Cross-validation of Answers with SUMO and GPT

Dan Lupu[1,*,†], Adrian Groza[1,†] and Adam Pease[2,†]

[1]Department of Computer Science, Technical University of Cluj-Napoca, Romania
[2]Articulate Software, USA

Abstract

We have developed a tool for fact-checking in automated question answering based on four technologies: (i) the Suggested Upper Merged Ontology (SUMO) for knowledge representation, (ii) the Vampire theorem prover [1] for fact verification, (iii) WordNet for lexical semantics and (iv) GPT (Generative Pre-trained Transformer) for concept learning and alignment. SUMO provides a structured representation of knowledge in an expressive logic, facilitating semantic understanding and analysis. Vampire serves as an automated reasoning tool to check the validity of facts and claims. WordNet and GPT contribute to concept learning and alignment, enhancing the system's ability to interpret natural language (NL) expressions and align them with the underlying ontological representations. By combining these components, the proposed framework offers a robust solution for fact-checking, combating misinformation, and promoting informed decision-making.

Keywords

fake news, generative pre-trained transformers, foundational ontologies, foundational language models

1. Introduction

A number of fact-checking systems rely on linking claims to fact-checked statements in structured data like databases [2], knowledge graphs (e.g. ClaimsKG [3], CompareNet [4]), or ontologies (e.g. FACE system [5]). The challenge is how the given statement is translated from natural language into the structured representation of the checker. Such translators include FRED that converts from NL into OWL [6] or various relation extractors [7]. However, since pre-trained transformers are considered the best current technology for translating between languages, it is natural to apply them to the task of converting a natural to a formal language, for fact-checking.

We developed here a fact checker that exploits the knowledge representation capabilities of Suggested Upper Merged Ontology (SUMO) [8]. The knowledge in SUMO is formalised in the SUO-KIF (Standard Upper Ontology Knowledge Interchange Format) format [9], which is a higher-order logic language [10]. To translate from NL to SUO-KIF we rely on the GPT-3 Curie model. We fine-tuned the Curie model for the current task of automatically generating SUO-KIF axioms from natural language. The Vampire theorem prover is used to signal if a given statement is "True", "False" or "Unknown", based on the available SUMO knowledge and

KBC-LM@ISWC2023: 1st 1st Workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM)
dan.lupu@gmail.com (D. Lupu); adrian.groza@cs.utcluj.ro (A. Groza); apease@articulatesoftware.com (A. Pease)
https://users.utcluj.ro/~agroza (A. Groza); https://www.adampease.org/ (A. Pease)
0000-0003-0143-5631 (A. Groza); 0000-0001-9772-1266 (A. Pease)

also provides supporting evidence for its conclusions. By enriching SUMO with a food domain ontology, we provide a running scenario for detecting fake claims regarding diets.

2. System architecture

Figure 1 illustrates the deployment diagram of the application, showing how three microservices interact with each other. The server-side includes the following microservices: (i) *ontology-rest* responsible for interacting with the ontology: ontology alignment, Vampire querying; (2) *fake-news-detector-api*: implementing the algorithm for text verification; (3) *gpt-translator*: serving as a REST interface for interacting with the fine-tuned models; (4) users interact with the system through a web page deployed by the *fake-news-detector-api*.

The content of SUMO [8] aids in understanding and logical thinking by precisely outlining a wide range of commonly used terms and how they relate to one another. These terms cover ideas from various areas including mathematics, social frameworks, procedures, time, and the physical world, among other categories. For the task of fact-checking, SUMO provides a series of advantages due to its broad coverage and formal semantics: (i) *Concise and clear knowledge:* used to establish the circumstances and interpretation of information, thus aiding in a distinct comprehension and assessment; (ii) *Reasoning:* the employment of logical axioms in SUMO enables the deduction of factual information or to signal conflicts with known facts within the ontology; (iii) *Interoperability:* SUMO is connected to WordNet [11], which assists in comprehending the meaning of natural language statements in terms of semantics; (iv) *Extensibility:* many domain ontologies have been built on top of SUMO, allowing a fact-checking app to serve diverse areas - as running scenario, we engineer an ontology for diet and nutrition domain.

SUMO also benefits from the Sigma Knowledge Engineering Environment (SigmaKEE) [12], through a set of features for browsing, editing, and managing SUMO, including inference capabilities, semantic integration, and NLP (i.e. SigmaNLP). Thus, *SigmaKEE* provides support for multiple automated theorem provers, including Vampire, EProver[13], and LEO-III[14].

The Algorithm 1 evaluates the truth value of an input text, yielding an output of 'True', 'False', or 'Unknown'. The algorithm processes the current claim *txt* which undergoes ontology matching, transforming it into a form suitable for further processing. This matched text is then translated into tSUO-KIF, creating a query. The query is then put through a theorem prover. If a proof is found for this query, it implies that the *txt* corresponds to a 'True' statement. If the theorem prover does not find a proof for the query, the algorithm takes the negation of the query and poses it to the theorem prover. This is necessary because not finding a proof for a query doesn't automatically mean the negation of the query is true. Under the open world assumption, the negation of a statement can be 'Unknown'. Finding a proof for the negated query means that the *txt* corresponds to a 'False' statement. In case of no proof found for the negated query, the validity of the claim is assessed as 'Unknown', as neither the query nor its negation could be proven. Finally, the output True, False, or Unknown is passed to the explanation step, which is an example of Explainable AI (XAI).

The execution of this algorithm corresponds to the flowchart in Figure 2. The main four steps: (1) ontology alignment; (2) translation to SUO-KIF, (3) theorem proving and (4) explanation are

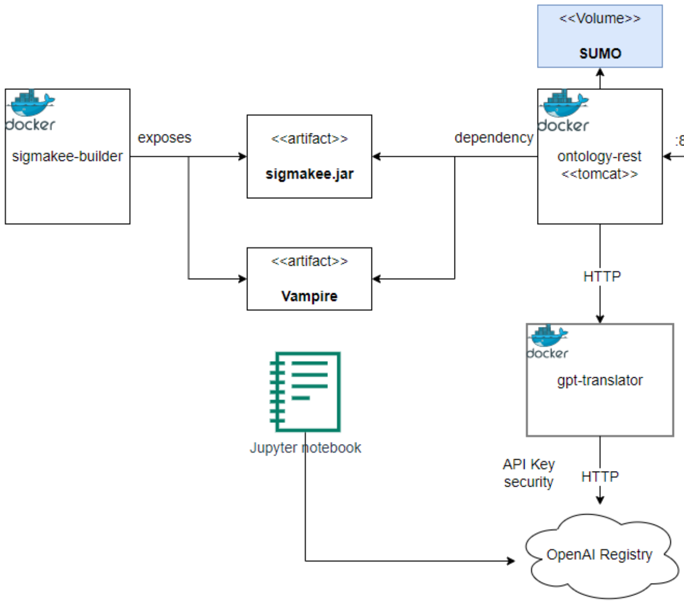

Figure 1: System architecture

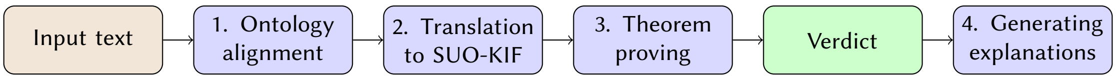

Figure 2: Fact checking flow

detailed in the following paragraphs.

First, for *aligning text to ontology*, we map and substitute tokens from a corpus of text to concepts from SUMO. The Natural Language Processing (NLP) pipeline defined in *SigmaNLP* [15] is used, where *SigmaNLP* is part of *SigmaKEE*. Relevant for our task is usage of WordNet and the corresponding SUMO-WordNet mappings [16]. Each token of the input text is substituted with its mapped concept. This might be a pitfall for multiple cases, where a concept from SUMO is represented with multiple words in natural language.

Second, automatic translation of input text to SUO-KIF was done by fine-tuning the GPT Curie model. We created a dataset containing pairs of different types of input text and their corresponding translation into SUO-KIF. It contains training data for the following SUO-KIF patterns and relations: *attribute*, *agent-patient*, *subclass*, *contains*, *part*. Table 1 shows the number of examples created for each type. The dataset is divided 80% for training and 20% to testing.

The translation is done in 2 steps (Listing 1 - 4). In the first step, text is translated to a "formal" format and then, the "formal" format is used for conversion to *SUO-KIF*:

```
{text   : "Deserts are dry"
 formal: "If ?X is an instance of desert, then ?X has the attribute dry"
 kif    : "(=> (instance ?X Desert) (attribute ?X Dry))"}
```

Listing 1: Training example for attribute

Algorithm 1 FactCheck

```
procedure FactCheck(txt, KB)
    MappedTxt ← AlignWithOntology(txt, KB)
    K ← TranslateToKif(MappedTxt)
    KB′ ← MergeOntologies(K, KB)
    hasProof, proofSteps ← RunProver(KB′)
    if hasProof == true then
        label ← TRUE
        explanation ← Verbalize(proofSteps)
    else
        K′ ← Negate(K)
        KB″ ← MergeOntologies(K′, KB)
        hasProof, proofSteps ← RunProver(KB″)
        if hasProof == true then
            label ← FALSE
            explanation ← Verbalize(proofSteps)
        label ← FALSE
        explanation ← Verbalize(proof)
    return label, explanation
```

```
{text   : "Apples and bananas are fruits"
 formal: "Apple and banana are subclasses of fruit",
 kif    : "(and (subclass Apple Fruit) (subclass Banana Fruit))"}
```

Listing 2: Training example for subclass relation

```
{text   : "Broccoli contains vitamins."
 formal: "If ?B is an instance of broccoli, then there exists ?V such
          that ?V is an instance of vitamin and ?V is part of ?B."
 kif    : "(=> (instance ?B Broccoli) (exists (?V)
               (and (instance ?V Vitamin) (part ?V ?B))))"}
```

Listing 3: Training example for contains/part relation

```
{text   : "Parasites damage cells."
 formal: "If ?P is an instance of parasite, then there exists ?D such
          that ?D is an instance of damaging, and ?P is the agent of
          ?D and cell is the patient of ?D."
 kif    : "(=> (instance ?P Parasite) (exists (?D)
            (and
                (instance ?D Destroying)
                (agent ?D ?P)
                (patient ?D Nutrient))))"}
```

Listing 4: Training example for agent-patient relation

Table 1
Dataset distribution by type

Type	Training	Testing
Agent-Patient	106	21
Attribute	150	30
Subclass	77	15
Contains-part	85	17
Total	418	83

Third, for theorem proving we relied on *Vampire* [1]. Since Vampire works on TPTP (Thousands of Problems for Theorem Provers) and thus, a conversion between SUO-KIF and TPTP was needed. *SigmaKEE* provides classes to make this conversion. Vampire attempts to refute a statement or claim, searching for a contradiction within a logical theory. It employs a cascade mode, which encompasses a series of increasingly specialized reasoning algorithms. This cascade mode enables Vampire to explore different strategies, including various forms of resolution, saturation, and quantifier elimination, to efficiently handle different types of logical problems. For practical reasons, we impose a time limit on Vampire. In cases where a conjecture contradicts the ontology, Vampire will often exceed the time limit, resulting in the absence of the output "Satisfiable", which signifies that the conjecture is in conflict with the ontology. Consequently, we explore proofs for both the conjecture and its negation in order to analyze the collective outcomes, as showed in Figure 2 and Algorithm 1.

Fourth, for *proof verbalisation*, Vampire outputs a lot of steps, which it used when searching for proof. For this case, only the axioms was selected and verbalized using the manually created formats in English for each term.

3. Running experiments

First, we compare GPT-based translations with SigmaNLP translations. Different from GPT, *Sigmakee* applies the agent-patient pattern. On the one hand, this pattern is adequate when the text represents processes. For instance *SigmaNLP* performs well in the agent-patient context (see Table 2). However, the translation does not explicitly indicate that "Salmon" is the cause of "Increasing"; rather, it suggests a "Increasing" process with two patients: "Salmon" and "Cholesterol". On the othe rhand, the pattern fails when the task includes the translation of predicates ”is”, ”have”, ”part”. For instance, given the claim "Broccoli contains vitamins.", the translation using the predicate *part* is more suitable than the agent-patient pattern (see Table 3).

The GPT-model was fine-tuned with 418 training examples (recall Table 1). For fine-tuning the Curie model we used 4 epochs, batch size of 1, learning rate multiplier (0.05, 0.1, or 0.2), while the parameter computing classification metrics was set on false. The cost for training is \$0.0030 / 1K tokens and the usage costs \$0.0120 / 1K tokens. On average, each example has 100 tokens, resulting in \$0.13 for training and \$0.1 for testing. On the 83 new examples used for testing, the accuracy was 0.84. This accuracy indicates the percentage of correct completions in the translation from natural language text to KIF. The tuned model achieved an accuracy of 0.96 for KIF translation and 0.82 for the formal model. By analysing errors, we observed that

Table 2
SigmaNLP (left) vs. GPT-based (right) translation of "Salmon increases cholesterol."

`(exists (?Salmon-1 ?increases-2 ?chol-3)`
`  (and`
`    (instance ?increases-2 Increasing)`
`    (patient ?increases-2 ?chol-3)`
`    (instance ?Salmon-1 Salmon)`
`    (patient ?increases-2 ?Salmon-1)`
`    (instance ?chol-3 Cholesterol)))`
	`(exists (?S ?L ?C)`
	`  (and`
	`    (instance ?S Salmon)`
	`    (instance ?L Increasing)`
	`    (instance ?C Cholesterol)`
	`    (agent ?L ?S)`
	`    (patient ?L ?C)))`

Table 3
SigmaNLP (left) vs. GPT-based (right) translation of "Broccoli contains vitamins."

`(exists (?vitamins-3 ?Broccoli-1)`
`  (and`
`    (instance ?Broccoli-1 Broccoli)`
`    (patient contains-2 ?Broccoli-1)`
`    (patient contains-2 ?vitamins-3)`
`    (instance contains-2 Attribute)`
`    (instance ?vitamins-3 Vitamin)))`
	`(=>`
	`  (instance ?C Broccoli)`
	`  (exists (?V)`
	`    (and`
	`      (instance ?V Vitamin)`
	`      (part ?V ?C))))`

most errors were related to the differences in variable names, which, in fact, do not affect the semantics. When testing with variables substituted with placeholders, the accuracy was 0.92. Out of 83 testing examples, 76 were translated correctly.

For testing fact-checking in the diet and nutrition domain, an domain ontology was built on top of SUMO. It includes the concepts required for the test set described in Table 4 and also concepts for various categories, including organic food, lipids, fibers, fatty acids, and axioms, aimed at confirming or contradicting test samples.

Next, we detail the computations for the claim: *Vegetables are healthy.* Our Diet ontology built on top of SUMO includes the following axioms:

```
(subclass Vegetable  FruitOrVegetable)
(instance Healthy BiologicalAttribute)
(=>
  (instance ?F FruitOrVegetable)
  (attribute ?F Healthy))
```

First, the text is aligned with ontology. That is, each token of the input sentence is not just mapped to a SUMO concept. Here the system computes the following output: *Vegetable Attribute Healthy* with the mappings *Vegetables = Vegetable*, *are = Attribute*, *a = [None]*, *healthy = Healthy.* Additionally to mapping, a semantic analysis was done, because the verb *are* was correctly mapped to *Attribute* and the article *a* is not mapped to anything, even though it has only one mapping to *AlphabeticCharacter* concept.

In the second step, the text is converted in to SUO-KIF format. The translator constructs the following intermediate format: *If ?V is an instance of vegetable, then ?V has attribute healthy* Based on it, the following SUO-KIF representation is obtained:

```
(=>
  (instance ?V Vegetable)
  (attribute ?V Healthy))
```

In the third step, the checker searches for a proof for the obtained query. For this, Vampire is run in cascade mode and a proof is found, which means the input text is true in relation with ontology.

In the final step, axioms from proof steps are automatically paraphrased in English by Sigma. SigmaKEE has a facility to convert SUO-KIF statements into natural language paraphrases in several different languages. This includes use of natural language templates for relations and logical operators and words or phrases for each term. These are built recursively for complex formulas.

```
1. for all a class , another class and an entity if the other class is an
   instance of class and the class is an instance of class , then if the
   entity is an instance of the class and the class is a subclass of the
   other class , then the entity is an instance of the other class
2. vegetable is an instance of class
3. vegetable is a subclass of fruit or vegetable
4. fruit or vegetable is an instance of class
5. for all an object if the object is an instance of fruit or vegetable ,
   then healthy is an attribute of the object
```

Listing 5: Automatic natural language praphrases using SigmaKEE

These paraphrases can be rephrased by GPT, with the following result:

```
1. If both a class (Class A) and another class (Class B) are instances
   of a common class (Class C), and an entity (Entity X) is an instance
   of Class A, and Class A is a subclass of Class B, then Entity X is
   also an instance of Class B.
2. Vegetable is an instance of a class (Class A).
3. Vegetable is a subclass of a class (Class B) that includes both
   fruits and vegetables.
4. Fruits or vegetables are instances of a class (Class C).
5. For any object (Object Y), if Object Y is an instance of a fruit or
   vegetable, then the object has a property "healthy."
```

Listing 6: Rephrasing the SigmaKEE paraphrases with GPT

Table 4 exemplifies some input text and the corresponding answer computed by the checker and also by GPT. In the first line, "All salt is unhealthy" is detected as false since in the knowledge base there are the following axioms "Some salt is unhealthy" and "Some salt is healthy". "Calcium strengthens bones" is detected as true since this knowledge appears in the ontology in the SUO-KIF format, and the translator has correctly converted the given text into SUO-KIF.

The processing time for fact-checking varies depending on the verdict. On average, "True" texts take 7.16 seconds, needing just one Vampire run. For "False" texts, the average time is

Table 4
Testing samples

No.	Sentence	Fact checker answer
1	Some sugar causes obesity	True
2	Calcium strengthens bones	True
3	Protein builds muscles	True
4	Broccoli contains vitamins	True
5	Fruits and vegetables are healthy	True
6	Salmon contains Omega3	True
7	Detox diets cleanse the blood	False
8	Some lipids are unhealthy	False
9	Eggs raises cholesterol	False
10	All salts are unhealthy	False
11	Some lipids are healthy	True
12	Coffee dehydrates you	Unknown
13	Athletes consume more protein	Unknown
14	Some detox diets cleanse the body	True

17.11 seconds, involving 2 runs where the first run might either time out or Vampire finishes processing, but the second run yields a proof. As for "Unknown" texts, they take 30-40 seconds on average, with 2 runs either reaching timeout or Vampire completing the processing.

4. Conclusion

We have shown initial experiments in the use of SUMO, Sigma, Vampire, and GPT Curie language model in an efficient method for verifying text validity. SUMO provides an expressive ontological framework that aids in stating precise meaning. Vampire uses SUMO's knowledge structure to systematically evaluate statement validity. We use a GPT model in translating the text into the SUO-KIF format, and then passing the result to Vampire. One output of this study is a dataset for training the GPT model. The GPT model facilitated text transformation for Vampire to evaluate, resulting in an advanced text credibility assessment system with fact-checking and misinformation detection.

Ongoing work consists of: assessing the system performance on larger sets of claims. The current running version of the tool, the diet ontology built on top of SUMO and the dataset used for fine-tuned the models are available at https://github.com/ldan22/fake-news-detector.git

Acknowledgments

A. Groza is supported by the grant CCCDI-UEFISCDI, project number PN-III-P2-2.1-PED-2021-2709, within PNCDI III of of the Ministry of Research, Innovation and Digitization, Romania

References

[1] L. Kovács, A. Voronkov, First-order theorem proving and Vampire, in: International Conference on Computer Aided Verification, Springer, 2013, pp. 1–35.

[2] B. Walenz, Y. Wu, S. Song, E. Sonmez, E. Wu, K. Wu, P. K. Agarwal, J. Yang, N. Hassan, A. Sultana, et al., Finding, monitoring, and checking claims computationally based on structured data, in: Computation+ Journalism Symposium, Citeseer, 2014.

[3] A. Tchechmedjiev, P. Fafalios, K. Boland, M. Gasquet, M. Zloch, B. Zapilko, S. Dietze, K. Todorov, ClaimsKG: A knowledge graph of fact-checked claims, in: The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18, Springer, 2019, pp. 309–324.

[4] L. Hu, T. Yang, L. Zhang, W. Zhong, D. Tang, C. Shi, N. Duan, M. Zhou, Compare to the knowledge: Graph neural fake news detection with external knowledge, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 754–763.

[5] A. Groza, Á. Katona, Fact-checking with explanations, in: 2022 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, 2022, pp. 150–157.

[6] F. Draicchio, A. Gangemi, V. Presutti, A. G. Nuzzolese, Fred: From natural language text to rdf and owl in one click, in: The Semantic Web: ESWC 2013 Satellite Events: Montpellier, France, May 26-30, 2013, Revised Selected Papers 10, Springer, 2013, pp. 263–267.

[7] S. Di, Y. Shen, L. Chen, Relation extraction via domain-aware transfer learning, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & Data Mining, 2019, pp. 1348–1357.

[8] I. Niles, A. Pease, Towards a Standard Upper Ontology, in: Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001, 2001, pp. 2–9.

[9] A. Pease, Standard Upper Ontology Knowledge Interchange Format, Unpublished language manual. Available at https://github.com/ontologyportal/sigmakee/blob/master/suo-kif.pdf (2009).

[10] C. Brown, A. Pease, J. Urban, Translating SUMO-K to Higher Order Set Theory, in: Prooeedings of Frontiers in Combining Systems (FroCoS), to appear, 2023.

[11] G. A. Miller, WordNet: a lexical database for English, Communications of the ACM 38 (1995) 39–41.

[12] A. Pease, S. Schulz, Knowledge engineering for large ontologies with Sigma KEE 3.0, in: International Joint Conference on Automated Reasoning, Springer, 2014, pp. 519–525.

[13] S. Schulz, E – A Brainiac Theorem Prover, Journal of AI Communications 15 (2002) 111–126.

[14] A. Steen, C. Benzmüller, The Higher-Order Prover Leo-III, CoRR abs/1802.02732 (2018). URL: http://arxiv.org/abs/1802.02732. arXiv:1802.02732.

[15] A. Pease, GitHub - ontologyportal/sigmanlp: NLP functionality on top of the Sigma system — github.com, https://github.com/ontologyportal/sigmanlp, 2023. [Accessed 19-Jul-2023].

[16] I. Niles, A. Pease, Mapping WordNet to the SUMO ontology, in: Proceedings of the IEEE International Knowledge Engineering Conference, 2003, pp. 23–26.