Truth-O-Meter: Collaborating with LLM in Fighting its Hallucinations

Boris Galitsky
Knowledge Trail Inc. San Jose CA US

Abstract

A text obtained by a Large Language Model (LLM) such as GPT4 usually has issues in terms of incorrectness and hallucinations. We build a fact-checking system 'Truth-O-Meter' which identifies wrong facts, comparing the generation results with the web and other sources of information, and suggests corrections. Text mining and web mining techniques are leveraged to identify correct corresponding sentences; also, the syntactic and semantic generalization procedure adopted to the content improvement task. To handle inconsistent sources while fact-checking, we rely on an argumentation analysis in the form of defeasible logic programming. We compare our fact checking engine with competitive approach based on reinforcement learning on top of LLM or token-based hallucination detection. It is observed that LLM content can be substantially improved for factual correctness and meaningfulness.
https://github.com/bgalitsky/Truth-O-Meter-Making-ChatGPT-Truthful

1. Introduction

We propose a novel fact checking framework for answering single/multi-hop questions across heterogeneous knowledge sources. The key novelty of our method is the introduction of the intermediary fact-checking modules into the current retriever-reader pipeline. Unlike previous methods that solely rely on the retriever for gathering all evidence in isolation, our intermediary fact-checker performs a chain of reasoning over the retrieved set. Our approach links the retrieved evidence with its related global context into graphs and organizes them into a candidate list of evidence chains.

Large Language Models (LLMs), including GPT-3 (Brown et al., 2020) and ChatGPT, have exhibited remarkable proficiency in producing natural language texts that are fluent, coherent, and informative.

The prevailing belief is that the great performance of these models is attributable to the vast amount of world knowledge encoded within them and their ability to generalize from that knowledge.
GPT 4, via its scale and reinforcement learning with human feedback, has gained significant attention. Fluent and comprehensive answers over a broad range of topics surpass prior public chatbots in terms of usefulness.

However, despite its powerful capabilities, several studies have consistently shown significant issues, including errors in terms of factuality, reasoning, math, coding, biases, etc. (Bang et al., 2023; Borji, 2023; Guiven, 2023). the encoding of knowledge in LLMs is not perfect, and the generalization of knowledge could cause "memory distortion." This can result in these models generating false information or "hallucinating”.

A majority of approaches to verification and correction of an LLM results identify relevant knowledge which is expected to improve the correctness of result and supply it to the LLM. However, it is frequently beneficial to do all fact-checking processes outside of LLM to assure no hallucination occurs again.

In some verification and correction sessions, coming back to LLM with relevant attributes knowledge and re-running the request helps. In other cases, LLM hallucinates again, and the knowledge session does not converge to the truthful and relevant results. We explore how to combine fact-checking outside of

LLM with the one re-iteration. We refer to the former as iterative fact checking. Along with the boost of truthfulness, we attempt to minimize the amount of calls to LLM API to limit the costs of content creation.

We tackle the hallucination detection problem from two directions. One is an improvement in selecting phrases which need fact-checking (Galitsky 2022). Too short or opinionated phrases do not need fact-checking.

Coming from the other direction, we handle multiple sources of information on the topic with potential conflict, such as inconsistent web search results. This is unlike the majority of hallucination detection studies we cite in this paper which assume that a single ground truth is available. Our intention is to identify and correct factual errors in the generated text. This work is expected to improve the overall factuality of LLMs.

Our contribution:

- Fact checking via web mining adjusted to LLM content versus human-made intentional misinformation
- Applicability to texts of various genres and mixture of formats and genres in a document
- Application to the whole document with opinionated data, background, captions, results, design notes etc.
- Finding proper level of granularity for fact-checking: from phrases to sentences, with contextualization
- Verification against various sources of different modalities with mutual inconsistency
- Verification against abstract canons, generalizations, rules, principles

1.1 Why LLMs hallucinate

It is not always clear in which case LLMs hallucinate. Hallucination concept is only applicable to factual and not opinionated texts. Hence we work with genres where factual correctness in essential. In fiction, poetry, lyrics, and stories LLMs have limited memorization for less frequent entities, and are prone to hallucinate on long-tail knowledge (Kandpal et al., 2022; Mallen et al., 2022). Therefore, we might be able to ask questions involving long-tail knowledge.

Unstructured factual knowledge typically exists at a document level (i.e., a group of factual sentences about an entity). This means that sentences can contain pronouns (e.g., she, he, it), making these sentences factually useless standalone. To illustrate with an example from Suvorov Wikipedia page, “He secured Russia's expanded borders and renewed military prestige and left a legacy of theories on warfare.” cannot be a useful standalone fact because it is unclear who “He” is. Due to the GPU memory limit and computation efficiency, it is common to chunk documents in LLM training corpus. This causes the “fragmentation” of information and leads to wrong associations of entities that appear in independent documents with similar contexts.

(Zhow et al 2023) explores characteristics of LLM-generated misinformation compared with content written by humans and then evaluates the applicability of existing solutions. The authors compiled human-created COVID-19 misinformation and abstracted it into narrative prompts for a language model to output LLM - misinformation. Significant linguistic differences are identified within human-LLM pairs, and patterns of misinformation in enhancing details, communicating uncertainties, drawing conclusions, and simulating personal tones are revealed. While existing models remained capable of classifying LLM-misinformation, a significant performance drop compared to misinformation in human communication was observed. Results suggested that existing information assessment guidelines had questionable applicability, as LLM-misinformation tends to meet criteria in evidence credibility, source transparency, and limitation acknowledgment. The authors draw implications for practitioners in content creating, researchers, and journalists, as LLM can create new challenges to the societal problem of misinformation.

1.2 Introductory Example

We show a query, ChatGPT response and step-by-step hallucination detection and correction procedure.

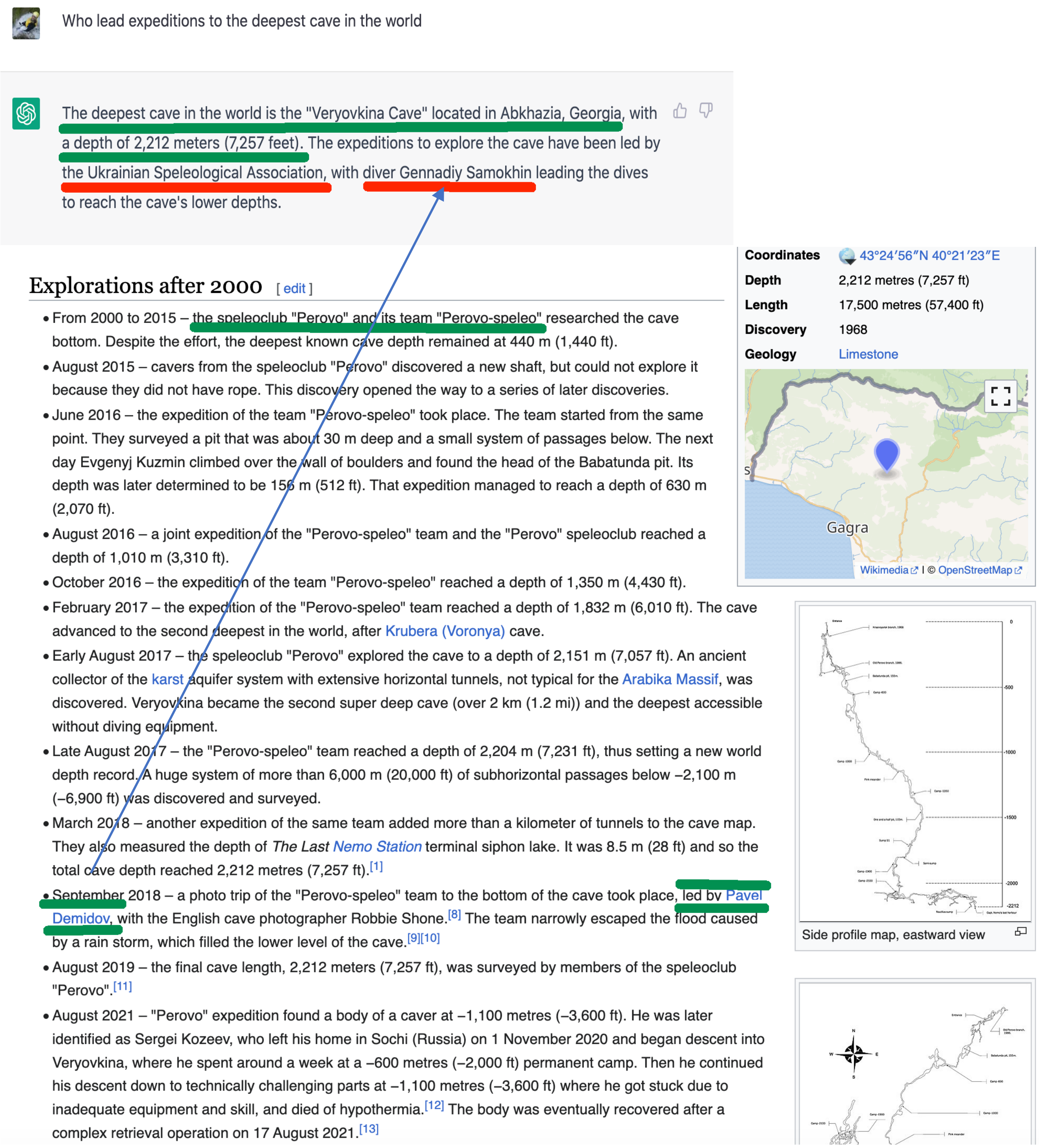


Fig. 1. An example of hallucination detection

There is a following list of steps which need to be taken to perform fact-checking and correction:

1) Fact – check LLM sentence by sentence and phrase (with attribute) by phrase
2) Start with the first sentence. Extract the main entity ‘Veryovkina cave’. Confirm that it is the deepest cave: we search the web for it and find a confirmation on Wikipedia.

3) Proceed to the second sentence. Extracting ‘led by<entity>’ and ‘ with <entity> leading the dives’ phrases. Both phrases are neither confirmed by the web nor by the Wikipedia page identified by the first sentence.
4) Coming to conclusion that the second sentence is totally wrong (is a complete hallucination). The hallucinated entities for persons and teams are real but associated with a different geo entity (cave).
5) It turns out that correct entities needs to be extracted from Wiki page identified by the query derived from the first sentence.
6) We use a retriever to obtain the value for ‘Expedition leader’ and substitute it into the second LLM sentence to correct it. Both organization name and individual (leader) name needs to be substituted.

Hence we will be identifying and repairing hallucination, moving in two directions:
1) Consolidating evidence from external knowledge for the LLM to generate responses grounded in evidence. LLM is considered as a black box and is called only once.
2) Revising LLM’s (candidate) responses using automated feedback (Peng et al 2023). LLM is called iteratively, each time with additional information (Section 3).

Hence for the LLM’s candidate response:

The deepest cave in the world is the "Veryovkina Cave" located in Abkhazia, Georgia, with a depth of 2,212 meters (7,257 feet). The expeditions to explore the cave have been led by the Ukrainian Speleological Association, with diver Gennadiy Samokhin leading the dives to reach the cave's lower depths.

We obtain the revised response where hallucination is identified and substituted

The deepest cave in the world is the "Veryovkina Cave" located in Abkhazia, Georgia, with a depth of 2,212 meters (7,257 feet). The expeditions to explore the cave have been led by "Perovo-speleo" team lead by Pavel Demidov and Piotr Luibimov” (citing https://en.wikipedia.org/wiki/Veryovkina_Cave)

2. Related Work

Existing work including the current study usually attempts to detect hallucinations based on a corresponding oracle reference at a phrase, sentence or paragraph level. However, ground-truth references may not be readily available for many free-form text generation applications, and sentence- or document- level detection may fail to provide the fine-grained signals that would prevent fallacious content in generation time. (Liu et al 2022) formulate a novel *token-level, reference-free* hallucination detection task.

Hallucination detection is important beyond content generation domain. Zhou et al 2020 formulate a prediction setting where each token in the output sequence is predicted to be hallucinating conditioned on the source input, and collect new manually annotated evaluation sets for this task. We also introduce a novel method for learning to model hallucination detection, based on pretrained language models fine tuned on synthetic data that includes automatically inserted hallucinations. Experiments on machine translation and abstract text summarization demonstrate the effectiveness of our proposed approach -- we obtain an average F1 of around 0.6 across all the benchmark datasets and achieve significant improvements in sentence-level hallucination scoring compared to baseline methods.

Reference-based hallucination detection has been proposed for abstractive summarization (Maynez et al., 2020), machine translation (Wang and Sennrich, 2020), and data- to-text generation (Rebuffel et al., 2021).

Popular sampling algorithms (e.g., top-p) in open-ended text generation can harm the factuality due to the "uniform randomness" introduced at every sampling step. (Lee at al 2022) propose the factual-nucleus sampling algorithm that dynamically adapts the randomness to improve the factuality of generation while maintaining quality. The authors analyze the inefficiencies of the standard training method in learning correct associations between entities from factual text corpus such as Wikipedia. We propose a factuality-enhanced training method is proposed that uses *topic-prefix* for better awareness of facts and sentence completion as the training objective, which can vastly reduce the factual errors.

(Lee at al 2022) explore training methods that can effectively leverage text corpus with rich facts. It turns out that directly continuing the training of LM on factual text data (Gururangan et al 2020) does not guarantee the improvement of factual accuracy. The authors propose factuality-enhanced training to address the underlying inefficiencies of this baseline. The authors improve the awareness of facts during training, and also formulate a sentence completion task as the new objective for continued LLM training .

On the contrary, in this work we attempt to distance ourselves from a learning-based approach to factuality and take a knowledge-based/reference approach.

Drawing inspiration from human behavior, Gou et al. (2023) have proposed a framework that allows Language Model Models (LLMs) to validate and refine their own outputs using external tools, similar to how humans employ tools for fact-checking or debugging. Traditionally considered as "black boxes," LLMs can now engage in a process of self-validation and progressive refinement. The framework begins with an initial output generated by the LLM. The system then interacts with appropriate tools to assess specific aspects of the generated text. Based on the evaluation provided by these tools, the output is subsequently revised, leading to an improved version.

Similarly, Peng et al. (2023) have developed a system that incorporates external knowledge into LLM-generated responses. This involves utilizing task-specific databases to ground the responses in relevant information. Furthermore, the system iteratively modifies the prompts given to the LLM, aiming to enhance the quality of the model's responses. Feedback generated by utility functions, such as the factuality score of a response, plays a crucial role in guiding this iterative refinement process. We refer to these approaches as *Iterative* and re-implement them to fit the Truth-O-Meter framework.

3. Iterative mode

Verification and correction can be done outside of an LLM. However, once relevant background knowledge is identified, it can be given to LLM and a request to incorporate it to revise the previous answer issued. This is the promising scenario we will be exploring in this section. Let us continue our example of verification session and now apply iterations to it.

1) Given a user query, Truth-O-Meter first retrieves evidence from external knowledge (e.g., Web or task-specific datasets) and, if necessary, further fuses evidence by linking obtained raw evidence with related context (e.g., information of the entity "**Veryovkina Cave**") and performing reasoning to form evidence chains (e.g., extracting information from list of references on a Wiki page).
2) Then, Truth-O-Meter queries ChatGPT using a prompt that contains the consolidated evidence for ChatGPT to generate a candidate response grounded in external knowledge (evidence).
3) Truth-O-Meter verifies the candidate response e.g., by checking whether it is inconsistent with evidence. If so, Truth-O-Meter generates a feedback message (e.g., about the caving team). The message is used to revise the prompt to query ChatGPT again.
4) The process iterates until a candidate response passes the verification and is sent to the user.

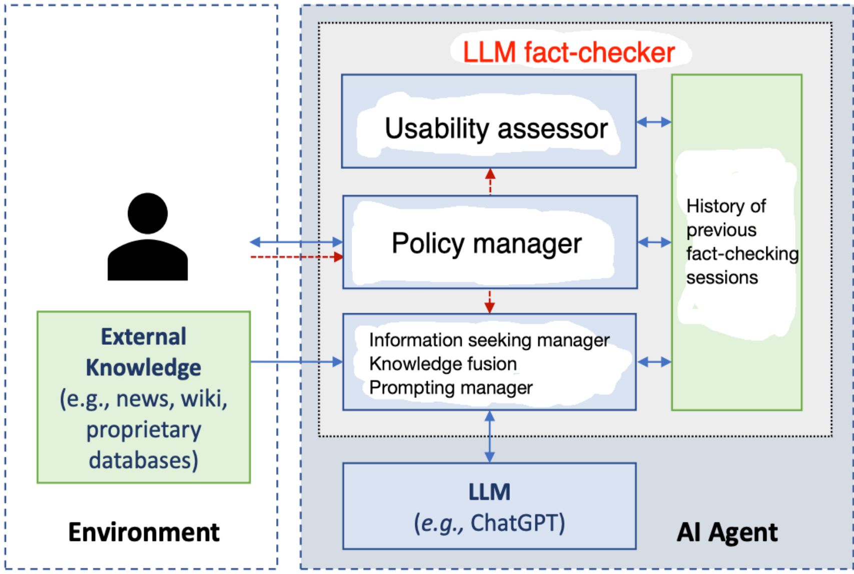


Fig. 2 Information flow for iterative Truth-O-Meter

Blue arrows denote data flow, and red arrows – content update

Truth-O-Meter reward *R* is determined as the inverse frequency of wrong facts returned to the user.

Policy manager selects the next action of Truth-O-Meter that leads to the best expected reward *R*. The These actions include:

1) acquiring evidence e for q from external knowledge;
2) calling the LLM to generate a current candidate response, and
3) sending a response to users if all available knowledge has been applied.

Truth-O-Meter's policy combines manually crafted rules, and training results on actual or imitated human-system interactions. For example, (Peng et al. 2023) implement a trainable policy π as a neural network model maximize the expected reward as θ:

$$\underset{\theta}{\operatorname{argmax}}\ E_{s \sim S; a \sim \pi_q}\ [R(s; a)]$$

Where: *S* is a set of epistemic states, including dialog history, user query, evidence, phrase and attribute/value substitutions, candidate/intermediate response;
A is a set of actions that Policy manager selects, including Information seeking manager to consult evidence from external knowledge and calling Prompting manager to query the LLM again to proceed with next iteration.
R(*s*; *a*) is the external reward received after taking action *a* in state *s*, which is provided by the environment (e.g., users or simulators);
π_q is computed using a logic programming formalism of reasoning about actions, computing possible actions and their resultant states iteratively (Levesque et al 1997, Galitsky 2006).

To maintain generality, we do not involve task-specific retrievers from Information seeking manager that may lead to an elevated assessment of performance. Instead, we employ the combined Google and Bing Search API to search queries generated by LLMs, scrape the resulting top-five HTML web pages, and extract a candidate answer by fuzzy-matching the snippet from Google. The maximum number of interactions with search engine is set to 5. Chain-of-thought prompting (Wei et al 2022) is used to produce an initial answer, and then up to three iterations are being corrected, stopping early if the answer remains the same for two consecutive corrections. We consider the plausibility and truthfulness are assessed with respect to R during our verification.

Iterations in Truth-O-Meter are defined as follows. Given input x and previous output o_i, LLMs interact with external tools to generate a possible defeat for o_i which we denote d_i. The task-specific defeat can be used to detail the attributes of the output we expect to evaluate, such as truthfulness, feasibility, appropriateness or safety. For different inputs, task-dependent, heuristically selected, or automatically selected appropriate tools can be chosen for verification.

LLMs can produce an enhanced answer conditioned on input x, previous output o_i, and defeat d_i from verification. Defeat generation plays a crucial role in the correction process as they spot errors, offer actionable suggestions, or identify plausible groundings through interaction with various knowledge sources, thus maintaining a new generation with the purpose to avoid errors it has already done. Motivated by the human process of iterative drafts refinement, we can iterate the process of verify-then-correct until specific stopping criteria are met, such as satisfying critiques from verification, reaching the maximum iterations n, or receiving
environmental feedback. This method facilitates continuous output improvement by systematically and sample-efficiently verifying and correcting errors resulting from interactions with the world.

Let us look at iterative hallucination detection and verification process:
We start with a question

Can patients with a single kidney take allopurinol to decrease uric acid level?

In the case of patients with a single kidney, it is important to consider their renal function before prescribing any medications. Allopurinol is primarily eliminated from the body through the kidneys, so if someone has reduced kidney function, the dosage of allopurinol may need to be adjusted.

…

their renal function before prescribing any medications. Allopurinol is primarily eliminated from the body through the kidneys, so if

The last sentence is not really informative, as the vast majority of drugs is eliminated from the body through kidney. Therefore, this raises a question, which leads in hallucination. However, it is caused by a meaningless, uninformative sentence above

According to ChatGPT, 'Propranolol is excreted in feces'
However, doing fact-checking on google, we find that it is NOT the case:

Google | Propranolol eliminated through liver

Propranolol is completely absorbed after oral administration and wide distributed throughout tissues. Elimination occurs almost wholly by metabolic transformation in the liver and excretion of the resultant products in the urine.

Hence ChatGPT hallucinated: *Excretion happens via urine via kidney.*
ChatGPT accepts the error but does not state explicitly:

> Apologies for any confusion caused by my previous response. You are correct that propranolol is primarily eliminated by metabolic transformation in the liver, followed by excretion of the resultant products in the urine. The liver plays a significant role in the metabolism of propranolol, and the drug undergoes extensive hepatic metabolism before being eliminated.

Background info here is as follows:

> Are all drugs excreted through the kidneys?
>
> Most drugs, particularly water-soluble drugs and their metabolites, are eliminated largely by the kidneys in urine. Therefore, drug dosing depends largely on kidney function. Some drugs are eliminated by excretion in the bile (a greenish yellow fluid secreted by the liver and stored in the gallbladder).

In summary, the major problem with iterative approach is that it can continue to hallucinate at each step of iterations. Therefore we need to decide for each correction session, if we use iterative approach or not. When we chose an action *a,* one option is to exit iterations and proceed with correction outside of LLM. It happens mostly if repetitive hallucination occurred in the previous iteration, or hallucination phrase is short.

4. Handling multiple mutually inconsistent facts obtained from authoritative sources

Fact verification is a challenging task which requires to retrieve relevant evidence from plain text and use the evidence to verify given claims. Many claims require to simultaneously integrate and reason over several pieces of evidence for verification. However, previous work employs simple models to extract information from evidence without letting evidence communicate with each other, e.g., merely concatenate the evidence for processing. Therefore, these methods are unable to grasp sufficient relational and logical information among the evidence. To alleviate this issue, (Zhou et al. 2019) propose a graph-based evidence aggregating and reasoning framework which enables information to transfer on a fully-connected evidence graph and then utilizes different aggregators to collect multi-evidence information.

(Yin et al 2020) proposed a method for using pre-trained NLI models as a ready-made zero-shot sequence classifiers. The method works by posing the sequence to be classified as the NLI premise and to construct a hypothesis from each candidate label. For example, if we want to evaluate whether a sequence belongs to the class "politics", we could construct a hypothesis of "This text is about politics.". The probabilities for entailment and contradiction are then converted to label probabilities.

A case of the claim that requires integrating multiple evidence to verify is shown in Fig. 3. This figure shows an example in the cases which needs multiple pieces of evidence to make the right inference. The ground truth evidence set contains the sentences from the article "Propranolol" with paragraph number 4 and 7. These two pieces of evidence are also ranked at top two in our retrieved evidence set. The representation for evidence, <Document, PositionInDocument> means the evidence is extracted from the document from a certain position.

Claim: Propranolol is eliminated from the body through routes other than the kidneys
Truth evidence (added in the training set to handle complex cases) **:**
Propranolol

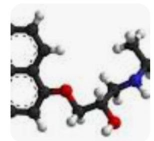
Clinical pharmacokinetics of propranolol

by PA Routledge · 1979 · Cited by 215 — Elimination occurs **almost wholly by metabolic transformation in the liver** and excretion of the resultant products in the urine.

Wikipedia
https://en.wikipedia.org › wiki › Propranolol

Propranolol

Most of the metabolites are **excreted in** the urine. **Propranolol** is a highly lipophilic drug achieving high concentrations **in** the brain. The duration of action of ...



Retrieved evidence:
< Propranolol, 4>, < Propranolol, 7>, …

Evidence:

Mayo Clinic
https://www.mayoclinic.org › drg-20071164

Propranolol (Oral Route) Side Effects

Jun 1, 2023 — This can damage the blood vessels of the brain, heart, and **kidneys**, resulting **in** a stroke, heart failure, or **kidney** failure. Lowering blood ...

Healthline
https://www.healthline.com › propranolol-oral-tablet

Propranolol: Side Effects, Uses, Dosage, and More

Feb 23, 2023 — After you take a single dose of propranolol, **half of the drug will be eliminated from your body in about 6 hours**.

Label: refuted

Fig. 3. Combining multiple evidence for verification

Let us imagine we got the following search results from the web search:

i: The deepest cave was explored by a team from Moscow (correct but too broad)
c: The deepest cave was explored in 1990s by Russians (different cave)
b: It was found by Perovo club from Moscow in 1970s (correct team but wrong year: partially acceptable for correction of content)
a: It was explored to the current bottom by Perovo club (fully correct)
h: Perovo club took part in exploration (correct but not strong enough)

We comment on suitability of each phrase for fact-checking (on the right).

We then apply a Natural Language Inference (NLI) model to determine which of these search results support or defeat which other:

We employ *Facebook/bart-large-mnli* model from HuggingFace and request to assess whether a hypothesis S_i is true (entailment), false (contradiction), or undetermined (neutral) given a premise S_j ($i \neq j$)

As a result, we get the following relations:

c ⇒ i (entails)
b ! ⇒ c (contradicts in years)
h ! ⇒ b (contradicts in years)
a ⇒ b
a ⇒ h

So *f* implies almost everything: how to derive it computationally? We need a machinery to select the best candidate for truth, given all mined facts. A defeasible reasoning approach is selected.

Defeasible logic programs contain intricate relationships within their knowledge, which can complicate the understanding of why certain information is accepted or rejected. During the reasoning process, dialectical trees are constructed to represent the argumentation. These trees, known as dialectical explanations, serve to justify the status of a literal. In the context of Truth-O-Meter, the primary focus lies in how users form mental images of how arguments are evaluated and compared to reach conclusions, which are essentially dialectical explanations.

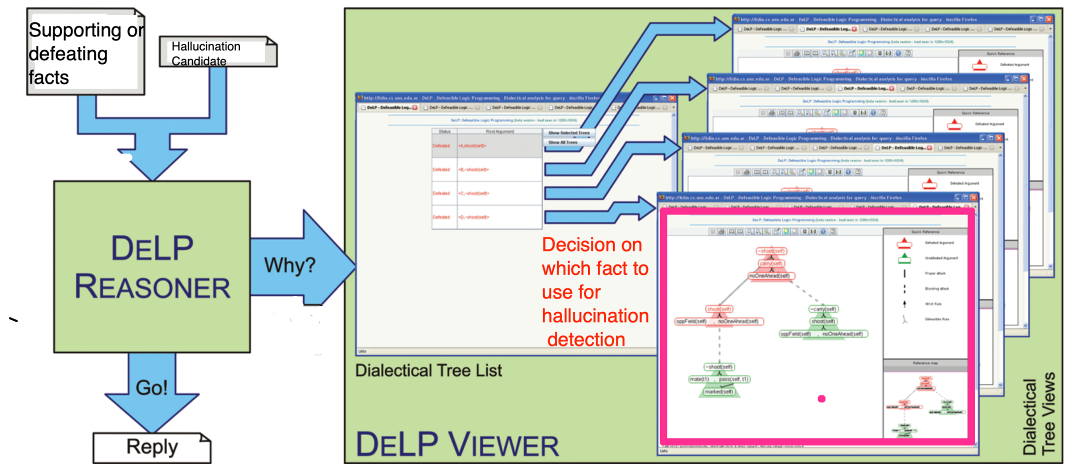


Fig. 4. DeLP processing

The Employed DeLP system, as described by Escarza et al. (2009, Fig. 4), consists of two primary components:

1. The defeasible reasoner, which is responsible for deriving arguments from the defeasible logic program, constructing dialectical trees, and analyzing defeating relationships to respond to user queries.
2. The DeLP Viewer, which serves as the visualization module and the tool presented in the article.

These two modules are interconnected through the dialectical explanation. Using a defeasible logic program based on "true" facts and a query, the DeLP component analyzes the logic program and deduces the most reliable "true" fact. Subsequently, the user can request the system to examine the dialectical explanation by accessing the visualization module. Initially, the DeLP Viewer presents a list of the dialectical trees involved in the reasoning process. The user has the option to choose which dialectical trees they want to visualize. For each selected tree, the system generates an interactive visual representation displayed in a separate window.

To deal with inconsistent rules, the rule engine need to implement a mechanism of rule justification, rejecting certain rules in the situation when other rules have fired. We select Defeasible Logic Programming (DeLP, García and Simari 2004) approach and present an overview of the main concepts associated with it.

A Defeasible logic program is a set of facts, strict rules Π of the form (A:-B), and a set of defeasible rules Δ of the form A- < B, whose intended meaning is “if B is the case, then usually A is also the case”. A DeLP for knowledge sources includes facts which are extracted from search results, and strict and defeasible clauses where the heads and bodies form commonsense reasoning rules.

Let P . (Π, Δ) be a DeLP program and L a ground literal. A defeasible derivation of L from P consists of a finite sequence $L_1, L_2, \ldots, L_n$. L of ground literals, such that each literal L_i is in the sequence because:

1) Li is a fact in Π, or
2) there exists a rule R_i in P (strict or defeasible) with head L_i and body $B_1, B_2, \ldots,$

B_k and every literal of the body is an element L_i of the sequence appearing before L_j ($j < i$).

Let h be a literal, and P . (Π, Δ) a DeLP program. We say that <A, h > is an argument for h, if A is a set of defeasible rules of Δ, such that:

1) there exists a defeasible derivation for h from (Π ∪ A);
2) the set (Π ∪ A) is non-contradictory; and
3) A is minimal: there is no proper subset A0 of A such that A0 satisfies conditions (1) and (2).

Hence an argument <A, h > is a minimal non-contradictory set of defeasible rules, obtained from a defeasible derivation for a given literal h associated with a program P.

We say that <A1, h1> attacks <A2, h2> iff there exists a sub-argument <A, h> of <A2, h2> (A A1) such that h and h1 are inconsistent (i.e. Π [{h, h1} derives complementary literals). We will say that <A1, h1> defeats<A2, h2> if <A1, h1> attacks <A2, h2> at a sub-argument <A, h> and <A1, h1> is strictly preferred (or not comparable to) <A, h>.

In the first case, we will refer to <A1, h1> as a proper defeater, whereas in the second case it will be a blocking defeater. Defeaters are arguments which can be in their turn attacked by other arguments, as is the case in a human dialogue. An argumentation line is a sequence of arguments where each element in a sequence defeats its predecessor. In the case of DeLP, there are a number of acceptability requirements for argumentation lines in order to avoid fallacies (such as circular reasoning by repeating the same argument twice).

Target claims can be considered DeLP queries which are solved in terms of dialectical trees, which subsumes all possible argumentation lines for a given query. The definition of dialectical tree provides us with an algorithmic view for discovering implicit self-attack relations in users' claims. Let <A0, h0> be an argument (target
claim) from a program P.

A dialectical tree for <A0, h0> is defined as follows:

1) The root of the tree is labeled with <A0, h0>
2) Let N be a non-root vertex of the tree labeled <An, hn> and Λ . [<A0, h0>, <A1, h1>, . . ., <An, hn>] (the sequence of labels of the path from the root to N).

Let [<B0, q0>, <B1, q1>, . . ., <Bk, qk>] all attack <An, hn>. For each attacker <Bi, qi> with acceptable argumentation line [Λ,<Bi, qi>], we have an arc between N and its child Ni.

A labeling on the dialectical tree can be then performed as follows:

1) All leaves are to be labeled as U-nodes (undefeated nodes).
2) Any inner node is to be labeled as a U-node whenever all of its associated children nodes are labeled as D-nodes.
3) Any inner node is to be labeled as a D-node whenever at least one of its associated children nodes is labeled as U-node.

For example, for the DeLP

$$\left\{ \begin{array}{llll} a \mathrel{-\!\!\prec} b & \sim b \mathrel{-\!\!\prec} e & \sim b \mathrel{-\!\!\prec} c, f & \sim f \mathrel{-\!\!\prec} i \\ b \mathrel{-\!\!\prec} c & e & f \mathrel{-\!\!\prec} g & i \\ c & \sim f \mathrel{-\!\!\prec} g, h & g & \sim h \mathrel{-\!\!\prec} k \\ \sim b \mathrel{-\!\!\prec} c, d & h \mathrel{-\!\!\prec} j & k & \\ d & j & & \end{array} \right\}$$

we obtain the following dialectical tree (Fig. 5).

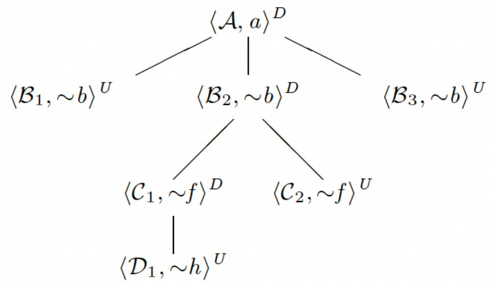


Fig. 5. Dialectical tree for a web mining results for our running example

Notice that facts {a, …, i} and relationships between them are obtained from search results and defeasible commonsense reasoning clauses are obtained from LLM (Li et al 2022). Hence *a*: 'It was explored to the current bottom by Perovo club' **is confirmed** and can be trusted (Fig. 5).

5. Correcting factual errors in syntactic space

For fact checking in domains with a focus on entities, such as product recommendation or ecommerce domains, we rely on syntactic structure. The syntactic match components consist of the following:

1. A set of rules applied to a dependency parse to identify the heads of entity mentions.
2. Rules based on token tags and text to identify the complete mention.
3. A system to link the identified entity mentions to corresponding structured evidence, verifying the factual nature of the text and distinguishing it from hallucinations.

In line with the findings of Estes et al. (2022), we employ phrases from the dependency parse tree originating from key verbs (e.g., "has," "is") and possessives (*poss*) to connect to product attributes. For instance, the direct object (*dobj*) of "has" always represents a product attribute phrase, as illustrated in the example "The table has four legs."

The dependency rules assist in determining the head token of a product or attribute. To identify the complete attribute and, if present, its value, we utilize a set of regular expressions based on part-of-speech tags. For instance, the pattern
/(NN|ADV) ADJ (ADP|PART) (VB|NN) NN? ADP?/.*
helps extract the complete attribute "*hard to shut down."*

Furthermore, in the domain of product recommendation where the statement structure tends to be relatively simple, we leverage two co-reference rules that establish connections between pronouns and product mentions. These rules have proven to be more accurate and faster than co-reference resolution using general-purpose models.

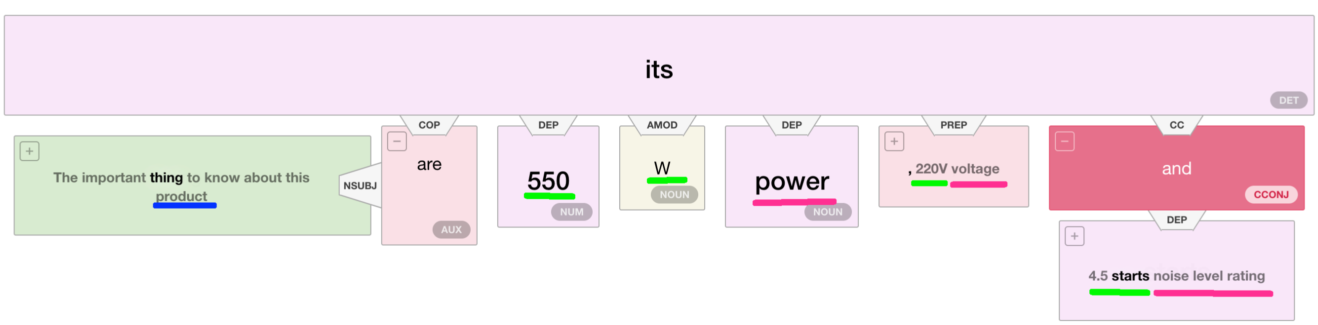


Fig. 5. Syntactic tree representation for a sentence containing products' attribute values

Identifying entity attributes from the dependency parse tree of an input claim whose factuality is to be assessed is shown in Fig. 5. Potential product mentions are shown in blue, attribute names in red, and attribute values in green.

The third component evaluates the extracted hypotheses against the catalog data. Once entities and their attributes are extracted, they can be matched against a product catalogue API such as eBay (https://github.com/timotheus/ebaysdk-python). We focus on the product names, attribute names, and attribute values that are part of our structured product catalog. For each hypothesis extracted from a claim, we iterate through the catalog attributes to find a match. We first check for an exact match. If there is none, we search for catalog and hypothesis attributes whose cosine similarity of their GloVe embeddings (Pennington et al., 2014) is close to 1.0. If any hypothesis fails to match, then the product-attribute claim is False. If all hypotheses evaluate to True, the claim is labeled True.

5.1 Fact-checking by question answering against sources and syntax-semantic alignment

Once we obtained the most similar text T2 assumed to be truthful for a given text T1, we can decide if T2 supports everything in T1. We use three types of approaches:

1) Baseline: we consider tokens which occurs in T1 but do not occur in T2 and therefore unconfirmed (should be rejected);
2) Instead of such tokens we attempt to obtain attribute values which occurs in T1 but do not occur in T2 and therefore unconfirmed (should be rejected). To achieve this, we generate questions and then use a question answering model to obtain answers as entities, attributes and values to find those which occurs in T1 but do not occur in T2 and therefore unconfirmed (should be rejected);
3) Instead of trusting a question-answering model with extracting entities, attributes and values, we do all the work with explicit syntactic and semantic representation. The advantage of this approach is that we obtain candidate substitutions from T2 as a by-product of fact-checking.
3)

Algorithm of semantic/factual similarity for fact-checking:

Input: two texts T1 and T2
Output: phrases or attribute values which are from T1 and should be confirmed by T2 but are not.

Derive questions from each text
$T1 \rightarrow Q1$
$T2 \rightarrow Q2$
Now we gave answers for these questions against original and other text:
We apply a machine reading comprehension model (retriever)
$<Q1, T1> \rightarrow A11$
$<Q2, T2> \rightarrow A22$
$<Q1, T2> \rightarrow A12$
$<Q2, T1> \rightarrow A21$
Next step is to find set-theoretic difference between answer values
$D = (A11 \cup A\ 21 \,/\, A12 \cup A\ 22)$
Return D

We now proceed to the second approach: Syntax-semantic alignment fact checking
Input: two texts T1 and T2
Output: phrases or attribute values which are from T1 and should be confirmed by T2 but are not.
Also: syntax-semantic alignment of two texts with mapping for confirming/defeating syntax-semantic nodes

---

Build syntactic dependency parse trees
T1 → synt(T1)
T2 → synt(T2)
Build Abstract Meaning Representation (AMR) trees
T1 → amr(T1)
T2 → amr(T2)
Build mapping
synt(T1) → amr(T1)
synt(T2) → amr(T2)

Build communicative diagram

$$\begin{array}{ccc} synt(T1) & \to & amr(T1) \\ | & & | \\ synt(T2) & \to & amr(T2) \end{array}$$

Derive confirmation and rejection mappings M
Diagram =

$$\begin{array}{ccc} synt(T1) & \to & amr(T1) \\ | \backslash & & | \backslash \\ synt(T2) & \to & amr(T2) \end{array}$$

Return Diagram. M

Once we have a candidate true text, we apply both algorithms to confirm / reject it, and whichever gives the lowest number of unconfirmed entities or tokens, is used.

5.2 Example of Alignment

Let us consider an example of a raw and a true decision for a patient with Liddle's syndrome (Aronson 2009). This genetic disorder is characterized by early, and frequently severe, high blood pressure associated with low plasma renin activity, metabolic alkalosis, low blood potassium, and normal to low levels of aldosterone.

A woman with Liddle's syndrome presented with severe symptomatic hypokalaemia. Her doctor reasoned as follows [text to be verified]:

– she has potassium depletion;
– spironolactone is a potassium-sparing drug;
– spironolactone will cause her to retain potassium;
– her serum potassium concentration will normalize.

She took a full dose of spironolactone for several days, based on this logical reasoning, but still had severe hypokalaemia.

Her doctor should have reasoned as follows [true text]:

– she has potassium depletion due to Liddle's syndrome, a channelopathy that affects epithelial sodium channels;
– there is a choice of potassium-sparing drugs;
– spironolactone acts via aldosterone receptors, amiloride and triamterene via sodium channels;
– in Liddle's syndrome an action via sodium channels is required.

When she was given *amiloride* instead of *spironolactone,* her serum potassium concentration rapidly rose to within the reference range. This stresses the importance of understanding the relationship between the pathophysiology of the problem and the mechanism of action of the drug. Channelopathies are diseases caused by a broken function of ion channel subunits or the proteins that regulate them. These diseases may be either congenital (often resulting from a mutation or mutations in the encoding genes) or acquired (often resulting from autoimmune attack on an ion channel).

Now let us try to apply generalization/alignment for raw and true texts. Alignment of a graph against a graph is a mapping retaining the labels on arcs between the source and target nodes of these two graphs. We coordinate an alignment between two semantic graphs and respective syntactic trees accordingly: semantic and syntactic relations between the source and target graphs are honored. Further details on alignment of linguistic structures are available in Chap. 3 and (Galitsky et al 2014).

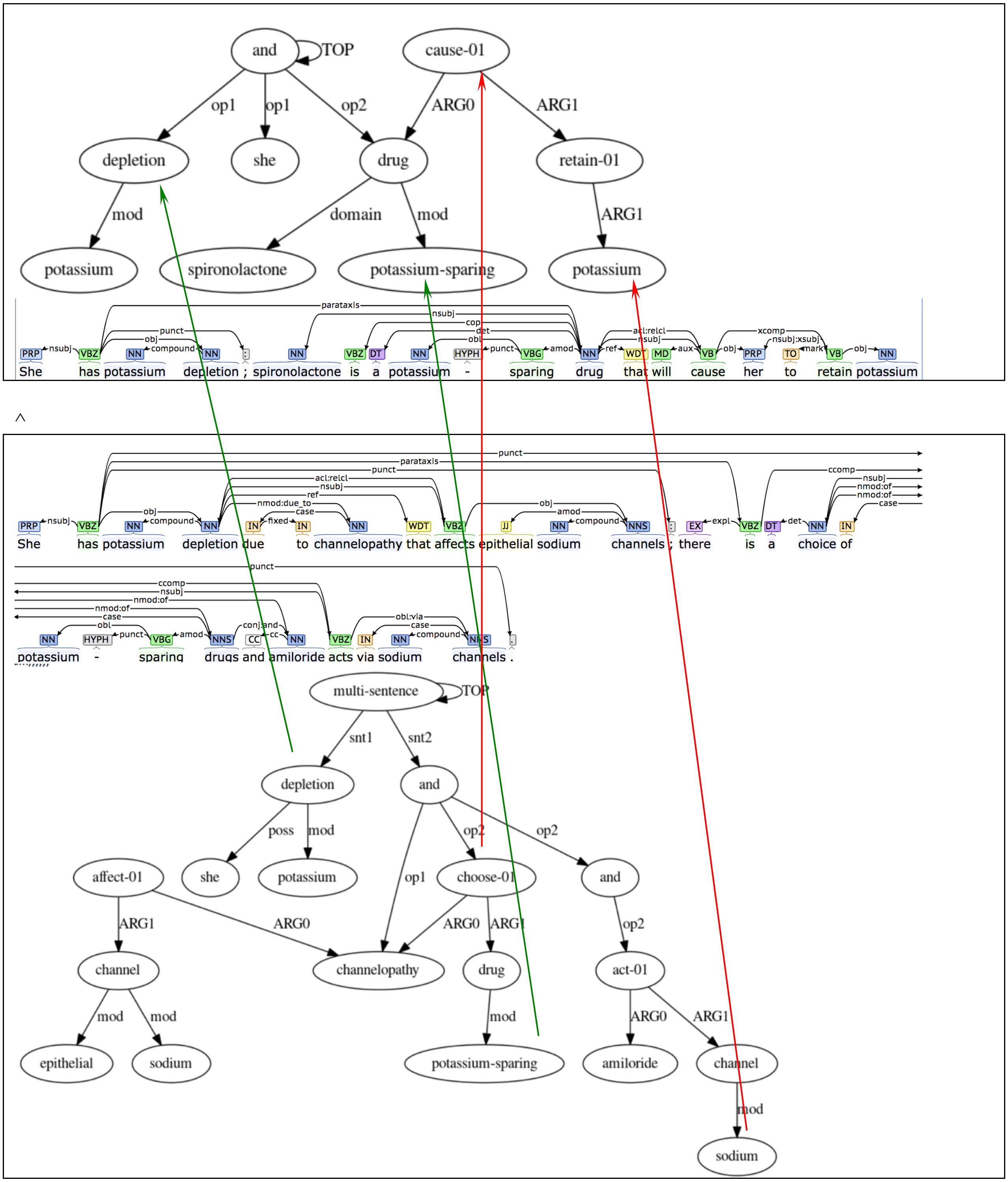


Fig. 6. A map between semantic and syntactic representations for raw (incorrect, on the top) and true (correct, on the bottom) treatment of a disease

The alignment maps *potassium depletion* into *potassium depletion, potassium-sparing* into *potassium-sparing, and [true] sodium channel* into [raw] *retain potassium,* and [true] mental action *cause* into raw mental action *chose.* Mapping into the same or synonymous entity is shown in green arrows, and the substitution of **LLM** with **true** is shown with red arrows (Fig. 6).

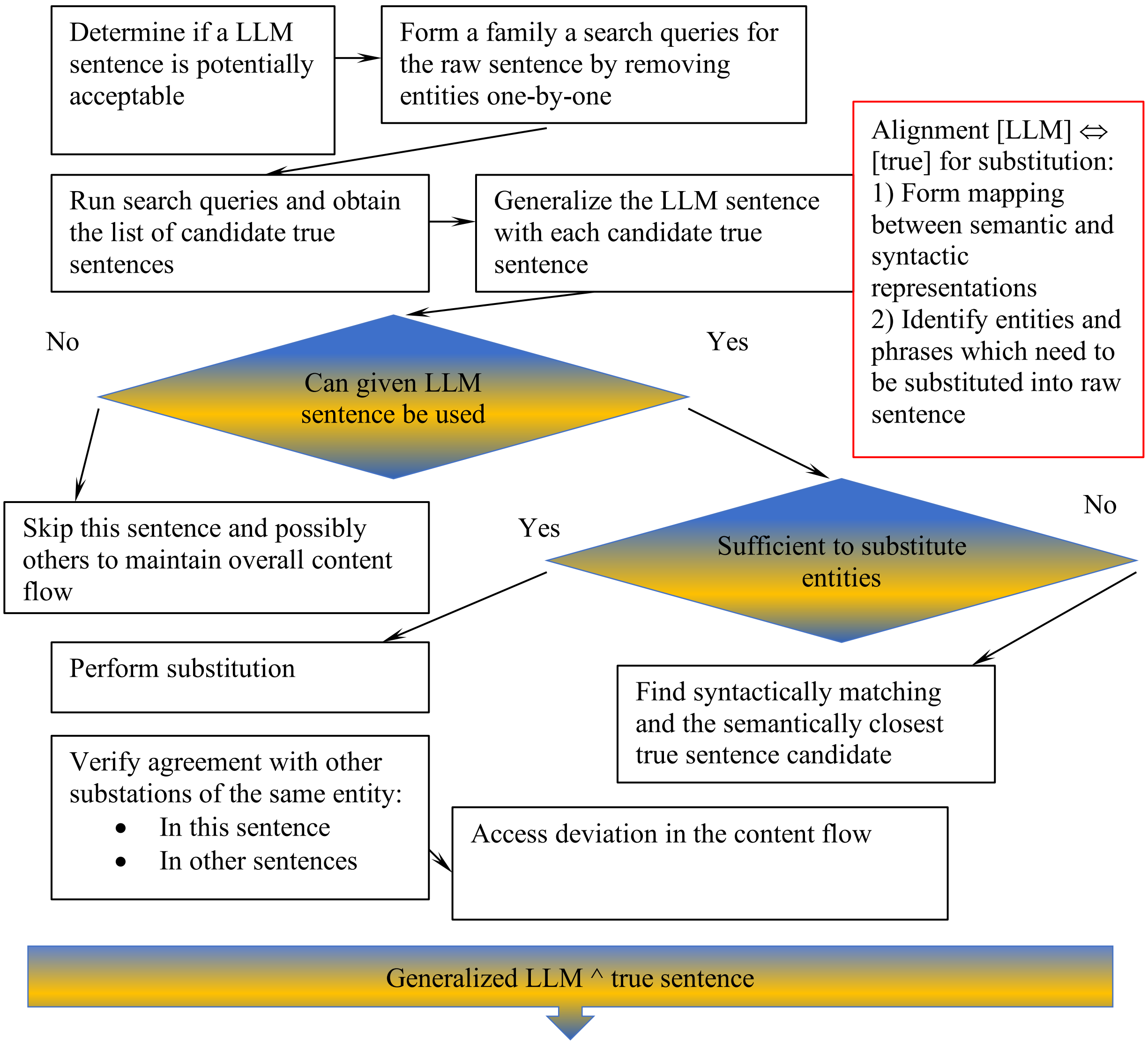


Fig. 7. Hallucination identification and correction architecture

For each sentence in the raw text, we perform a deterministic fact-checking. We iterate through each sentence in the raw text and try to correct it if necessary, modifying its syntactic structure to an as little degree as possible. Fig. 7 shows the correction procedure for a sentence. We first take a sentence and apply syntactic criteria to determine if it is worth retaining in the corrected content. We then proceed to fact-checking, forming a family of queries from this sentence to obtain the candidate true ones.

Candidate true sentences are extracted from search results snippets and identified documents. The true sentences are extracted and matched against the given raw sentence to determine what is the optimal (minimal change) substitution. Syntactic and semantic alignments are built and the entities and phrases to be substituted are determined. The architecture chart shows major decisions on whether this raw sentence can potentially be corrected and if *yes,* should it be an *entity substitution* or a *phrase substitution*. For multiple sentences, substitutions should be propagated according to the structure of coreferences in the raw text.

6. Evaluation

We provide a thorough evaluation of Truth-O-Meter in a broad range of domain, from token-level and Fact Extraction and Verification Annotation Platform to question answering (QA), information seeking dialogues and health recommendations. We compare our performance for a one-shot Thuth-O-Meter, its iterative collaboration mode Thuth-O-Meter-i and a hybrid mode, where the mode is decided individually for each phrase being verified.

6.1 Hallucination types

We first enumerates the types of hallucinations and show that only the first type is relatively frequent. This assessment is made in all our evaluation domains in this section. Table 1 includs the types of hallucination in LLM's texts along with its relevant frequency.

Table 1. Hallucination types

Hallucination type	Example	% of identified cases
Domain-specific Knowledge	Born in Saints Peterburg (Moscow), Suvorov studied military history as a young boy and joined the Imperial Russian Army.	89
Commonsense knowledge	The sports award is received at the start (end) of the competition	4
Incoherence or improper collocation	They describe the civil disobedience by many (people) in their town	4
Unrelated to the central topic	Composer Tchaikovsky's work was first publicly performed in 1865. In 1868, his First Symphony was well-received as he established himself as an artist (Piano Concerto No.1)	2
Conflict with preceding context	Tchaikovsky was born on May 7, 1840, in Kamsko-Votkinsk, Vyatka, Russia. He was the second eldest of his parents' six surviving offspring. As a youngest () member of the family…	1

In terms of the interaction mode, there are following hallucination scenarios (Williamsn 2023):

1) Hallucination based in *dialogue history*. Dialogue history-based hallucinations are produced when an LLM confuses names or relations of entities. For example, if the user mentions that their wife Mary likes shopping, and later informs their niece Lynn is coming to show her recent purchase, the LLM might incorrectly connect Mary and Lynn together as the same person. Furthermore, during a dialogue, the LLM can form incorrect conjectures based on previous mistakes within the dialogue tracking, breaking the context and content of a conversation in a sequence.
2) Hallucination in *abstractive summarization*. Although summarization is useful in condensing information, generative summarization systems make errors and deviate between the original and generated data.

3) Hallucination in generative question answering. It occurs when an LLM makes an erroneous inference from its source information and produces an incorrect answer to a user question. This can happen even when relevant source material is available. For example, if a user asks, "Where a swimmer can be attacked by a sea urchin: Andaman sea or Mediterranean sea?" and context is provided stating that the first choice is true, an LLM may still wrongly respond "Mediterranean sea " due to its own prior knowledge about Mediterranean sea being a top destination for beach vacations. Rather than accurately recalling the pre-existing source information, LLM can ignore the evidence and perform an unjustified inference based on its existing knowledge.
4) General data generation hallucination. LLM model generates outputs that may appear plausible or coherent but are not supported by factual or reliable information. It is a type of error where the model fabricates details or makes assumptions that go beyond the input data it has been trained on. This can result in the generation of false or misleading information that may seem convincing to humans but lacks a proper factual basis. Unlike other types of hallucination, the root cause of a general data hallucination is an overextension beyond training data rather than an incorrect inference or lack of grounding. The mode essentially imagines new information that isn't warranted by its training data.

Our approach can handle all four above cases, but the focus is 4)

6.2 Token-level hallucination correction

We first evaluate our web mining module on the annotated hallucination dataset HADES (Liu et al 2021) which contains about 9000 texts where about a half of them contains hallucination.

Building the HADES dataset, (Liu et al 2021) perturb "raw text" web data into "perturbed text" with out-of-box BERT model. Human annotators were asked whether the perturbed text spans are hallucinations given the original text. Verbs and nouns and other parts-of speech occur in approximately 1:1:1 ration, with about a half of verbs and nouns hallucinating. About a one-eighth of NER such as *organizations* are hallucinating. Liu et al. 2021 split the dataset into train, validation and test sets with sizes of 8754, 1000, 1200 respectively. "hallucination" cases slightly outnumber "not hallucination" cases, with a ratio of 54.5%/45.5%.

Testing HADES dataset showing that web mining can detect most of artificially created hallucinations (Table 2). We only identify wrong facts here and do not correct them. A complete list of resources for matching includes YouTube, eBay, Google knowledge graph, Twitter and Reddit.

Table 2: Token-level hallucination detection

	P	R	F1
As detected by the authors (Liu et al 2021)	0.68	0.81	0.74
Whole web matching	0.60	0.74	0.66
Wikipadia matching	0.65	0.72	0.68
Matching with the totality of resources	0.66	0.74	0.70

One can see that Truth-O-Meter is only 5.4% short of the native HADES performance (done without using external resources, just the training dataset. We conclude that Truth-O-Meter performs comparably in this specific HADES domain; therefore, we can expect satisfactory performance beyond token-based fact-checking.

6.3 Evaluation against Fact Extraction and Verification Annotation Platform

We conduct our experiments on the large-scale dataset FEVER (Thorne et al. 2018, https://github.com/awslabs/fever). The dataset consists of 185k annotated claims with a set of 5+ million Wikipedia documents from a Wikipedia dump. We used the dataset partition from the FEVER Shared Task. We compare Truth-O-Meter and GEAR based on BERT (Zhow et al 2019). In the BERT-Concat system, the authors concatenate all evidence into a single sentence and utilize BERT to predict the relation between the concatenated evidence and the claim. In the training phase, the ground truth evidence is added into the retrieved evidence set with relevance score of one and select five pieces of evidence.

In the BERT-Pair system, (Zhow et al 2019) utilize BERT to predict the label for each evidence claim pair. Concretely, the authors use each evidence claim pair as the input and the label of the claim as the prediction target. In the training phase, the ground truth evidence is chosen for supported and defeated claims and the retrieved evidence for claims which do not require evidence. In the test phase, labels are predicted for all retrieved evidence-claim pairs. Because different evidence-claim pairs may have inconsistent predicted labels, an aggregator is leveraged to obtain the final claim label.

In Table 3, we show a FEVER score for fact-checking

Table 3. Evaluation in Fact Extraction platform

	FEVER score	P	R	F1
ChatGPT baseline				
GEAR (Zhou et al 2019)	67.1	70.6	81.6	75.7
Truth-O-Meter	69.1	75.1	78.8	76.9
Truth-O-Meter-i	66.8	76.0	80.6	78.2

We observe a similar performance of

6.4 Automated evaluation on QA datasets

We experiment with four datasets: SQuAD2.0, Natural Question (Kwiatkowski et al 2019) that employs multi-reference annotations to resolve ambiguity, TriviaQA (Joshi et al. 2017) and HotpotQA (Yang 2018). SQuAD is extended via an open-domain question answering benchmark that considers multi-step joint reasoning over both tabular and textual information. It consists of around 40K instances built upon Wikipedia, including 400K tables and 6M passages as the knowledge source. Solving the questions in OTT-QA requires diverse reasoning skills and can be divided into three categories: single-hop questions (1/10), two-hop questions (3/5), and multi-hop questions (1/3).

As a representative case in the customer support scenario, we utilize DSTC11 Track 5 (Kim et al., 2023), which builds upon the DSTC9 Track 1 dataset by integrating subjective knowledge from customer reviews alongside factual knowledge from frequently asked questions (FAQs). This combination provides users with a more comprehensive and informative experience when interacting with the QA system. The dataset assesses the QA system's capability to comprehend user review posts and FAQs, generating responses based on both sources. To evaluate the system's performance, we randomly selected 500 examples from each dataset's validation set and presented the results as F1 scores.

In Table 4 we compare performance of verification and correction answering question on various datasets. We compare approaches with different architectures and distinct roles of LLMs themselves, varying retrieval model and fact-checking sequence. Empty cells denote the lack of results for a particular evaluation setting by a given system.

Table 4. Identification of hallucination and correction in answering questions

Method	SQuAD extension	Natural Question	TriviaQA	HotpotQA	DSTC11 Track 5
Chain of thoughts (Wei et al 2022)		64.3	79.2	42.8	
CRITIC* (Gou et al 2023)		79.9	**86.6**	**56.9**	
LLM-Augmenter CORE (Peng et al.)					50.83
RARR (Gao et al 2023)		41.5			
Truth-O-Meter	**81.0**	78.2	84.9	51.3	**55.3**
Truth-O-Meter-i	79.3	**80.3**	86.1	50.1	53.0

We observe that Truth-O-Meter is best at DSTC11 followed by Truth-O-Meter-i and then by LLM-Augmenter. Truth-O-Meter-I demonstrates a superior performance at Natural Question followed by CRITIC*. In other QA domains, Truth-O-Meter family show an inferior performance in comparison to competitive systems.

6.5 Information-seeking dialogues

We now zoom-in on Truth-O-Meter overall quality of generated answers, correcting ChatGPT in customer support domain. We evaluate its performance on information-seeking dialog tasks using both automatic metrics and human evaluations. Following the literature, we consider commonly used metrics, Knowledge F1 (KF1) and BLEU-4, in grounded conversational response generation and task-oriented dialog. BLEU (Papineni et al., 2002) measures the overlap between the model's output and the ground-truth human response, while KF1 (Lian et al., 2019) measures the intersection with the knowledge that the human used as a reference during dataset collection. Moreover, ROUGE-1 (Lin, 2004) is used as this assessment have been found to best correlate with human judgment on customer support tasks (Kim et al., 2020) as well as BARTScore as one of the best model-based metrics (Yuan et al., 2021). DSTC11 Track 5 (Kim et al., 2023) with incorporated subjective knowledge from customer reviews in addition to factual knowledge from FAQs is employed.

Table 5: Evaluation scores in information-seeking dialogues (in %)

Method	KF1	BLEU	ROUGE	BARTScore
ChatGPT	26.7	1.0	16.8	0.25
LLM-AUGMENTER (Peng et al 2023)	36.41	7.6	**22.8**	0.35
chatGPT + Truth-O-Meter	33.5	8.0	21.8	0.30
chatGPT + Truth-O-Meter-i	32.1	**8.5**	23.3	**0.36**
chatGPT + Truth-O-Meter + Truth-O-Meter-i online selector	**33.9**	7.5	23.6	0.32

Collaborative mode of Truth-O-Meter is not preferred in all measurement setting of the resultant content, but only for KF1. Interactive Truth-O-Meter-I wins in two measurement settings. Overall, one can observe a 3-10% boost in performance for different measurements moving from LLM-AUGMENTER to Truth-O-Meter.

We now proceed to the performance of Truth-O-meter collaborative in Table 6 for a general information request prompt. We select four domains where precise answers are needed, and *four domain in the humanities*(shown in *italic*). In the second column, we show the number and % of sentences which were properly classified as hallucination (a precision of detection).

Table 6: Hallucination detection in various content domains

The class of common symptoms	#/% of properly identified hallucinations sentences		#/% of sentences accepted as they are		Total # of true sentences used
Health	54	88.4	1026	94.1	17.1
Geography	48	86.2	1251	96.3	16.3
Legal	74	83.4	980	92.8	19.0
Engineering	57	88.1	1340	94.7	18.7
Average	**58.2**	**86.52**	**1149**	**94.47**	**17.77**
History	*28*	*88.6*	*1420*	*97.9*	*20.4*
Marketing	*45*	*85.2*	*1237*	*98.6*	*19.3*
Art	*17*	*89.6*	*1478*	*96.3*	*17.5*
Politics	*52*	*86.5*	*1513*	*98.1*	*19.0*
Average	**35.5**	**87.47**	**1412**	**97.72**	**19.05**

The reader can observe that hallucination rate in natural sciences are almost twice as high as in humanities, having the recognition precision comparable. In humanities, 30% more sentences are acceptable as they are, since the notion of falsehood is fuzzy and acceptance rate is significantly higher.

6.6 Personalized drug intake recommendation domain

We generate personalized drug intake recommendations for patients based on their unique set of medical conditions. In input may include a wide range of patient-specific data, including medical history, genetic information, lifestyle factors, and environmental influences. Ideally, the personalized drug intake recommendations are designed to cater to the specific needs of each patient, taking into account factors such as drug interactions, potential side effects, and the patient's overall health status.

We obtain the list of drugs from https://www.rxlist.com/. 4k drugs names are selected. The list of medical conditions is available from https://www.medicinenet.com/diseases_and_conditions/article.htm. We select 700 conditions.

Firstly, we form the pairs of drug-condition (disease) where the relationships are *drug-treats-disease* as well as *side-effect-of-drug-under-condition, drug-cause-side-effect-under-condition*. We refer to this data as D-C pair set. In most cases, D is not intended to treat C: instead, we are concerned whether D *has a side-effect* on C, D *might-affect* C, D *might-cause-complication-of* C. Our choice of D-C association is connected with a *personalized recommendation* for a patient with C to take drug D.

These pairs are formed through exhaustive iteration through all drug-condition pairs and retaining these with an explicit relation. These relations were identified in texts about drugs and illnesses at https://www.rxlist.com/ and illnesses at https://www.medicinenet.com. As a result, we compile a list of 6500 pairs where we can experiment with writing a personalized content for drug D and a patient with condition C.

We show the correction characteristics in Table 7. In the leftmost column, the class of symptoms is shown for each assessment group. In columns two to four, we show:

- Average # of corrected sentences for a recommendation (10 sentences average);
- % of properly identified hallucinations sentences;
- % of properly corrected hallucinations sentences.

Table 7. Correction characteristics

The class of common symptoms	Average # of corrected sentences for a recommendation	% of properly **identified** hallucinations sentences	% of properly **corrected** hallucinations sentences
Bloating	0.8	86.3	77.1
Cough	0.7	82.6	79.3
Diarrhea	1.0	85.8	76.3
Dizziness	0.7	88.4	80.1
Fatigue	0.9	84.6	79.2
Fever	0.7	85.0	78.8
Headache	0.8	85.6	76.5
Muscle Cramp	0.8	83.5	77.0
Nausea	0.7	86.4	75.9
Throat irritation	0.6	83.4	79.3
Average	**0.77**	**85.16**	**77.95**

Table 8. The error rate of repaired generated content, %

The class of common symptoms	Rate of wrong facts per text for LLM content	Rate of wrong facts per text for **corrected** content	Rate of wrong facts per sentence for LLM content	Rate of wrong facts per sentence for **corrected** content
Bloating	29.6	5.0	3.5	0.31
Cough	23.0	6.4	3.1	0.34
Diarrhea	26.1	6.7	2.9	0.28
Dizziness	24.5	6.6	3.0	0.29
Fatigue	24.0	5.1	3.4	0.27
Fever	25.7	4.8	2.7	0.29
Headache	26.6	5.1	3.2	0.30
Muscle Cramp	24.1	5.6	3.5	0.26
Nausea	27.0	6.1	3.3	0.27
Throat irritation	25.9	5.9	3.0	0.31
Average	25.65	5.73	3.16	0.292

We observe that only 1/4 of LLM texts can be accepted without repair, according to the automated fact-checking. After correction, more than 94% of texts do not contain wrong facts. 26% of wrong facts are in

LLM text, which is reduced to about 3% by the correction procedure, according to the automated fact-checking (Table 8).

To automatically verify the truthfulness of the repaired generated content, we repeat the repair procedure and measure the frequency of cases when the repaired sentence or repaired phrase is not confirmed by any true sentence or phrase.

6.7 Error analysis

In our analysis, the main errors of our framework come from the upstream document retrieval and sentence selection components which can not extract sufficient evidence for inferring. For example, to verify the claim "Dexter is available on Netflix", we need the evidence "You can watch all eight seasons of *Dexter* on Showtime, as well as the new series, *New Blood*. All eight seasons of *Dexter* are available on Amazon Prime U.S., but *New Blood* is only available with the **Showtime** premium add-on." and
"Prime members can subscribe to **Showtime** with Prime Video Channels. 7-Day Free Trial then only $10.99/month. Prime membership needed.".

Unfortunately, the entity linking method employed in our text retrieval engine failed to retrieve the "Showtime" document solely from parsing the content of the text for truth-checking. Consequently, the claim verification component was unable to reach a proper conclusion due to insufficient evidence. To understand the claim verification process in this scenario, we conduct fact-checking on an evidence-enhanced development set. In this set, we added the ground truth evidence with the highest relevance score to the evidence set, ensuring that each claim in the evidence-enhanced set was supported by both the ground truth evidence and the retrieved evidence.

6.8 Examples of repairs for hallucination

Q: Rolwaling Valley

Text with suspicious facts:
The Rolwaling Valley is located at high altitude, with the highest point on the trek reaching ~~6,181 meters~~
Text with suggested rewrites:
The Rolwaling Valley is located at high altitude, with the highest point visible on the trek reaching *7181 meters*
Citing: [Rolwaling Valley trek | Rolwaling Tashi Lapcha Pass itinerary]
The Rolwaling Valley is known for its stunning mountain scenery, including the towering peaks of Gauri Shankar (7,145 meters) and Melungtse (7,181 meters). The valley is also home to several glaciers and pristine alpine lakes, including Tsho Rolpa, which is one of the largest glacial lakes in Nepal.

Q: Why is Florida real estate cheap?

Text with suspicious facts:
Florida real estate is relatively inexpensive compared to some other popular states:
Lower taxes.
Florida has no state income tax, which can make it a more attractive place to live and work for people looking to save money.
Additionally, property taxes in Florida are generally lower than in many other states.
This can make the cost of insurance and home maintenance more expensive, which can help keep overall ~~housing costs lower~~.
Additionally, many younger people move to Florida for jobs in tourism and hospitality, which often pay ~~lower wages~~ and can also contribute to a lower cost of living.

Text with suggested rewrites:
Florida real estate is relatively inexpensive compared to some other popular states:
Lower taxes.
Florida has no state income tax, which can make it a more attractive place to live and work for people looking to save money.
Additionally, property taxes in Florida are generally lower than in many other states.
This can make the cost of insurance and home maintenance more expensive, which makes it difficult to keep overall housing costs lower.
Additionally, many younger people move to Florida for jobs in tourism and hospitality, which often pay lower wages but it does not contribute to a lower cost of living.
Citing: Florida is the least affordable place to live in the U.S.
In February, Realtor.com designated Miami as America's least affordable place to live. Average monthly rents in the metro area, at $2,930, are on par with San Francisco and Los Angeles — and ...

One can observe that a factual and commonsense error is corrected on the top in travel/geographical domain, while a commonsense error is corrected on the bottom in the domain of economics, where the correlations between factors need to be made correct. Instead of updating the factors themselves, we correct the relations, in particular, turning the polarity to negative.

Since LLMs lack true comprehension of the source text, instead relying on pattern recognition and statistics, they may, as a result, distort or even entirely fabricate details, inferring unsupported causal relationships or retrieving unrelated background knowledge. One can observe that without a grounding in common sense or factual knowledge, LLMs can get lost and generate hallucinations.

7. Discussions

hallucinations a feature in large language models. You want to see the models hallucinate if you want them to be creative. For example, if you ask ChatGPT or other Large Language Models to give you a plot of a fantasy story, you want it not to copy from any existing one but to generate a new character, scene, and storyline. This is possible only if the models are not looking up data that they were trained on.

Another reason you may want hallucinations is when looking for diversity, for example, asking for ideas. It is like asking the models to brainstorm for you. You want to have derivations from the existing ideas that you may find in the training data, but not exactly the same. Hallucinations can help you explore different possibilities.

Many language models have a "temperature" parameter. You can control the temperature in ChatGPT using the API instead of the web interface. This is a parameter of randomness. The higher temperature can introduce more hallucinations.

In automated fact-checking, Nakov et al. (2021) have recently attempted to connect research to users by examining the concerns of professional fact-checkers. Generally, they find that organisations desire automation beyond veracity prediction models, which cannot function in isolation of media monitoring, claim detection, and evidence retrieval.

In their recent work, Liu et al. (2023) have introduced RETA-LLM, a comprehensive toolkit aimed at supporting research and fostering the advancement of retrieval-augmented Language Model (LLM) systems. This innovative toolkit offers a complete pipeline that empowers researchers and users to effortlessly design personalized LLM-based systems tailored to specific domains. In comparison to earlier retrieval-augmented LLM systems, RETA-LLM stands out for its inclusion of versatile plug-and-play modules, facilitating seamless interaction between Information Retrieval systems and LLMs. These modules encompass request rewriting, document retrieval, passage extraction, answer generation, and fact checking, collectively enhancing the overall performance and effectiveness of LLM-based systems.

Numerous LLMs for text generation (Radford et al., 2018) have been proposed over the years, including GPT-3 (Ouyang et al., 2022) and ChatGPT. However, most of them do not naturally incorporate external knowledge. To address this limitation, various works augment LLMs with knowledge consisting of *e.g.,* personalized recommendations (Ghazvininejad et al., 2017), Wikipedia article and web search (Shuster et al., 2022), structured and unstructured knowledge of task-oriented dialog (Peng et al., 2022).

Other attempts to combine black- box LLMs with external knowledge, such as incorporating external knowledge into prompts (Lazaridou et al., 2022), making GPT-3 more faithful (He et al., 2022), and combining web knowledge with GPT-3 (Nakano et al., 2021) have been made. Also, Shi et al. (2023) tune the ranker of a black-box LLM. Schick et al. (2023) tune black-box LLMs' access to different APIs and show improvement on a variety of understanding and reasoning tasks. However, a detailed account of how to combine multiple inconsistent sources of external knowledge, how to adjust LLM results to the identified knowledge, maintaining the resultant discourse has been developed in this paper.

7.1 Example of official fact-checking by humans

https://www.cnn.com/2023/06/13/politics/fact-check-trump-federal-documents-indictment/index.html

Trump claimed in: "You're watching Joe Biden try to jail his leading political opponent. Think of it – this is like third world country stuff."

CNN: *This claim is not supported by any evidence. There is no sign that Biden has been involved in the decision to criminally investigate or prosecute Trump; ordinary citizens on a Florida grand jury voted to indict Trump, and the prosecution is led by a special counsel, Jack Smith. Smith was appointed in November 2022 by Attorney General Merrick Garland, a Biden appointee, but that is clearly not proof that Biden was involved in the prosecution effort.*

ChatGPT (background info): In a democratic system, politicians, including presidents, should engage in healthy competition, debate, and discourse to win the support of voters. They can differentiate themselves from their opponents by presenting their policies, achievements, and visions for the country.

It is essential for a president, or any political leader, to prioritize the well-being of the nation and its citizens rather than seeking to harm opponents. Their focus should be on implementing effective governance, collaborating with other political stakeholders, and addressing the needs and concerns of the people they serve.

Engaging in fair and respectful competition encourages a vibrant democracy and contributes to the overall progress of a nation.

Author (Personal experience): Both FBI and police in California are reluctant to open investigation or help in any way when an average taxpayer citizen is obviously a victim of a crime. FBI and police as a law enforcement branch of government neither serve the people nor address their concerns in most cases but instead pursue certain political goals

8. Conclusions

Content verification problems are known to be highly challenging, and with the rapid progress of generative AI, the ease of content manipulation and spreading misinformation on a large scale has significantly increased. Whether it is malicious individuals or discrete enthusiasts, the potential for creating deep fakes of political leaders using AI tools has been demonstrated through the manipulation of authentic videos, audios, and images. Media extensively highlights the dangers of deep fakes, as they are utilized to sow discord within society and disrupt public order. Consequently, it becomes crucial to examine the role of generative AI in the face of the growing threat of misinformation, which poses a significant challenge and risk to the functioning of the public sphere and democracy (Kuehn and Salter 2020).

Fact-checking is the prevailing method used to counter misinformation, involving the evaluation of information for its accuracy and truthfulness. Although research has shown its effectiveness in debunking fake news, the labor-intensive nature of fact-checking poses scalability challenges (Micallef et al., 2022). LLM-generated misinformation can blend factual statements with falsehoods, manipulate the context and scope of facts, evoke emotional responses, and draw biased conclusions, presenting additional hurdles for fact-checkers and content moderators in assessing the potential for misleading readers.

Our project on correcting generative content began well before ChatGPT (Galitsky 2022), and we have observed substantial advancements in content correction algorithms, leading to the development of Truth-O-Meter. As LLMs improved, so did these algorithms, enabling higher performance when applied to newer models like GPT-4, especially after fine-tuning them with earlier, less accurate LLMs. Additionally, Truth-O-Meter's efficacy is enhanced by incorporating additional fact-checking plug-and-play components over time. As both LLMs and their correction algorithms progress in tandem, the wider acceptance of LLMs by the general public at the end of 2022 necessitates reliable verification and correction modules like Truth-O-Meter for their trustworthy use.

In this study, fact-checking is implemented as a collaboration of human and LLM in creating a trusted content. A general case of such collaboration is described in (Goldberg et al 2022) as approach to a Bi-directional Decision Support System (DSS) as an intermediary between an expert and a ML system for choosing an optimal solution. As a first step, such DSS analyzes the stability of expert decision and looks for critical values in data that support such a decision. If the expert's decision and that of a machine learning system continue to be different, the DSS makes an attempt to explain such a discrepancy.

It is well known that deep learning in general and LLMs in particular have strong limitations due to a lack of explainability and weak defense against possible adversarial attacks. In (Galitsky et al 2023) we proposed a *Meta-learning/Deep networks* $\rightarrow$ *kNN* architecture that overcomes these limitations by integrating deep learning with explainable nearest neighbor learning. We evaluated the proposed architecture for pre-ChatGPT content creation tasks and observed a significant improvement in performance along with enhanced usability by team members. We observed a substantial improvement in question answering accuracy and also the truthfulness of generated content due to the application of the Shaped-Charge learning approach. Hence Truth-O-Meter architecture is a successful example of a Shaped-Charge learning framework; in our future studies we plan to present a complete account including meta-level.

In conclusion, we summarize the following advantages of our approach to tackling hallucination:

- Fact checking via web mining adjusted to LLM by means of collaborating with LLM and applying a spectrum of text matching approaches;
- Applicability to texts of various genres and mixture of formats and genres in a document. By relying on a diversity of text matching algorithms acting on phrase and sentence level, we identify hallucination in various parts of documents;
- Application to the whole document with opinionated data, background, captions, results, design notes etc. Truth-O-Meters only applies fact-checking to the phrases and sentences where hallucination can potentially occur;

- Performing proper contextualization to identify and fully substitute facts from what is expressed in text;
- Handling sources with inconsistency by finding least defeated authoritative source and avoiding most defeated

We conclude that Truth-O-Meter significantly reduces LLM's hallucinations while retaining the fluency and informativeness of its generated responses.

References

1. Aronson, J (2009). Medication errors: What they are, how they happen, and how to avoid them. QJM: Monthly Journal of the Association of Physicians. 102. 513-21.
2. Epstein D, Illia Polosukhin, Jacob Devlin, Kenton Lee (2019) Natural questions: a benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:453–466,
3. Escarza, S., Larrea, M.L., Castro, S.M., & Martig, S. (2009). DeLP viewer: a defeasible logic programming visualization tool.
4. Estes A, Nikhita Vedula, Marcus Collins, Matt Cecil, and Oleg Rokhlenko. 2022. Fact Checking Machine Generated Text with Dependency Trees. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 458–466, Abu Dhabi, UAE. Association for Computational Linguistics.
5. Galitsky B (2022) Improving open domain content generation by text mining and alignment. In Artificial Intelligence for Healthcare Applications and Management. Elsevier.
6. Galitsky B, Ilvovsky D, Goldberg S. Shaped-Charge Learning Architecture for the Human–Machine Teams. Entropy. 2023; 25(6):924. https://doi.org/10.3390/e25060924
7. Galitsky, B. Employing abstract meaning representation to lay the last-mile toward reading comprehension. In Artificial Intelligence for Customer Relationship Management; Springer: Berlin/Heidelberg, Germany, 2020; pp. 57–86.
8. Galitsky, B. Merging deductive and inductive reasoning for processing textual descriptions of inter-human conflicts. J Intell Inf Syst 27, 21–48 (2006).
9. Gao L, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan (2023). Attributed text generation via post-hoc research and revision. arXiv preprint arXiv:2210.08726.
10. Garcia A, Simari G (2004) Defeasible logic programming: an argumentative approach. Theory Pract Logic Program 4:95–138
11. Goldberg, S.; Pinsky, E.; Galitsky, B. A bi-directional adversarial explainability for decision support. Hum. Intell. Syst. Integr. 2021, 3, 1–14.
12. Gururangan S, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: adapt language models to domains and tasks. In ACL, 2020.
13. He H, Hongming Zhang, and Dan Roth. 2022. Rethinking with retrieval: Faithful large language model inference. arXiv preprint arXiv:2301.00303.
14. Hecham A (2016) DEFT, an open source Java tool for Defeasible Datalog. https://github.com/raoufhec/DEFT/
15. James Thorne, Andreas Vlachos, ChristosChristodoulopoulos, and Arpit Mittal. (2018) Fever: a large-scale dataset for fact extraction and verification. In Proceedings of NAACL-HLT, pages 809–819.
16. Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, Maosong Sun (2019) GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification
17. Joshi M, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. (2017) TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. ACL, 1601–1611
18. Kim S, Spandana Gella, Di Jin, Alexandros Papangelis, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. (2023) DSTC11 track proposal: Task-oriented conversational modeling with subjective knowledge. https://github.com/alexa/dstc11-track5.
19. Kuehn KM and Leon A Salter. 2020. Assessing digital threats to democracy, and workable solutions: a review of the recent literature. International Journal of Communication 14 (2020), 22.
20. Kwiatkowski T, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti,

21. Lazaridou A, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. arXiv preprint arXiv:2203.05115.
22. Le, and Denny Zhou. (2022) Chain of thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems
23. Lee, Nayeon, Wei Ping, Peng Xu, Mostofa Patwary, Mohammad Shoeybi and Bryan Catanzaro. "Factuality Enhanced Language Models for Open-Ended Text Generation." ArXiv abs/2206.04624 (2022)
24. Levesque H.J., Lin F., Reiter R. (1997) Defining Complex Actions in the Situation Calculus. Technical Report, Department of Computer Science, University of Toronto
25. Li XL, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A Systematic Investigation of Commonsense Knowledge in Large Language Models. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 11838–11855
26. Lian R, Min Xie, Fan Wang, Jinhua Peng and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. In International Joint Conference on Artificial Intelligence.
27. Lin C-Y. 2004. ROUGE: A package for automatic evaluation of summaries. In ACL workshop, pages 74–81.
28. Liu T, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. (2022) A Token-level Reference-free Hallucination Detection Benchmark for Free-form Text Generation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.
29. Maynez J, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factu- ality in abstractive summarization. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1906–1919, On- line. Association for Computational Linguistics.
30. Micallef N, Vivienne Armacost, Nasir Memon, and Sameer Patil. 2022. True or False: Studying the Work Practices of Professional Fact-Checkers. Proceedings of the ACM on Human-Computer Interaction 6, CSCW1 (2022), 1–44.
31. Nakano R, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders 2021. WebGPT: Browser-assisted question answering with human feedback. arXiv preprint arXiv:2112.09332.
32. Nakov P, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barron- ´ Cedeno, Paolo Papotti, Shaden Shaar, and Giovanni ˜ Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, pages 4551–4558. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
33. Ouyang L, 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 E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. ArXiv, abs/2203.02155.
34. Papineni K, Salim Roukos, ToddWard, and Wei- Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In ACL, pages 311–318.
35. Peng, Baolin & Galley, Michel & He, Pengcheng & Cheng, Hao & Xie, Yujia & Hu, Yu & Huang, Qiuyuan & Liden, Lars & Yu, Zhou & Chen, Weizhu & Gao, Jianfeng. (2023). Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback. 10.48550/arXiv.2302.12813.
36. Pennington J, Richard Socher, and Christopher D Manning. 2014. GLOVE: Global vectors for word representation. In Proceedings of the EMNLP, 1532–1543.
37. Rebuffel C, Marco Roberti, Laure Soulier, Geof- frey Scoutheeten, Rossella Cancelliere, and Patrick Gallinari. 2021. Controlling hallucinations at word level in data-to-text generation. arXiv preprint arXiv:2102.02810.
38. Wang C and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural ma- chine translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Lin- guistics, pages 3544–3552, Online. Association for Computational Linguistics.
39. Wei J, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V
40. Williams A, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In NAACL-HLT, pages 1112–1122.

41. Williams A. (2023) Fact or Fiction: What Are the Different LLM Hallucination Types? https://www.holisticai.com/blog/types-of-llm-hallucinations
42. Yin, W., Jamaal Hay, Dan Roth (2020) Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach . arXiv:1909.00161
43. Yuan W, Graham Neubig, and Pengfei Liu. 2021. BARTScore: Evaluating generated text as text generation.
44. Zhou J, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. (2019). GEAR: Graph-based Evidence Aggregating and Reasoning for Fact Verification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 892–901, Florence, Italy. Association for Computational Linguistics.
45. Zhou J, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic Lies: Understanding AI-Generated Misinformation and Evaluating Algorithmic and Human Solutions. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 436, 1–20.
46. Zhou, Chunting & Gu, Jiatao & Diab, Mona & Guzman, Paco & Zettlemoyer, Luke & Ghazvininejad, Marjan. (2020). Detecting Hallucinated Content in Conditional Neural Sequence Generation.