbo-0301"). According to the input type, a different query is sent to ChatGPT, e.g., for the entity case, the query to ChatGPT follows: “Give me K RDF N-triples using DBpedia format for the entity E", where K is the number of facts and E is the entity name (both K and E are given by the user).

Step 2. Triples Generation from ChatGPT. We decided to use ChatGPT for generating the triples, since ChatGPT can usually create dereferencable DBpedia URIs and valid RDF N-triples. Concerning this step, GPT•LODS collects the ChatGPT response and keeps only the RDF N-triples (since sometimes ChatGPT also returns a text with the RDF N-triples), which are shown to the user in an HTML table. Additionally, the user can export the generated triples.

Step 3. Facts Validation using RDF Knowledge Graphs and Word Embeddings. The user selects which KG to use for validating the generated ChatGPT facts; DBpedia [5] or LODsyndesis [7]. Concerning LODsyndesis, it contains 2 billion facts from 400 RDF KGs, by having precomputed and stored the transitive closure of owl:sameAs, owl:equivalentProperty and owl:equivalentClass relationships, and all the available facts for each real entity in the same index entry. By using DBpedia, SPARQL queries are sent for retrieving the desired data, whereas for LODsyndesis, we use its REST API. The user can select to validate either a single fact or all the facts. Concerning the validation, it is based on 3 rules: A) Same/Equivalent Triple, B) Same Subject-Predicate or Subject-Object and C) Most Similar Triples.

Concerning rule A, we search in the KG if the same or an equivalent (for LODsyndesis KG) triple exists, and in such a case it returns the triple (and its provenance) to the user, e.g., see rule A in the right side of Fig. 2. Regarding rule B, we search if the same subject-predicate or the same subject-object exists in the KG, and in such a case we return the results in descending order according to their similarity score with the ChatGPT fact, e.g., see rule B in Fig. 2, where we found the correct population of Salzburg (i.e., same subject-predicate but different object). Concerning rule C (executed if the previous two failed), we collect all the triples containing

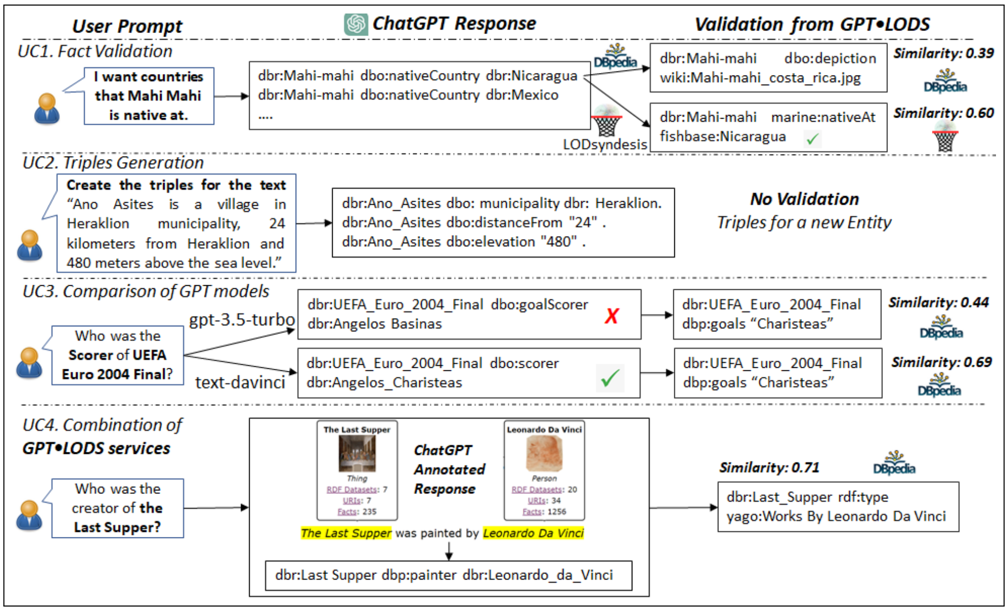

User Prompt
ChatGPT Response
Validation from GPT•LODS
UC1. Fact Validation
I want countries that Mahi Mahi is native at.
dbr:Mahi-mahi dbo:nativeCountry dbr:Nicaragua
dbr:Mahi-mahi dbo:nativeCountry dbr:Mexico
....
DBpedia
LODsyndesis
dbr:Mahi-mahi dbo:depiction wiki:Mahi-mahi_costa_rica.jpg
dbr:Mahi-mahi marine:nativeAt fishbase:Nicaragua ✓
Similarity: 0.39
Similarity: 0.60
UC2. Triples Generation
Create the triples for the text "Ano Asites is a village in Heraklion municipality, 24 kilometers from Heraklion and 480 meters above the sea level."
dbr:Ano_Asites dbo: municipality dbr: Heraklion.
dbr:Ano_Asites dbo:distanceFrom "24" .
dbr:Ano_Asites dbo:elevation "480" .
No Validation
Triples for a new Entity
UC3. Comparison of GPT models
Who was the Scorer of UEFA Euro 2004 Final?
gpt-3.5-turbo
text-davinci
dbr:UEFA_Euro_2004_Final dbo:goalScorer dbr:Angelos Basinas X
dbr:UEFA_Euro_2004_Final dbo:scorer dbr:Angelos_Charisteas ✓
dbr:UEFA_Euro_2004_Final dbp:goals "Charisteas"
Similarity: 0.44
Similarity: 0.69
UC4. Combination of GPT•LODS services
Who was the creator of the Last Supper?
The Last Supper
Thing
RDF Datasets: 7
URIs: 7
Facts: 235
ChatGPT Annotated Response
Leonardo Da Vinci
Person
RDF Datasets: 20
URIs: 34
Facts: 1256
The Last Supper was painted by Leonardo Da Vinci
dbr:Last Supper dbp:painter dbr:Leonardo_da_Vinci
Similarity: 0.71
dbr:Last_Supper rdf:type yago:Works By Leonardo Da Vinci

Figure 3: Use Cases with real examples from the online Fact Checking Service of GPT•LODS

the main entity of the triple (e.g., Salzburg) and we create the embeddings for each of these triples and the desired ChatGPT fact, by using a sentence similarity library from Hugging Face (https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2). Finally we return the top-K similar triples according to the cosine similarity of their vector with the vector of the given ChatGPT fact. In Fig. 2 (lower right side), DBpedia uses a different property for the "mayor" and a literal instead of URI for "Harald Preuner", however, GPT•LODS managed to verify the fact.

Step 4. Browse/Export the Results. The user can browse (through HTML tables) for each ChatGPT fact the corresponding triple(s) in the KG with their provenance, the possible differences between the ChatGPT fact and the triple(s) from the KG, and the cosine similarity score of their vectors. Apart from the HTML tables, one can export the results in JSON format.

Code & Evaluation. In https://github.com/mountanton/GPToLODS_FactChecking, one can browse the code and a preliminary evaluation where the results are promising; for 1,000 manually-labelled ChatGPT facts for famous Greek persons (including 812 correct and 188 erroneous facts), by using the above steps and LODsyndesis, we verified the 92.2% of the correct ChatGPT facts and we found the correct answer for the 57.1% of the erroneous ChatGPT facts.

4. Use Cases & Demonstration

We present four use cases, where the fact checking service of GPT•LODS can be useful, i.e., UC1-UC4, which are also presented in the following tutorial video: https://youtu.be/5DW1d37aPMc. Finally, the demo webpage is available in https://demos.isl.ics.forth.gr/GPToLODS/FactChecking.

UC1. Fact Validation & Question Answering from Multiple KGs. The user can verify ChatGPT facts from one or more RDF KGs, i.e., for confirming correct ChatGPT facts or/and for finding the correct answer for erroneous ChatGPT facts. The presented process can be also exploited for Question Answering applications, e.g., see the examples of Figures 1 and 2.

Moreover, through LODsyndesis more facts can be verified even for the same entity, e.g, in the UC1 of Fig. 3 the ChatGPT fact verified by using LODsyndesis (specifically from the Fishbase KG) and not from DBpedia (which does not contain the native countries of the fish Mahi-mahi).

UC2. Triples Generation. The user can generate (and export) RDF triples either for well known entities and facts or for any given text by exploiting ChatGPT. In the first case they can be possibly used for KG completion (creating new triples for existing KG entities). In the second one, the triples can be used for KG generation, e.g., in Fig. 3 (see UC2), GPT●LODS generated triples for a given text about a village in Crete, for which a DBpedia page does not exist.

UC3. Comparison of GPT models. One can use different GPT models, for comparing the validity of answers for the same questions. For instance, Fig. 3 shows a real example, where the "text-daVinci" model provided the correct answer for the user question, whereas "gpt-3.5-turbo" provided an erroneous one (although it is newer comparing to daVinci). In both cases, GPT●LODS found the correct answer for the desired fact (however with a different similarity score).

UC4. Combination of GPT●LODS services - Annotation and Fact Checking. GPT●LODS also offers a service [4] which can annotate ChatGPT textual responses with links to LODsyndesis and DBpedia (i.e., for information enrichment). The latter service is also connected with the presented fact checking service; specifically the generated ChatGPT annotated textual response can be converted to RDF triples (using ChatGPT) and the triples are validated using the presented steps. In such a case two requests are sent to ChatGPT, e.g., see UC4 in Fig. 3.

5. Conclusion

In this paper, we demonstrated the fact checking service of GPT●LODS, which exploits SPARQL queries and sentence similarity techniques (based on word embeddings), for validating at real time any ChatGPT response from multiple RDF Knowledge Graphs. We presented all the steps and use cases of GPT●LODS, including fact validation, triples generation and others. As a future work, we plan to extend the service i) for providing a REST API and ii) for adding more features, e.g., providing the service as a chat-box and exploiting knowledge from web search engines.

Acknowledgments. This work has received funding from the European Union's Horizon 2020 coordination and support action 4CH (Grant agreement No 101004468).

References

[1] G. Vassiliou, et al., SummaryGPT: Leveraging ChatGPT for summarizing knowledge graphs, ESCW (2023).

[2] R. Peeters, C. Bizer, Using ChatGPT for entity matching, arXiv preprint arXiv:2305.03423 (2023).

[3] R. Omar, O. Mangukiya, P. Kalnis, E. Mansour, ChatGPT versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots, arXiv:2302.06466 (2023).

[4] M. Mountantonakis, Y. Tzitzikas, Using multiple RDF knowledge graphs for enriching ChatGPT responses, in: ECML/PKDD 2023 Demo paper, 2023.

[5] J. Lehmann, et al., DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semantic web 6 (2015) 167–195.

[6] C.-E. González-Gallardo, E. Boros, N. Girdhar, A. Hamdi, J. G. Moreno, A. Doucet, Yes but.. can ChatGPT identify entities in historical documents?, arXiv preprint arXiv:2303.17322 (2023).

[7] M. Mountantonakis, Y. Tzitzikas, Content-based union and complement metrics for dataset search over RDF knowledge graphs, ACM JDIQ 12 (2020) 1–31.