UnCommonSense in Action! Informative Negations for Commonsense Knowledge Bases

Hiba Arnaout*
harnaout@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Tuan-Phong Nguyen*
tuanphong@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Simon Razniewski
srazniew@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Gerhard Weikum
weikum@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

ABSTRACT

Knowledge bases about commonsense knowledge i.e., CSKBs, are crucial in applications such as search and question answering. Prominent CSKBs mostly focus on positive statements. In this paper we show that materializing important negations increases the usability of CSKBs. We present UnCommonSense, a web portal to explore informative negations about everyday concepts: (i) in a research-focused interface, users get a glimpse into results-per-steps of the methodology; (ii) in a trivia interface, users can browse fun negative trivia about concepts of their choice; and (iii) in a query interface, users can submit triple-pattern queries with explicit negated relations and compare results with significantly less relevant answers from the positive-only baseline. It can be accessed at: https://uncommonsense.mpi-inf.mpg.de/.

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**.

KEYWORDS

knowledge base; commonsense; negation

ACM Reference Format:
Hiba Arnaout, Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2023. UnCommonSense in Action! Informative Negations for Commonsense Knowledge Bases. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3539597.3573027

1 INTRODUCTION

Motivation and Problem. Commonsense knowledge is important for many applications like question answering and dialogue agents. This knowledge is often stored in triple form in commonsense knowledge bases (CSKBs), e.g., `(vinegar, HasProperty, acidic)`. Recently, we have seen a rising interest in constructing, curating, and querying such CSKBs. State-of-the-art CSKBs mainly focus on storing positive information, and collect little negative information. This poses a major limitation when downstream use has to decide whether absent information is false or missing [2]. With the open-world assumption (OWA) that most large-scale CSKBs postulate, an absent statement is *unknown*, regardless of whether it is false in reality. For instance, in Ascent [5], we know that *elephants* have tusks. This information is expressed in triple form as `(elephant, HasA, tusk)`. Due to the OWA, absent information about further properties are not known to be true or false for a fact, e.g., "*Are elephants nocturnal?*", "*Can they jump like many other land mammals?*". To empower downstream use cases, explicit assertion of negated statements can be very useful, e.g., `(elephant, NotHasProperty, nocturnal)`, `(elephant, NotCapableOf, jump)`.

Approach. The system demonstrated in this paper relies on the UnCommonSense method [1]. In a nutshell: given a target concept, e.g., *elephant*, comparable concepts are computed by employing structured taxonomies and latent similarity measures, e.g., other *wild animals* like *zebra, tiger, lion*. Among these comparable concepts the local closed-world assumption (LCWA) is postulated (where *some* parts of the KB are considered complete). Under this, any positive statement that holds for *at least one* of the comparable concepts and *not* the target concept is a candidate negative statement. Restricting the inferences to information about *comparable*, rather than *random* (e.g., *cake, newspaper*), concepts produces much more relevant candidate statements, in this case animal-related statements such as `(elephant, NotIsA, carnivore)`. Nonetheless, due to the incompleteness of large-scale CSKBs, inferred negations might be inaccurate, i.e., missing positives. For instance, `(elephant, HasA, eye)` is a missing statement from Ascent. Moreover, lightly-canonicalized CSKBs might contain multiple phrases indicating the same meaning, e.g., `(elephant, HasProperty, big animal)` but `(lion, HasProperty, large animal)`. This contradiction between information about target concept *elephant* and comparable concept *lion* will be overlooked during the previous inference step. To overcome these issues, we scrutinize the candidates against related statements in the input CSKB using sentence embeddings [6] and against a pre-trained language model (LM) as an external source of latent knowledge [4]. Finally, the potentially large set of candidates is ranked by computing informativeness using statistical scores, i.e., relative frequency within groups of comparable concepts (or

*Equal contribution.

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02.
https://doi.org/10.1145/3539597.3573027

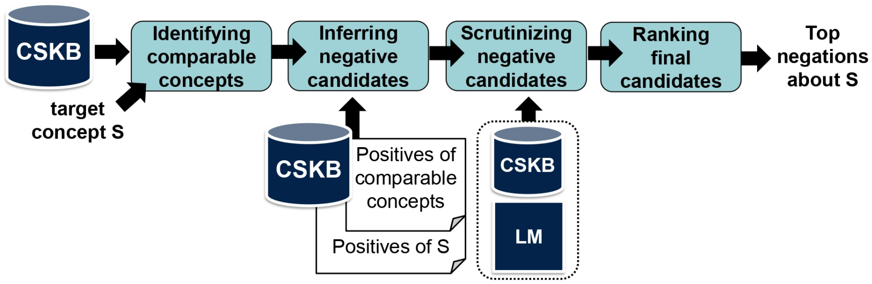

Figure 1: Overview of the UnCommonSense methodology.

type siblings). For example, while *elephant* cannot, 67% of its type siblings can jump. This procedure generates negations of significantly higher accuracy and informativeness than previous methods. Further details are in [1].

Demo. We present the UnCommonSense portal, where researchers can get a better understanding of the UnCommonSense method and where general users can browse top negative trivia about concepts of their choice and query the KB using explicit negated relations. The method is applicable to any other KB, e.g., ConceptNet [9], but we pick Ascent due to its higher coverage of statements in the ConceptNet schema. The demo is accessible at https://uncommonsense.mpi-inf.mpg.de/.

2 UNCOMMONSENSE

2.1 Method Description

The portal is based on the research published in [1] and shown in Figure 1. Given a target concept *S* (*elephant*) and a CSKB *K* (Ascent):

(1) We identify comparable concepts: To ensure *highly thematic* candidate negations, we begin by finding relevant context, i.e., parts of the CSKB where LCWA can be reasonably postulated. We opt for a combination of two complementary similarity measures: (i) using hypernymy relations [3], we collect concepts that share a class with *S*, e.g., *elephant*, *squirrel*, and *lion* are all taxonomic siblings under *land mammals*; then (ii) using latent representations [10], we compute cosine similarity between embeddings of *S* and unordered taxonomic siblings collected in (i), e.g., *lion* is closer to *elephant* than *squirrel*. We consider the closest-*k* siblings for the next step.

(2) We infer negative candidates: We generate the initial candidate set by computing the difference between *positives about the comparable concepts* and *positives about S*. For example, if `(HasA, tongue)`, `(CapableOf, jump)`, and `(HasProperty, carnivore)` hold for the comparable concept *lion* and `(HasA, tongue)` holds for *elephant*, the initial candidate negations are `(elephant, NotCapableOf, jump)` and `(elephant, NotHasProperty, carnivore)`.

(3) We scrutinize candidate negations: To remove candidates that might be *falsely* identified as negative, due to the incompleteness or lack of phrase disambiguation in *K*, we perform two kinds of plausibility checks: (i) in KB-based check, we compute the semantic-similarity [6] to get rid of any remaining contradictions that were not covered by the exact matching of the inference step. For instance, candidate `(elephant, NotHasProperty, large animal)` is eliminated due to the positive statement `(elephant, HasProperty, big animal)` in *K*; (ii) in LM-based check, we probe pre-trained LMs for any possibly missing information from *K*. For instance, "[MASK] eat grass." with prediction *elephant* at position 76 eliminates the candidate `(NotCapableOf, eat grass)`. Moreover, to remove candidates that are nonfactual, we eliminate any candidate that is identified as *too generic*, i.e., it holds for $\geq$ 5% of all concepts in *K*, e.g., `(HasProperty, amazing)` holds for 16% of all concepts.

(4) We score by informativeness: With a potentially large number of remaining candidates, ranking is crucial. We quantify the inforamtiveness of a certain candidate negation by how *uncommon it is amongst comparable concepts*, i.e., using statistical frequency within the group of siblings. For example, 3 out of 4 siblings of elephants *can jump* (while elephants cannot) and 1 out of 4 siblings *has hoof* (while elephants do not). Therefore, the former is more noteworthy.

2.2 Web Portal

Implementation. The web portal is implemented in Python using the Django framework[1]. We use nginx[2] as web sever and store our datasets in a PostgreSQL database. The demo is deployed on a Debian virtual machine at the Max Planck Institute for Informatics that has 8GB of RAM and 50GB storage.

Data and Method Hyperparameters. This demo covers all 8029 *primary* concepts in Ascent. The data follows the established ConceptNet schema i.e., canonicalized concepts and relations (both positive and negative). We initially produce 6.7 billion negations from assuming CWA, which are reduced to 47.2 million negations after LCWA is postulated, and lastly to 6.4 million final negations [3] after the scrutinizing step. We set the hyperparameters to their best-performing values as reported in [1], namely we set the number of siblings to 30, nonfactual statement threshold to 0.05, semantic similarity threshold to 0.7, and rank threshold of LM to 50.

[1]https://www.djangoproject.com/
[2]https://www.nginx.com/
[3]We release JSON-formatted data dumps at: https://uncommonsense.mpi-inf.mpg.de/download.

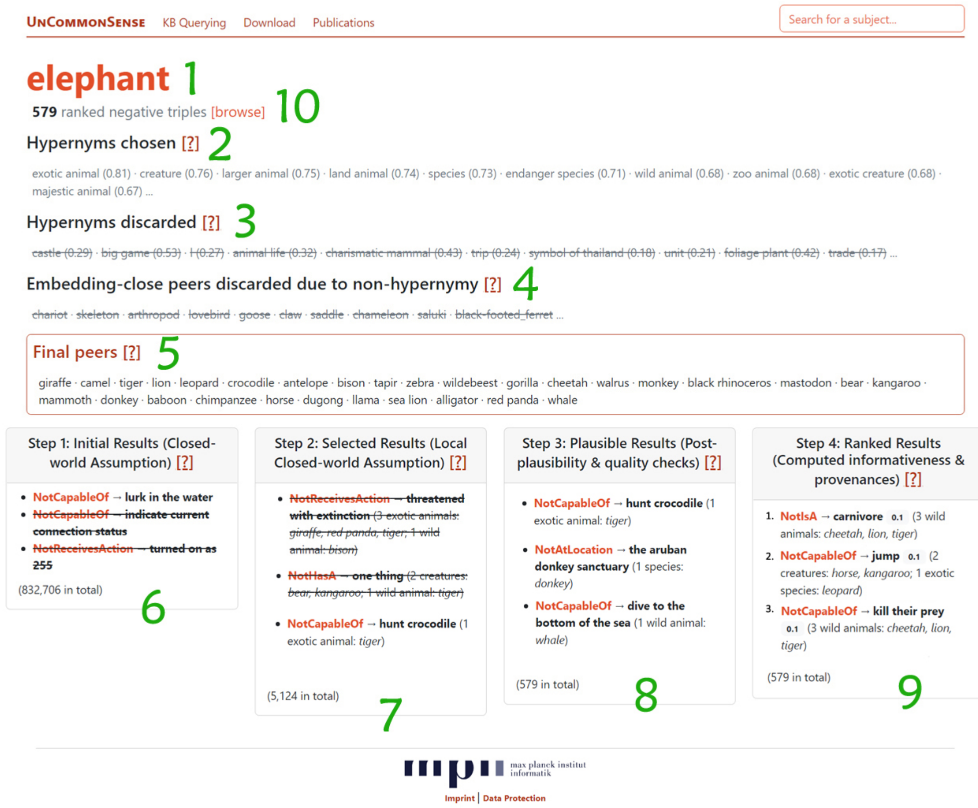

Figure 2: A look into how UnCommonSense collects informative negations about *elephant*.

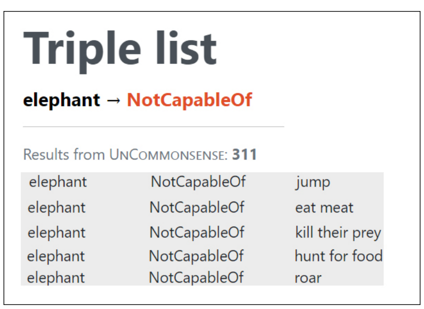

Figure 3: Trivia about *elephant* with relation `NotCapableOf`.

3 DEMONSTRATION EXPERIENCE

We show how users can interact with the web portal through three scenarios: 1) Understanding the methodology on collecting type siblings and inferring informative negations, 2) Exploring negative statements about a concept of choice, and 2) Querying the CSKB using explicit negative relations.

Scenario 1 - Inside UnCommonSense. The main function of UnCommonSense allows users to understand the various steps of the methodology (see Figure 2). This interface has a target concept field **(1)**, which takes an Ascent primary concept as input (i.e., "search for a subject" auto-completion field at top-right side). The ranked list of comparable concepts are displayed in **(5)**, e.g., *giraffe*. Users can refer to **(2)**, **(3)**, and **(4)** for a better understanding of their retrieval: high-confidence hypernyms are retained, e.g., *land animal*, while low-confidence or noisier ones are discarded, e.g., *trip*. Moreover, highly related concepts that are not taxonomic siblings and have been discarded are also displayed, e.g. *chariot* is related to *elephant*, but inconsistent by type. To give the user a feel of the full size of the negation sets at every step of the process, we display the the total number of results at the bottom of boxes **(6)**, **(7)**, **(8)**, and **(9)**. The initial step, see **(6)**, where the naive CWA is postulated, *elephant* received more than 832k candidate statements. This includes an overwhelming number of nonsensical negations, e.g., (`NotCapableOf, indicate current connection status`). The crossed out negations do not proceed to the next step, see **(7)**, where the LCWA is postulated using the comparable concepts. The number of candidates decreases by 162 times. These statements are thematic but not yet scrutinized for plausibility and quality. The crossed out negations here do not make it to the next step, see **(8)**, e.g., (`elephant, NotReceivesAction, threatened with extinction`) contradicts the positive statement in Ascent (`elephant, ReceivesAction, endangered`)[4] with semantic similarity of 0.72 between the two phrases. Finally, 579 plausible negations are ranked by informativeness, see **(9)**, e.g., (`elephant, NotCapableOf, jump`). More on the ranking metrics in [1]. For users interested in browsing the final negations and not

[4]https://ascentpp.mpi-inf.mpg.de/primary-subjects/elephant

Results from UnCommonSense & Ascent++

{ X | <X ; NotAtLocation ; the oven> }

Found **784** results for x, showing the first 20 results:

#	Subject	Score
1	cheeseburger	0.75
2	sandwich	0.70
3	yorkshire pudding	0.70
4	wheat	0.69
5	turnip	0.68
6	salad	0.67
7	babka	0.66
8	appetizer	0.65
9	strudel	0.65
10	tortilla chip	0.62

Results from Ascent++

{ X } \ { X | <X ; AtLocation ; the oven> }

Found **6798** results for x, showing the first 20 results:

#	Subject	Score
1	newsroom	0.00
2	erosion	0.00
3	mink	0.00
4	candidiasis	0.00
5	fishing rod	0.00
6	fix	0.00
7	fundamentalist	0.00
8	mathematics	0.00
9	balsam	0.00
10	pipe	0.00

Figure 4: Querying for food that *doesn't* require the usage of an oven.

particular steps, they can click on browse, see **(10)**, and will be directed to our next interface.

Scenario 2 - Knowledge Exploration. The user is an elementary school student who is fascinated by the animal kingdom. She has explored many positive statements about them in Ascent [5], namely about their properties and what they are capable of doing. Next, she would like to explore more on things she might *not* be aware of. By querying *elephant* in UnCommonSense (see Figure 3), she learns that, unlike other *exotic animals*[6] such as *leopard*, *elephants* cannot jump. She also learns that they do not attack preys. This made perfect sense since they also do not eat meat nor hunt.

Scenario 3 - Querying CSKB. The user is preparing for a meal and looking for ideas that do not require an oven, since he does not own one. He queries Ascent using UnCommonSense, i.e., Ascent plus explicit negations, by matching the triple-pattern `<?x NotAtLocation oven>` with explicit instances (pre-computed and scrutinized negated statements). Results are then sorted by descending informativeness. Top results are shown in Figure 4, e.g., *cheeseburger* and *salad*, all of which not requiring an oven. On the right side, one can also see that if the user were to query positive-only Ascent (baseline following CWA), 84% (6.7k) of all Ascent's concepts would be returned as plausible answers. The set is also *unranked*, hence the score=0, with many irrelevant answers, such as *newsroom* and *mathematics*.

4 RELATED WORK

Commonsense knowledge acquisition includes several large-scale projects including ConceptNet [9], Quasimodo [7], and Ascent [5]. Eventhough their main focus is positive knowledge, some of them allow the addition of negative statements. For example, ConceptNet [9] contains 6 pre-defined negative relations, e.g., NotIsA, NotCapableOf, which we adapt in our demo. The portion of negative statements in its latest version is less than 2%. Quasimodo [7] contains 350k statements with negative *polarity*, yet many have quality issues, e.g., `(elephant, NotCapableOf, quit smoking)`. On actively collecting informative commonsense negations, NegatER [8] proposes using graph-based triple corruption and fine-tuned LMs to discover meaningful negations. A detailed comparison to this work is in [1].

5 CONCLUSION

In this demo, we present UnCommonSense, a web portal for inspecting informative negative statements about everyday concepts in a large-scale commonsense knowledge base.

ACKNOWLEDGMENTS

Funded by the German Research Foundation - Project 453095897 - "Negative Knowledge at Web Scale".

REFERENCES

[1] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z Pan. 2022. UnCommonSense: Informative Negative Knowledge about Everyday Concepts. In *CIKM*.

[2] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. 2006. Inconsistencies, Negations and Changes in Ontologies. In *AAAI*.

[3] Sven Hertling and Heiko Paulheim. 2017. WebIsALOD: Providing Hypernymy Relations Extracted from the Web as Linked Open Data. In *ISWC*.

[4] Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen-tau Yih, Hao Ma, and Madian Khabsa. 2020. Language Models as Fact Checkers?. In *ACL, FEVER workshop*.

[5] Tuan-Phong Nguyen, Simon Razniewski, Julien Romero, and Gerhard Weikum. 2022. Refined Commonsense Knowledge from Large-Scale Web Contents. *TKDE* (2022).

[6] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP*.

[7] Julien Romero, Simon Razniewski, Koninika Pal, Jeff Z. Pan, Archit Sakhadeo, and Gerhard Weikum. 2019. Commonsense Properties from Query Logs and Question Answering Forums. In *CIKM*.

[8] Tara Safavi, Jing Zhu, and Danai Koutra. 2021. NegatER: Unsupervised Discovery of Negatives in Commonsense Knowledge Bases. In *EMNLP*.

[9] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*.

[10] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia. In *EMNLP*.

[5] https://ascentpp.mpi-inf.mpg.de

[6] Explanations are omitted for readability