Neguess: Wikidata-entity guessing game with negative clues

Aditya Bikram Biswas, Hiba Arnaout, and Simon Razniewski

Max Planck Institute for Informatics, Saarland, Germany
{adbiswas,harnaout,srazniew}@mpi-inf.mpg.de

Abstract. We present Neguess, an entity-guessing game with unique emphasis on challenging *negative* clues. The clues have been automatically generated using the peer-based negation inference methodology [3]. The game can be used i) as an entertaining way to familiarize participants with the novel area of explicit negative knowledge in open-world knowledge bases; and ii) has the potential to be adopted in pedagogical approaches, like game-based teaching practices.
The demo is available at: https://neguess.mpi-inf.mpg.de.

Keywords: Negation · RDF · Knowledge Bases · Wikidata

1 Introduction

Knowledge bases (KBs) operate under the open-world assumption (OWA), meaning that statements asserted in them, in the form of *(subject; predicate; object)* are *true*, like *(Denmark; member of; European Union)*, and statements not asserted are *unknown*, like *(Iceland; member of; European Union)*. Given that existing web-scale KBs are far from complete, it is not realistic to assume that absent information is false. It is also not realistic to add every possible negation to the KB (e.g., more than 280k actors with no Oscars[1]). For this reason, we have seen a rising interest in augmenting open-world KBs with *useful* negative statements. In [3], interesting negations are inferred about a given entity based on observations made on similar entities. For instance, *Iceland* is a European country like *Denmark*, however, the former does not have the statement asserting its membership in the *European Union*. In [5], an anti-KB containing common factual mistakes has been built, through mining Wikipedia edit logs. In [8], the focus is on obtaining meaningful negative information in commonsense KBs.

Neguess (short for "entity-**guess**ing game with **neg**ative clues") builds on the methodology introduced in [3], and shows multiple-choice guessing cards, where the clues are entirely negated assertions, i.e., *properties not satisfied by the correct answer*. For every guessing card, i) it picks a random entity as the right answer ii) retrieves similar entities for wrong answers, (e.g., other countries

[1] https://w.wiki/3ZB9

from the same continent), and iii) compiles challenging negative clues that are mostly, or fully, applicable to the correct entity.

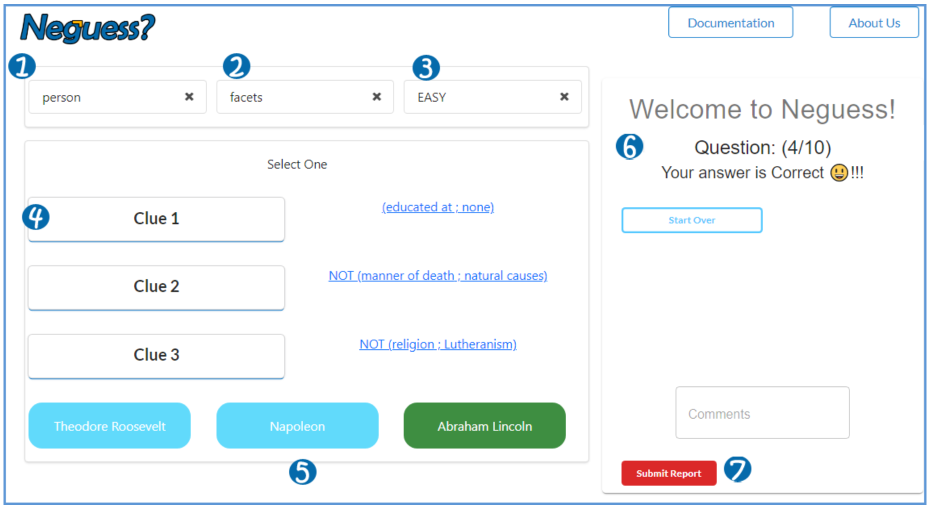

Fig. 1. An overview of a Neguess round.

Peer-based negation inference. Neguess relies on the so-called *peer-based inference methodology* [3] to compile interesting negative statements. In particular, given an entity e from KB, the method:

1. Collects e's peers using a predefined similarity function (e.g., embedding-based similarity [10]). Peer grouping is based on three different functions (i) structured facets of entity e [4]. For instance, people sharing the same *occupation, nationality, similar field of work*, (ii) graph-based measures like *distance or connectivity* [6,7], expressed as the number of predicate-object pairs two entities share, and (iii) cosine similarity based on Wikipedia embeddings [10].
2. Produces a set of candidate negations (i.e., statements that are asserted in KB for at least one peer, but not for e).
3. Scores the set of candidates using various ranking metrics (e.g., frequency, unexpectedness, etc.). The need for ranking stems from the very large set of correct negative statements inferred. For example, a person-entity is *not* married to millions of people.

Further details about the methodology are in [3] and [2].

2 System Overview

Neguess cards. We make use of the method, described briefly in Section 1, to generate three challenging negative clues for Wikidata [9] entities from diverse types. A challenging negative clue is equivalent to an inferred negative statement with a high score. Figure 1 shows a sample card game. Players can pick the type of entities to guess (1); pick the similarity function to be used for collecting the peers (i.e., the multiple options) (2); and pick the difficulty of the clues (3). Here, difficulty reflects how unique are the clues to the correct answer. For instance, the multiple options are the famous world leaders *Roosevelt*, *Napoleon*, and *Lincoln* (5). They have been chosen as peers because they share the occupation "statesperson" (relying on structured facets of the subject [4]). In this case, the difficulty is set to *easy* and is reflected as 2/3 of the clues are unique to the correct answer, making it somehow more distinguishable. The clues are shown in two possible structured forms: i) *(p; none)*, e.g., *(educated at; none)* and ii) *NOT (p; o)*, e.g., *NOT (manner of death; natural causes)*. Unlike the others, *Lincoln* was shot in the famous theatre incident. He is also known as one of few *American* presidents with no formal education. The third clue does not contribute to the answer and is there to confuse the player, as all of them are *not Lutheran*. Moreover, players can track their progress in the game (6). Finally, players can report a card if it contains any incorrect negations or technical problems (7).

Implementation and web interface. The Neguess front-end or the web interface is developed using React JS [2], a JavaScript library to build user interfaces. The back-end is developed using Spring Boot [3] with JAVA running on Apache Tomcat server. We use PostgreSQL to create and manage our database. It stores around 3m negative clues about 40k popular Wikidata entities from 5 diverse types, namely, people, countries, literature work, organizations, and businesses. Neguess runs on a server with capacity 1 TB and a 8 GB RAM. The average speed of retrieving a guessing card is 3 s.

3 Demonstration Experience

Can you neguess? A player, who is very confident of her knowledge about countries, chooses the type "country" with difficulty "hard". She gets two consecutive cards, shown in Figure 2. The focus of the first card is countries of *central and south America*. She knows that the main emergency number in *Argentina* is 911, so she immediately disregard this country as the answer. She is certain that *Chile* does not share a border with *Columbia*, so *Chile* is a likely option as the card's answer. She clicks on the *Central American Bank for Economic Integration* and is lead to the Wikidata (and then Wikipedia) page of the institution. She finds out that *Guatemala* is one of the founding members. She clicks on *Chile* as her final (and correct) pick!
Her second card covers *Gulf* countries. She does not know which electric plug type these countries use, so this clue was not helpful to her. She is certain that none are in *Africa*. However, the first clue confused her the most. They are all countries known for their oil production, so how is it possible that (at least)

[2] https://reactjs.org/
[3] https://spring.io/projects/spring-boot

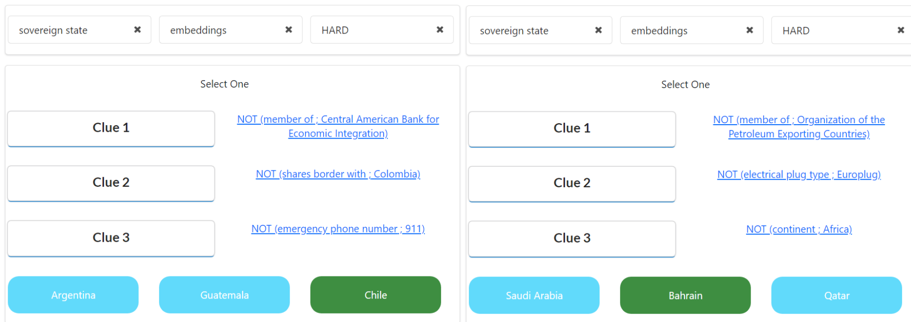

Fig. 2. Two Neguess cards about countries.

one of them is not a member of *OPEC*? Hesitant about these clues, she picks *Bahrain* as a lucky guess. She answered correctly, but still not sure which clues are applicable. She checks *Bahrain*'s Wikidata page and does not find the *OPEC* membership. She googles the fact and learns that *Bahrain* is *not* a member of *OPEC* but OPEC+, a division for non-OPEC countries which export crude oil.

Beyond fun and games. Neguess can be used to understand the peer-based negation inference method it is based on [3]. By choosing embeddings [10] as a similarity function for instance, countries which have latent shared information start to appear together in a guessing card (for instance, *U.S.* and *Russia*). On the other hand, when the peering function is changed to graph-based measures (computed as *p-o* pair combinations entities share), countries which share a lot of geographical information start to appear together (for instance, *U.S.* and *Mexico*). In addition, Neguess could be used as an entertaining tool to find and understand modelling issues in Wikidata. One clue for a person card, including three famous computer scientists, is *NOT (field of work; computer science)*. This is clearly an incorrect card that must be reported. Moreover, digging deeper into the reason this card was generated, we find that two of these computer scientists had *Informatics* and *Information Technology* as their field of work. Finally, we use the game to gather feedback on the correctness of the inferred negation. A player can flag a card and add her comment on the informativeness or correctness of the clues. In future work, we would like to give players more opportunity to give feedback (e.g., flagging individual clues, or correcting clues if they wish to).

4 Discussion

In order to compile the set of negative clues for this game, the peer-based methodology infer useful negative statements by assuming completeness in parts of the KBs, namely within peer groups. Although this approach outperformed baselines methods in [3], inferences (i.e., clues) may still be incorrect. At the moment, we allow players to flag cards as incorrect, and would like to use this feedback in the future to affect the display/disregard of erroneous cards. In addition, we understand that wrapping up the negative statements in a game setting would not allow users to inspect specific entities of interest. Another platform, built upon the same research work, has been published recently, where users can explore useful negation through an entity summarization and structured question answering interfaces [1].

Acknowledgments This work is supported by the German Science Foundation (DFG: Deutsche Forschungsgemeinschaft) by grant 4530095897: “Negative Knowledge at Web Scale”.

References

1. Arnaout, H., Razniewski, S., Weikum, G., Pan, J.Z.: Wikinegata: a knowledge base with interesting negative statements. PVLDB (2021)
2. Arnaout, H., et al.: Negative knowledge for open-world wikidata. Wiki Workshop at WWW (2021)
3. Arnaout, H., Razniewski, S., Weikum, G.: Enriching knowledge bases with interesting negative statements. In: AKBC (2020)
4. Balaraman, V., et al.: Recoin: Relative completeness in Wikidata. Wiki Workshop at WWW (2018)
5. Karagiannis, G., et al.: Mining an ”anti-knowledge base” from Wikipedia updates with applications to fact checking and beyond. In: VLDB (2019)
6. Petrova, A., et al.: Entity comparison in rdf graphs.. ISWC (2017)
7. Ponza M., Ferragina P., C.S.: A two-stage framework for computing entity relatedness in Wikipedia. CIKM (2017)
8. Safavi, T., Koutra, D.: Generating negative commonsense knowledge. KR2ML (2020)
9. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledge base. CACM (2014)
10. Yamada, I., et al.: Wikipedia2Vec: An optimized tool for learning embeddings of words and entities from Wikipedia. EMNLP (2020)