Web Semantics: Science, Services and Agents on the World Wide Web 71 (2021) 100661

Contents lists available at ScienceDirect

Web Semantics: Science, Services and Agents on the World Wide Web

journal homepage: www.elsevier.com/locate/websem

Negative statements considered useful

Hiba Arnaout [a,*], Simon Razniewski [a], Gerhard Weikum [a], Jeff Z. Pan [b]

[a] Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken 66123, Germany
[b] School of Informatics, The University of Edinburgh, Informatics Forum, Edinburgh EH8 9AB, Scotland, United Kingdom

ARTICLE INFO

Article history:
Received 7 September 2020
Received in revised form 3 September 2021
Accepted 15 September 2021
Available online 21 September 2021

Keywords:
Knowledge bases
Negative knowledge
Information extraction
Statistical inference
Ranking

ABSTRACT

Knowledge bases (KBs) about notable entities and their properties are an important asset in applications such as search, question answering and dialog. All popular KBs capture virtually only positive statements, and abstain from taking any stance on statements not stored in the KB. This paper makes the case for explicitly stating salient statements that do *not* hold. Negative statements are useful to overcome limitations of question answering systems that are mainly geared for positive questions; they can also contribute to informative summaries of entities. Due to the abundance of such invalid statements, any effort to compile them needs to address ranking by saliency. We present a statistical inference method for compiling and ranking negative statements, based on expectations from positive statements of related entities in peer groups. Experimental results, with a variety of datasets, show that the method can effectively discover notable negative statements, and extrinsic studies underline their usefulness for entity summarization. Datasets and code are released as resources for further research.

1. Introduction

Motivation and Problem. Structured knowledge is crucial in a range of applications like question answering, dialog agents, and recommendation systems. The required knowledge is usually stored in KBs, and recent years have seen a rise of interest in KB construction, querying and maintenance, with notable projects being Wikidata [1], DBpedia [2], Yago [3], or the Google Knowledge Graph [4]. These KBs store positive statements such as *"Renée Zellweger won the 2020 Oscar for the best actress"*, and are a key asset for many knowledge-intensive AI applications.

A major limitation of all these KBs is their inability to deal with negative information [5]. At present, most major KBs only contain positive statements, whereas statements such as that *"Tom Cruise did not win an Oscar"* could only be inferred with the major assumption that the KB is complete — the so-called *closed-world assumption* (CWA). Yet as KBs are only pragmatic collections of positive statements, the CWA is not realistic to assume, and there remains uncertainty whether statements not contained in a KBs are false, or truth is merely unknown to the KB.

Not being able to formally distinguish whether a statement is false or unknown poses challenges in a variety of applications. In medicine, for instance, it is important to distinguish between knowing about the absence of a biochemical reaction between substances, and not knowing about its existence at all. In corporate integrity, it is important to know whether a person was never employed by a certain competitor, while in anti-corruption investigations, absence of family relations needs to be ascertained. In data science and machine learning, on-the-spot counterexamples are important to ensure the correctness of learned extraction patterns and associations.

State of the Art and its Limitations. Absence of explicit negative knowledge has consequences for usage of KBs: for instance, today's *question answering* (QA) systems are well geared for positive questions, and questions where exactly one answer should be returned (e.g., quiz questions or reading comprehension tasks) [6,7]. In contrast, for answering negative questions like *"Actors without Oscars"*, QA systems lack a data basis. Similarly, they struggle with positive questions that have no answer, like *"Children of Emmanuel Macron"*, too often still returning a best-effort answer even if it is incorrect. Materialized negative information would allow a better treatment of both cases.

Approach and Contribution. In this paper, we make the case that important negative knowledge should be explicitly materialized. We motivate this selective materialization with the challenge of overseeing a near-infinite space of false statements, and with the importance of explicit negation in search and question answering.

We consider three classes of negative statements: (i) grounded negative statements *"Tom Cruise is not a British citizen"*, (ii) conditional negative statements *"Tom Cruise has not won an award from the Oscar categories"* and (iii) universal negative statements *"Tom Cruise is not member of any political party"*. In a nutshell,

* Corresponding author.
E-mail addresses: harnaout@mpi-inf.mpg.de (H. Arnaout), srazniew@mpi-inf.mpg.de (S. Razniewski), weikum@mpi-inf.mpg.de (G. Weikum), j.z.pan@ed.ac.uk (J.Z. Pan).

https://doi.org/10.1016/j.websem.2021.100661
1570-8268/

given a KB and an entity e, we select highly related entities to e (we call them *peers*). We then use these peers to derive positive expectations about e, where the absence of these expectations might be interesting for e. In this approach, we are assuming completeness within a group of peers. More precisely, if the KB does not mention the *Nobel Prize in Physics* as an award won by *Stephen Hawking*, but does mention it for at least one of his peers, it is assumed to be false for *Hawking*, and not a missing statement. This is followed by a ranking step where we use predicate and object prominence, frequency, and textual context in a learning-to-rank model.

The salient contributions of this paper are:

1. We make the first comprehensive case for materializing *useful* negative statements, and formalize important classes of such statements.
2. We present a judiciously designed method for collecting and ranking negative statements based on knowledge about related entities.
3. We show the usefulness of our models in use cases like entity summarization, decision support, and question answering.
   Experimental datasets and code are released as resources for further research.[1]

The present article extends the earlier conference publication [8] in several directions:

1. We extend the statistical inference to ordered sets of related entities, thereby removing the need to select a single peer set, and obtaining finer-grained contextualizations of negative statements (Section 5);
2. To bridge the gap between overly fine-grained grounded negative statements and coarse universal negative statements, we introduce a third notion of negative statement, *conditional negative statements*, and show how to compute them post-hoc (Section 6);
3. We evaluate the value of negative statements in an additional use case, with hotels from Booking.com (Section 8).

2. State of the art

2.1. Negation in existing knowledge bases

Deleted Statements. Statements that were once part of a KB but got subsequently deleted are promising candidates for negative information [9]. As an example, we studied deleted statements between two Wikidata versions from 1/2017 and 1/2018, focusing in particular on statements for people (close to 0.5 m deleted statements). On a random sample of 1k deleted statements, we found that over 82% were just caused by ontology modifications, granularity changes, rewordings, or prefix modifications. Another 15% were statements that were actually restored a year later, so presumably reflected erroneous deletions. The remaining 3% represented actual negation, yet we found them to be rarely noteworthy, i.e., presenting mostly things like corrections of birth dates or location updates reflecting geopolitical changes.

In Wikidata, erroneous changes can also be directly recorded via the deprecated rank feature [10]. Yet again we found that this mostly relates to errors coming from various import sources, and did not concern the active collection of interesting negations, as advocated in this article.

Count and Negated Predicates. Another way of expressing negation is via counts matching with instances, for instance, storing 5 children statements for *Trump* and numerical statement `(number of children; 5)` allow to infer that anyone else is not a child of *Trump*. Yet while such count predicates exist in popular KBs, none of them has a formal way of dealing with these, especially concerning linking them to instance-based predicates [11].

Moreover, some KBs contain relations that carry a negative meaning. For example, DBpedia has predicates like *carrier never available* (for phones), or *never exceed alt* (for airplanes), Knowlife [12] contains medical predicates like *is not caused by* and *is not healed by*, and Wikidata contains *does not have part* and *different from*. Yet these present very specific pieces of knowledge, and do not generalize. Although there have been discussions to extend the Wikidata data model to allow generic opposites,[2] these have not been worked out so far.

Wikidata No-Values. Wikidata can capture statements about *universal absence* via the "no-value" symbol [13]. This allows KB editors to add a statement where the object is empty. For example, what we express as $\neg\exists x$`(Angela Merkel; child; x)`, the current version of Wikidata allows to be expressed as `(Angela Merkel; child; no-value)`.[3] As of 8/2021, there exist 135k of such "no-value" statements, yet only used in narrow domains. For instance, 53% of these statements come for just two properties *country* (used almost exclusively for geographic features in Antarctica), and *follows* (indicating that an artwork is not a sequel).

2.2. Negation in logics and data management

Negation has a long history in logics and data management. Early database paradigms usually employed the closed-world assumption (CWA), i.e., assumed that all statements not stated to be true were false [14,15]. On the Semantic Web and for KBs, in contrast, the open-world assumption (OWA) has become the standard. The OWA asserts that the truth of statements not stated explicitly is unknown. Both semantics represent somewhat extreme positions, as in practice it is neither conceivable that all statements not contained in a KB are false, nor is it useful to consider the truth of all of them as unknown, since in many cases statements not contained in KBs are indeed not there because they are known to be false. Between these two assumptions, there is also the so-called local (partial) closed-world assumption [16], where open-world assumption is used in general, while the closed-world assumption can be applied to some predicates (classes or properties).

In limited domains, logical rules and constraints, such as Description Logics [17,18] or OWL, can be used to derive negative statements. An example is the statement that every person has only one birth place, which allows to deduce with certainty that a given person who was born in *France* was not born in *Italy*. OWL also allows to explicitly assert negative statements [19], yet so far is predominantly used as ontology description language and for inferring intensional knowledge, not for extensional information (i.e., instances of classes and relations), with a few exceptions, like the rewriting based approach to instance retrieval for negated concepts, based on the notion of inconsistency-based first-order-rewritability [20]. Different levels of negations and inconsistencies in Description Logic-based ontologies are proposed in a general framework [5].

In [21,22], a thorough study on negative information in the Resource Description Framework (RDF) argues in favor of explicit negation. In particular, it makes the point that any knowledge

[1] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/knowledge-base-recall/interesting-negations-in-kbs/.

[2] https://www.wikidata.org/wiki/Wikidata:Property_proposal/fails_compliance_with.

[3] https://www.wikidata.org/wiki/Q567.

representation formalism must be able to deal with *informative* negative information, on top of informative positive information. The authors then propose ERDF (extended RDF), where an ERDF triple can be either positive or negative. The framework also distinguishes between two kinds of negation: weak ("she doesn't like snow") and strong ("she dislikes snow"). The former is denoted using the $\sim$ symbol, and the latter using the $\neg$ symbol.

The notion of `noValue` in RDF was introduced in [23]. It has been recently adapted in [24] for representing no-value information in RDF and incorporating such information into query answering. The intuition behind it is to distinguish whether a result set of a SPARQL query is empty due to lack of information or actual negation.

The AMIE framework [25] employed rule mining to predict the completeness of properties for given entities. This corresponds to learning whether the CWA holds in a local part of the KB, inferring that all absent values for a subject–predicate pair are false. For our task, this could be a building block, but it does not address the inference of *useful* negative statements.

RuDiK [26] is a rule mining system that can learn rules with negative atoms in rule heads (e.g., people born in *Germany* cannot be *U.S.* president). This could be utilized towards predicting negative statements. Unfortunately, such rules predict way too many – correct, but uninformative – negative statements, essentially enumerating a huge set of people who are not *U.S.* presidents. The same work also proposed a precision-oriented variant of the CWA that assumes negation only if subject and object are connected by at least one other relation. Unfortunately, this condition is rarely met in interesting cases. For instance, most of the negative statements in Table 6 have alternative connections between subject and object in Wikidata.

2.3. Related areas

Linguistics and Textual Information Extraction (IE). Negation is an important feature of human language [27]. While there exists a variety of ways to express negation, state-of-the-art methods are able to detect quite reliably whether a segment of text is negated or not [28,29]. There is also work on using knowledge graphs to help detect false statements in texts, such as news [30].

A body of work targets negation in medical data and health records. In [31], a supervised system for detecting negation, speculation and their scope in biomedical data is developed, based on the annotated BioScope corpus [32]. In [33], the focus is on negations via the keyword "not". The challenge here is the right scoping, e.g., "Examination could not be performed due to the Aphasia" does not negate the medical observation that the patient has Aphasia. In [34], a rule-based approach based on NegEx [35], and a vocabulary-based approach for prefix detection were introduced. PreNex [36] also deals with negation prefixes. The authors propose to break terms into prefixes and root words to identify this kind of negation. They rely on a pattern matching approach over medical documents.

In [37], an anti-knowledge base containing negations is mined from Wikipedia change logs, with the focus however being again on factual mistakes, and precision, not interestingness, is employed as main evaluation metric. In [38], the focus is to obtain meaningful negative samples for augmenting commonsense KBs. We explore text extraction in more details in the proposed *pattern-based query log extraction* method in our earlier conference publication [8].

Statistical Inference and KB Completion. As text extraction often has limitations, data mining and machine learning are frequently used on top of extracted or user-built KBs, in order to detect interesting patterns in existing data, or in order to predict statements not yet contained in a KB. There exist at least three popular approaches, rule mining, tensor factorization, and vector space embeddings [39]. Rule mining is an established, interpretable technique for pattern discovery in structured data, and has been successfully applied to KBs for instance by the AMIE system [40]. Tensor factorization and vector space embeddings are latent models, i.e., they discover hidden commonalities by learning low-dimensional feature vectors [41]. To date, all these approaches only discover positive statements. On the other hand, if one considers logical entailments as a means to enhance such rule mining and latent model based approaches, such as in an iterative manner [42], negative statements in theory can be discovered with the help of disjoint axioms; however, the quality of knowledge graph completion methods still have room for improvement. Recently, an inference model has been proposed to build a knowledge graph with commonsense contradictions [43], like "Wearing a mask is seen as responsible" is the contradiction of "Not wearing a mask is seen as carefree".

Ranking KB Statements. In applications such as entity summarization over web-scale KBs, returned result sets are often very large. Ranking statements is a core task in managing access to KBs, with techniques often combining generative language-models for queries on weighted and labeled graphs [44–46]. In [47], the authors propose a variety of functions to rank values of type-like predicates. These algorithms include retrieving entity-related texts, binary classifiers with textual features, and counting word occurrences. In [48], the focus is on identifying the informativeness of statements within the context of the query, by exploiting deep learning techniques. In this work, applications such as entity summarization returns a set of *negative* statements. To assign each statement a relevance score, we use a mixture of the metrics that are usually used for ranking positive statements (e.g., frequency of property), and metrics that are specific for negative statements (e.g., unexpectedness).

3. Model

For the remainder we assume that a KB is a set of statements, each being a triple $(s; p; o)$ of subject s, property p and object o.

Let K^i be an (imaginary) ideal KB that perfectly represents reality, i.e., contains exactly those statements that hold in reality. Under the OWA, (practically) available KBs, K^a contains correct statements, but may be incomplete, so the condition $K^a \subseteq K^i$ holds, but not the converse [49]. We distinguish two forms of negative statements.

Definition 1 (*Negative Statements*)**.**

1. A grounded negative statement $\neg(s, p, o)$ is satisfied if $(s, p, o) \notin K^i$.
2. A universally negative statement $\neg\exists o : (s, p, o)$ is satisfied if there exists no o such that $(s; p; o) \in K^i$.

An example of a grounded negative statement is that *"Bruce Willis was not born in the U.S"*., and is expressed as $\neg$`(Bruce Willis; born in; U.S.)`. An example of a universally negative statement is that *"Leonardo DiCaprio has never been married"*, expressed as $\neg\exists o$ `:(Leonardo DiCaprio; spouse; o)`. Both types of negative statements represent standard logical constructs, and could also be expressed in the OWL ontology language. Grounded negative statements could be expressed via negative property statements (e.g., `NegativeObjectPropertyAssertion (:born In :Bruce Willis :U.S.)`), while universally negative statements could be expressed via `ObjectAllValuesFrom` or `owl:complementOf` [13] (e.g., `ClassAssertion (ObjectAllValuesFrom (:spouse owl:Nothing) :Leonardo Dicaprio)`).

Table 1
Discovering candidate statements for *Brad Pitt* from one peer group with 3 peers.

Russel Crowe	Tom Hanks	Denzel Washington	Brad Pitt	Candidate statements
(award; Oscar for Best Actor)	(award; Oscar for Best Actor)	(award; Oscar for Best Actor)	(citizen; U.S.)	¬(award; Oscar for Best Actor), 1.0
(citizen; New Zealand)	(citizen; U.S.)	(citizen; U.S.)	(child; x)	¬(occup.; screenwriter), 1.0
(child; y)	(child; z)	(child; u)		$\neg\exists l$(instagram; l), 0.67
(occup.; screenwriter)	(occup.; screenwriter)	(occup.; screenwriter)		$\neg w$(convicted; w), 0.33
(convicted; v)	(instagram; r)	(instagram; f)		¬(citizen; New Zealand), 0.33
(instagram; t)				

Without further constraints, for these classes of negative statements, checking that there is no conflict with a positive statement is trivial. In the presence of further constraints or entailment regimes, one could resort to (in)consistency checking services [17,50,51].

Yet compiling negative statements faces two other challenges. First, being not in conflict with positive statements is a necessary but not a sufficient condition for correctness of negation, due to the OWA. In particular, K^i is only a virtual construct, so methods to derive correct negative statements have to rely on the limited positive information contained in K^a, or utilize external evidence, e.g., from text. Second, the set of correct negative statements is near-infinite, especially for grounded negative statements. Thus, unlike for positive statements, negative statement construction/extraction needs a tight coupling with ranking methods.

Research Problem 1. Given an entity e, compile a ranked list of useful grounded negative and universally negative statements.

4. Peer-based statistical inference

We next present a method to derive useful negative statements by combining information from similar entities ("peers") with supervised calibration of ranking heuristics. The idea is that peers that are similar to a given entity can give expectations on relevant statements that *should* hold for the entity. For instance, several entities similar to the physicist *Stephen Hawking* have won the *Nobel in Physics*. We may thus conclude that him not winning this prize could be an especially useful statement. Yet related entities also share other traits, e.g., many famous physicists are *U.S.* citizens, while *Hawking* is *British*. We thus need to devise ranking methods that take into account various clues such as frequency, importance, unexpectedness, etc.

Peer-based Candidate Retrieval. To scale the method to web-scale KBs, in the first stage, we compute a candidate set of negative statements using the CWA on certain parts of the KB, to be ranked in the second stage. Given a subject e, we proceed in three steps:

1. *Obtain peers:* We collect entities that set expectations for statements that e could have, the so-called *peer groups* of e. These groups can be based on (i) structured facets of the subject [52], such as *occupation, nationality*, or *field of work* for people, or classes/types for other entities, (ii) graph-based measures such as distance or connectivity [53], or (iii) entity embeddings such as TransE [54], possibly in combination with clustering, thus reflecting latent similarity.
2. *Count statements:* We count the relative frequency of all predicate–object pairs (i.e., `(_,p,o)`) and predicates (i.e., `(_,p,_)`) within the peer groups, and retain the maxima, if candidates occur in several groups. This way, statements are retained if they occur frequently in at least one of the possibly orthogonal peer groups.
3. *Subtract positives:* We remove those predicate–object pairs and predicates that exist for e.

Algorithm 1 shows the full procedure of the peer-based inference method. In line 2, groups of peers $P[]$ are selected based on some blackbox function *peer_groups*.

$P = [P_1, \ldots, P_n]$, with $n >= 1$.

Every group P_i is a set of peers, defined as follows.

$P_i = \{pe_1, \ldots, pe_m\}$, with $m <= s$.

Subsequently, for each peer group, it collects all the positive information that these peers have (line 7 and 8), and stores them as a list of candidate statements.

$candidates = \{st_1, \ldots, st_w\}$.

A statement st_j in *candidates* is either a predicate P or a predicate–object pair PO. After collecting information about the peers, the loop at line 11 iterates over the list of unique statements *ucandidates*, computes their relative frequency, and stores them in the final list of negations N. N is a list of negation objects,[4] where every object consists of a negation statement and its score.

$N = [(\neg st_1, sc_1), \ldots, (\neg st_r, sc_r)]$.

Across peer groups, it retains the maximum relative frequencies (hence, line 13), if a property or statement occurs across several. Before returning the top k results as output (line 19), it subtracts those already possessed by entity e (line 18).

Example 1. Consider the entity *e=Brad Pitt*. Table 1 shows a few examples of his peers and candidate negative statements. We instantiate the peer group choice to be based on structured information, in particular, shared occupations with the subject, as in Recoin [52]. In Wikidata, *Pitt* has 9 occupations, thus we would obtain 9 peer groups of entities sharing one of these with *Pitt*.

P = [actors, film directors, …, models], with $n = 9$.

For readability, let us consider statements derived from only one of these peer groups, *actor*. Let us assume 3 entities in that peer group.

P_{actor} = {Russel Crowe, Tom Hanks, Denzel Washington}

The list of negative candidates, *candidates*, are all the predicate and predicate–object pairs shown in the columns of the 3 actors. And in this particular example, N is just *ucandidates* with scores for only the *actor* group.

$$\begin{aligned} N = [&(\neg(\text{award; Oscar for Best Actor}), 1.0),\\ &(\neg\exists x(\text{instagram; } x), 0.67),\\ &(\neg(\text{citizen; New Zealand}), 0.33),\\ &(\neg\exists x(\text{convicted; } x), 0.33),\\ &(\neg\exists x(\text{child; } x), 1.0),\\ &(\neg(\text{occupation; screenwriter}), 1.0),\\ &(\neg(\text{citizen; U.S.}), 0.67)]. \end{aligned}$$

[4] Here, object is meant as a data type and not a KB-triple object.

Algorithm 1: Peer-based candidate retrieval algorithm.

```
1  Input  : knowledge base KB, entity e, peer collection function peer_groups, max. size of a peer group s, number of results k
   Output: k-most frequent negative statement candidates for e
2  P[]= peer_groups(e, s)          ▷ List of peer group(s); Group P_i at position i is one group (set) with at most s peers.
3  N[]= ∅                          ▷ Ranked list of negative statements about e.
4  for P_i ∈ P do
5      candidates = []             ▷ Positive statements (i.e., predicate and predicate–object pairs) of P_i members.
6      for pe ∈ P_i do
7          candidates+=collectP(pe)      ▷ Collecting predicates that hold for one peer (pe).
8          candidates+=collectPO(pe)     ▷ Collecting predicate–object pairs that hold for pe.
9      end
10     ucandidates = unique(candidates)  ▷ List of unique statements in candidates.
11     for st ∈ ucandidates do
12         sc = count(st,candidates) / s   ▷ sc computes how many peers share the statement st, normalized by s.
13         if getnegation(N, st).score < sc then
14             setscore(N, st, sc)
15         end
16     end
17 end
18 N-=inKB(e, N)                   ▷ Remove statements e already has.
19 return max(N, k)
```

Candidates that *hold* for *Pitt* are then dropped.

$$N = [(\neg(\text{award; Oscar for Best Actor}), 1.0),\\ (\neg\exists x(\text{instagram; } x), 0.67),\\ (\neg(\text{citizen; New Zealand}), 0.33),\\ (\neg\exists x(\text{convicted;} x), 0.33),\\ (\neg(\text{occupation; screenwriter}), 1.0)].$$

The top-k of the rest of candidates in N are finally returned. The top-3 negative statements, for this example, are ¬(award; Oscar for Best Actor), ¬(occupation; screenwriter), and ¬∃x(instagram; x).

The "if" statement at line 13 is only needed when multiple peer groups are considered for an entity. In the case where a negative statement is inferred from more than 1 group, only the version with the highest score is added to the final set. In the original (*full*) example, *Pitt* belongs to the group *actor* and the group *model*. The negation ¬(occupation; screenwriter) was inferred twice, once from each group, with a relative frequency of 0.9 from the *actor* group and 0.2 from the *model* group. We add the one with the higher score to the final set and disregard the other one. An alternative is to combine or compute the average of the scores across groups.

Note that without proper thresholding, the candidate set grows very quickly, for instance, if using only 30 peers, the candidate set for *Pitt* on Wikidata is already about 1500 statements.

Ranking Negative Statements. Given potentially large candidate sets, in a second step, ranking methods are needed. Our rationale in the design of the following four ranking metrics is to combine frequency signals with popularity and probabilistic likelihoods in a *learning-to-rank model*.

1. *Peer frequency (PEER):* The statement discovery procedure already provides a relative frequency, e.g., 0.9 of a given actor's peers are married, but only 0.1 are political activists. The former is an immediate candidate for ranking.
2. *Object popularity (POP):* When the discovered statement is of the form ¬(s; p; o), its relevance might be reflected by the popularity[5] of the Object. For example, ¬(Brad Pitt; award; Oscar for Best Actor) would get a higher score than ¬(Brad Pitt; award; London Film Critics' Circle Award), because of the high popularity of the *Academy Awards* over the *London Film Award*.
3. *Frequency of the Property (FRQ):* When the discovered statement has an empty Object ¬∃x(s; p; x), the frequency of the Property will reflect the authority of the statement. To compute the frequency of a Property, we refer to its frequency in the KB. For example, ¬∃x(Joel Slater; citizen; x) will get a higher score (4.1 m citizenships in Wikidata) than ¬∃x(Joel Slater; twitter; x) (294k twitter usernames).
4. *Pivoting likelihood (PIVO):* In addition to these frequency/view-based metrics, we propose to consider textual background information about e in order to better decide whether a negative statement is relevant. To this end, we build a set of statement pivoting classifier [55], i.e., classifiers that decide whether an entity has a certain statement (or property), each trained on the Wikipedia embeddings [56] of 100 entities that have a certain statement (or property), and 100 that do not.[6] To score a new statement (or property) candidate, we then use the pivoting score of the respective classifier, i.e., the likelihood of the classifier to assign the entity to the group of entities having that statement (or property).

The final score of a candidate statement is then computed as follows.

Definition 2 (*Ensemble Ranking Score*)**.**

$$Score = \begin{cases} \lambda_1\text{PEER} + \lambda_2\text{POP}(o) + \lambda_3\text{PIVO} \\ \quad \textit{if} \quad \neg(s; p; o) \textit{ is satisfied} \\ \lambda_1\text{PEER} + \lambda_4\text{FRQ}(p) + \lambda_3\text{PIVO} \\ \quad \textit{if} \quad \neg\exists x(s; p; x) \textit{ is satisfied} \end{cases}$$

Hereby λ_1, λ_2, λ_3, and λ_4 are parameters to be tuned on data withheld from training.

5. Order-oriented peer-based inference

In the previous section, we assume a binary peer relation as the basis of peer group computation. In other words, for

[5] Wikipedia page views.

[6] On withheld data, linear regression classifiers achieve 74% avg. accuracy on this task.

each entity, any other entity is either a peer, or is not. Yet in expressive knowledge bases, relatedness is typically graded and multifaceted, thus reducing this to a binary notion risks losing valuable information. We therefore investigate, in this section, how negative statements can be computed while using ordered peer set.

Orders on peers arise naturally when using real-valued similarity functions, such as Jaccard-similarity, or cosine distance of embedding vectors. An order also naturally arises when one uses temporal or spatial features for peering. Here are some examples:

1. *Spatial:* Considering the class *national capital*, the peers closest to *London* are *Brussels* (199 miles), *Paris* (213 miles), *Amsterdam* (223 miles), etc.
2. *Temporal:* The same holds for temporal orders on attributes, e.g., via his role as president, the entities most related to *Biden* are *Trump* (predecessor), *Obama* (pre-predecessor), *Bush* (pre-pre-predecessor), etc.

Formalization. Given a target entity e_0, a similarity function $sim(e_a, e_b) \rightarrow \mathcal{R}$, and a set of candidate peers $E = \{e_1, \ldots, e_n\}$, we can sort E by sim to derive an ordered list of sets $L = [S_1, \ldots, S_n]$, where each S_i is a subset of E that consists of highly related entities to e_0.

Example 2. Let us consider temporal recency of having won the *Oscars for Best Actor/Actress* as similarity function w.r.t. the target entity *Olivia Colman*. The ordered list of closest peer sets S is `[{Frances McDormand, Gary Oldman}, {Emma Stone, Casey Affleck}, {Brie Larson, Leonardo DiCaprio}, {Julianne Moore, Eddie Redmayne}.., {Janet Gaynor, Emil Jannings}]`.

Given an index of interest m ($m \leq n$), we have a prefix list $S_{[1,m]}$ of such an ordered peer set list L. For any negative statement candidate *stmt*, we can compute two ranking features:

1. *Prefix-volume (VOL)*: The prefix volume denotes the size of the prefix in terms of peer entities considered, i.e., $VOL = |S_1 \cup \cdots \cup S_m|$. Note that the volume should not be mixed with the length m of the prefix, which does not allow easy comparison, as sets may contain very different numbers of members.
2. *Peer frequency (PEER)*: As in Section 4, *PEER* denotes the fraction of entities in $S_1 \cup \cdots \cup S_m$ for which *stmt* holds, i.e., FRQ/VOL, where FRQ is the number of entities sharing the statement.

Note that these two ranking features change values with prefix length. In addition, we can also consider static features like *POP* and *PIVO*, as introduced before.

Consider the entity *e=Olivia Colman* from our example, with prefix length 3. For the statement `(citizen of; U.S.)`, FRQ is 5 and VOL is 6, i.e., unlike *Olivia Colman*, 5 out of the 6 winners of the previous 3 years are U.S. citizens. Now considering prefix length 2, for the statement `(occupation; director)`, FRQ is 1 and VOL is 4, i.e., unlike *Olivia Colman*, 1 out of the 4 winners of the previous 2 years are directors.

We can now proceed to the actual problem of this section.

Research Problem 2. Given an entity e and an ordered set of peers, compile a ranked list of useful negative statements.

Ranking. What makes a negative statement from an ordered peer set *informative*? It is easy to see that a statement is preferred over another, if it has both a higher peer frequency (*PEER*) and prefix volume (*VOL*). For example, the statement `¬(citizen of; U.S.)` above is preferable over `¬(occupation; director)`, due to it being both reported on a larger set of peers, and with higher

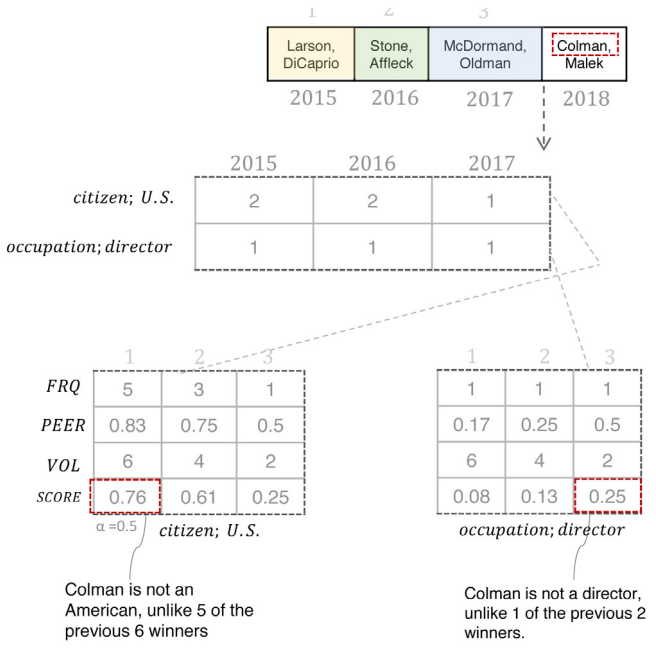

Fig. 1. Retrieving useful negative statements about *Olivia Colman*, using an ordered peer group.

relative frequency. Yet statements can be incomparable along these two metrics, and this problem even arises when comparing a statement with itself over different prefixes: Is it more helpful if 3 out of the previous 4 winners are *U.S.* citizens, or 7 out of the previous 10?

To resolve such situations, we propose to map the two features into a single one as follows:

$$score(stmt, L, m) = \lambda \cdot PEER + (1 - \lambda) \cdot log(FRQ) \quad (1)$$

where λ is again a parameter allowing to trade off the effects of the two variables. Note that we propose a logarithmic contribution of *FRQ* - this is based on the rationale that larger number of peers is preferable. For example, for the same *PEER* value 0.5, we can have a statement with 5 peers out of 10 and 1 peer out of 2.

Given the above example, the score for *Olivia Colman*'s negative statement `¬(citizen of; U.S.)` at prefix length 3 and $\alpha = 0.5$ is 0.76, with verbalization as "unlike 5 of the previous 6 winners". The same statement with prefix length 2 will receive a score of 0.61, with verbalization as "unlike 3 of the previous 4 winners". As for `¬(occupation; director)` at prefix length 3 and $\alpha = 0.5$ is 0.08, with verbalization as "unlike 1 of the previous 6 winners". The same statement with prefix length 2 will receive a score of 0.13, with verbalization as "unlike 1 of the previous 4 winners". This example is illustrated in Fig. 1.

Computation. Having defined how statements over ordered peer sets can be ranked, we now present an efficient algorithm, Algorithm 2, to compute the optimal prefix length per statement candidate, based on a single pass over the prefix. Given the entity *e=Olivia Colman*, ordered sets of her peers are collected in line 2.

$L =$ [winners of Oscar, winners of BAFTA,
..., recipients of CBE].

For readability, we proceed with one ordered peer group, namely the winners of Oscar for Best Actor/Actress. The group contains

Table 2
Negative statements about *Einstein*, before and after lifting.

Grounded negative statements	Conditional negative statements
¬(educated at; MIT)	¬∃*o*(educated at; *o*) (*o*; located in; U.S.)
¬(educated at; Stanford)	¬∃*o*(educated at; *o*) (*o*, instance of; private university)
¬(educated at; Harvard)	

ordered winners prior to *e*.

$$L_{\text{winners of Oscar}} = [\{\text{Frances McDormand, Gary Oldman}\},\ \{\text{Emma Stone, Casey Affleck}\},\ \{\text{Brie Larson, Leonardo DiCaprio}\},\ \{\text{Julianne Moore, Eddie Redmayne}\},\ \ldots,\ \{\text{Janet Gaynor, Emil Jannings}\}].$$

Similar to the previous algorithm, all statements of the peers are then retrieved from the KB (line 11 and 12). For every candidate statement *st*, the score(s) of the statement is computed with different prefix lengths (loop at line 28), starting with *pos* (position of *e* in the ordered set) and stopping at the start position 1. The maximum score is then returned with its corresponding values of *FRQ* and *VOL*, i.e., *max_frq* and *max_vol* (line 38). The returned candidate statement with its highest score (within one ordered group of peers L_i) is compared across many ordered groups of peers (i.e., other groups in *L*), to be either replaced or disregarded from the final list of negations *N*.

6. Conditional negative statements

In our negation inference methods, we generate two classes of negative statements, grounded negative statements, and universally negative statements. These two classes represent extreme cases: each grounded statement negates just a single assertion, while each universally negative statement negates all possible assertions for a property. Consequently, grounded statements may make it difficult to be concise, while universally negative statements do not apply whenever at least one positive statement exists for a property. A compromise between these extremes is to restrict the scope of universal negation. For example, it is cumbersome to list all major universities that *Einstein* did not study at, and it is not true that he did not study at any university. However, salient statements are that he *did not study at any U.S. university*, or that he *did not study at any private university*. We call these statements *conditional negative statements*, as they represent a conditional case of universal negation. In principle, the conditions used to constrain the object could take the form of arbitrary logical formulas. For proof of concept, we focus here on conditions that take the form of a single triple pattern.

Definition 3. A conditional negative statement takes the form ¬∃*o*: (s; p; *o*), (o; p'; o'). It is satisfied if there exists no *o* such that (s; p; *o*) and (o; p'; o') are in K^i.

In the following, we call the property p' the *aspect* of the conditional negative statement.

Example 3. Consider the statement that Einstein did not study at any *U.S.* university. It could be written as ¬∃*o* : `(Einstein; education; o)`, `(o; located in; U.S.)`. It is true, as *Einstein* only studied at *ETH Zurich, Luitpold-Gymnasium, Alte Kantonsschule Aarau*, and *University of Zurich*, located in *Switzerland* and *Germany*. Another possible conditional negative statement is ¬∃*o* : `(Einstein; education; o)`, `(o; type; private University)`, as none of these schools are private.

As before, the challenge is that there is a near-infinite set of true conditional negative statements, so a way to identify interesting ones is needed. For example, *Einstein* also did not study at any *Jamaican* university, nor did he study at any university that *Richard Feynman* studied at, etc. One way to proceed would be to traverse the space of possible conditional negative statements, and score them with another set of metrics. Yet compared to universally negative statements, the search space is considerably larger, as for every property, there is a large set of possible conditions via novel properties and constants (e.g., *"that was located in Armenia/Brazil/China/Denmark/.."*., *"that was attended by Abraham/Beethoven/Cleopatra/..."*). So instead, for efficiency, we propose to make use of previously generated grounded negative statements: In a nutshell, the idea is first to generate grounded negative statements, then in a second step, to *lift* subsets of these into more expressive conditional negative statements. A crucial step is to define this lifting operation, and what the search space for this operation is.

With the *Einstein* example, shown in Table 2, we could start from three relevant grounded negative statements that *Einstein* did not study at *MIT, Stanford*, and *Harvard*. One option is to lift them based on aspects they all share: their locations, their types, or their memberships. The values for these aspects are then automatically retrieved: they are all located in the *U.S.*, they are all private universities, they are all members of the *Digital Library Federation*, etc., however, not all of these may be interesting. So instead we propose to *pre-define* possible aspects for lifting, either using manual definition, or using methods for facet discovery, e.g., for faceted interfaces [57]. For manual definition, we assume the condition to be in the form of a single triple pattern. A few samples are shown in Table 3. For `educated at`, it would result in statements like "e was not educated in the *U.K.*" or "e was not educated at a public university"; for `award received`, like "e did not win any category of *Nobel Prize*"; and for `position held`, like "e did not hold any position in the *House of Representatives*".

Research Problem 3. Given a set of grounded negative statements about an entity *e*, compile a ranked list of useful conditional negative statements.

We propose an approach with Algorithm 3. Consider *e* = *Einstein*, and the set of possible aspects *ASP* for lifting containing only two aspects about `educated at`, for readability.

ASP = [(educated at: located in, instance of)].

The three grounded negative statements about *Einstein* with `educated at` property are:

NEG = [¬(educated at: MIT, Stanford, Harvard)].

The loop at line 3 considers every property (*neg*.*p*) in *NEG* (e.g., `educated at`), and collect its aspects at line 4. For this example, the list of aspects *asp* for this predicate consists of the location and the type of the educational institution.

asp = [located in, instance of].

At line 5, the loop visits every aspect *a* in *asp* and look for aspect values (i.e., the locations and types of Einstein's schools). *neg*.*o* are the objects that share the same predicate in the grounded negative statements list.

neg.*o* = [MIT, Stanford, Harvard].

Algorithm 2: Order-oriented peer-based candidate retrieval algorithm.

Input : knowledge base *KB*, entity *e*, ordered peer collection function *ordered_peers*, number of results *k*, hypeparameter of scoring function α
Output: top-*k* negative statement candidates for *e*

L[]= *ordered_peers*(*e*) ▷ List of ordered peer group(s); Group L_i at position *i* is one ordered group (list).
N[]= ∅ ▷ Ranked list of negative statements about *e*.
for $L_i \in \boldsymbol{L}$ **do**
  candidates = []
  pos=position(L_i, *e*) ▷ Position of *e* in the ordered set.
  for $pe \in L_i$ **do**
    if *pe* == *p* **then**
      continue
    end
    candidates+=*collectP*(*pe*)
    candidates+=*collectPO*(*pe*)
  end
  ucandidates = *unique*(*candidates*)
  for $st \in ucandidates$ **do**
    sc = *scoring*(st, L_i, e, pos, α) ▷ Dynamic scoring of every statement st with different prefix lengths.
    if *getnegation*(N, st).*score* $<$ sc **then**
      setscore(N, st, sc)
    end
  end
end
N-=*inKB*(*e*, *N*)
return *max*(*N*, *k*)

Function `scoring(`*st, S, e, pos,* α`)`**:**
  max_sc = - inf; max_frq = - inf; max_vol = - inf; ▷ Initializing the maximum score, frequency, and volume for statement *st*.
  frq = 0; vol=0; ▷ Initializing the frequency and volume of statement *st*.
  for *j = pos; j >= 1; j−−* **do**
    vol += countentities($S[j]$) ▷ Computing number of entities at position *j*.
    frq += countif(*st*, *candidates*, $S[j]$) ▷ Computing number of entities at position *j* that share *st*.
    sc = $\alpha * \frac{frq}{vol} + (1 - \alpha) * log(frq)$ ▷ Computing the score of *st* at position *j*.
    if *sc* > *max_sc* **then**
      max_sc = sc;
      max_frq = frq;
      max_vol = vol;
    end
  end
return max_sc, max_frq, max_vol

For every object *o*, aspect values are collected and their relative frequencies are stored. For readability, line 7 is only a high level version of this step. As mentioned before, the aspects are manually pre-defined and their values are automatically retrieved.

getaspvalues(Wikidata, located in, MIT) = [U.S].
getaspvalues(Wikidata, located in, Stanford) = [*U.S*].
getaspvalues(Wikidata, located in, Harvard) = [*U.S*].
getaspvalues(Wikidata, instance of, MIT) =
[institute of technology, private university].
getaspvalues(Wikidata, instance of, Stanford) =
[research university, private university].
getaspvalues(Wikidata, instance of, Harvard) =
[research university, private university].

Hence the aspect value for `educated at`, namely `(located in; U.S.)` receives a score of 3, and is added to the conditional negation list *cond_NEG*. After retrieving and scoring all the aspect values, the top-2 (with k =2) conditional negative statements are returned. In this example, the final results are *cond_NEG* = [(¬∃*o*`(Einstein; educated at;` *o*`)` (*o*`; located in; U.S.)`, 3), (¬∃*o*`(Einstein; educated at;` *o*`)` (*o*, `instance of; private university)`, 3)].

Table 3
A few samples of property aspects.

Property	Aspect(s)
educated at	located in; instance of;
award received	subclass of;
position held	part of;

7. Experimental evaluation

7.1. Peer-based inference

Setup. We instantiated the peer-based inference method with 30 peers, popularity based on Wikipedia page views, and peer groups based on entity occupations. The choice of this simple peering function was inspired by Recoin [52]. In order to further ensure relevant peering, we also only considered entities as candidates for peers, if their Wikipedia viewcount was at least a quarter of that of the subject entity. We randomly sampled 100 popular Wikidata people. For each of them, we collected 20 negative statement candidates: 10 with the highest *PEER* score, 10 being chosen at random from the rest of retrieved candidates. We then

Algorithm 3: Lifting grounded negative statements algorithm.

Input : knowledge base *KB*, entity *e*, aspects $ASP = [(x_1: y_1, y_2, ..), ..., (x_n: y_1, y_2, ..)]$, grounded negative statements about *e* $NEG = [\neg(p_1: o_1, o_2, ..), ..., \neg(p_m: o_1, o_2, ..)]$, number of results *k*
Output: *k*-most frequent conditional negative statements for *e*

```
cond_NEG= Ø                                          ▷ Ranked list of conditional negations about e.
for neg.p ∈ NEG do
    asp = getspects(neg.p, ASP)                      ▷ Retrieving aspects of predicate neg.p.
    for a ∈ asp do
        for o ∈ neg.o do
            cond_NEG += getaspvalues(KB, a, o)       ▷ Collecting aspect values about o.
        end
    end
end
cond_NEG-=inKB(e, cond_NEG)
return max(cond_NEG, k)
```

used crowdsourcing[7] to annotate each of these 2000 statements on whether it was interesting enough to be added to a biographic summary text (Yes/Maybe/No). Each task was given to 3 annotators. Interpreting the answers as numeric scores (1/0.5/0), we found a standard deviation of 0.29, and full agreement of the 3 annotators on 25% of the questions. Our final labels are the numeric averages among the 3 annotations.

Parameter Tuning. To learn optimal parameters for the ensemble ranking function (Definition 2), we trained a linear regression model using 5-fold cross validation on the 2k labels for usefulness. Four example rows are shown in Table 4. Note that the ranking metrics were normalized using a ranked transformation to obtain a uniform distribution for every feature.

The average obtained optimal parameter values were −0.03 for *PEER*, 0.09 for *FRQ(p)*, −0.04 for *POP(o)*, and 0.13 for *PIVO*, and a constant value of 0.3., with a 71% out-of-sample precision.

Ranking Metric. To compute the ranking quality of our method against a number of baselines, we used the Discounted Cumulative Gain (DCG) [58], which is a measure that takes into consideration the rank of relevant statements and can incorporate different relevance levels. DCG is defined as follows:

$$DCG(i) = \begin{cases} G(1) & if\ i = 1 \\ DCG(i-1) + \frac{G(i)}{log(i)} & otherwise \end{cases}$$

where i is the rank of the result within the result set, and $G(i)$ is the relevance level of the result. We set $G(i)$ to a value between 1 and 3, depending on the annotator's assessment. We then averaged, for each result (statement), the ratings given by all annotators and used it as the relevance level for the result. Dividing the obtained DCG by the DCG of the ideal ranking, we obtained the normalized DCG (nDCG), which accounts for the variance in performance among queries (entities).

Baselines. We used three *baselines*: As a naive baseline, we randomly ordered the 20 statements per entity. This baseline gives a lower bound on what any ranking model should exceed. We also used two competitive embedding-based baselines, TransE [54] and HolE [59]. For these two, we used pretrained models, from [60], on Wikidata (300k statements) containing prominent entities of different types, which we enriched with all the statements about the sampled entities. We plugged their prediction score for each candidate grounded negative statement.[8]

Results. Table 5 shows the average *nDCG* over the 100 entities for top-k negative statements for k equals 3, 5, 10, and 20. As one can see, our ensemble outperforms the best baseline by 6 to 16% in *nDCG*. The coverage column reflects the percentage of statements that this model was able to score. For example, for the *Popularity of Object*, *POP*(*o*) metric, a universally negative statement will not be scored. The same applies to TransE and HolE.

Ranking with the *Ensemble* and ranking using the *Frequency of Property* outperforms all other ranking metrics and the three baselines, with an improvement over the random baseline of 20% for k=3 and k=5. Examples of ranked top-3 negative statements for *Albert Einstein* are shown in Table 6. The random rank basically display any candidate negation if it holds for at least one peer. For instance, *Omar Sharif* is *Einstein*'s peer under the *non-fiction writer* group. This makes the negation "Tarek Sharif not a child of Einstein" possible, hence, the necessity for a ranking step. Moreover, *Omar Sharif* is also an actor, which brings other topics to the result set of *Einstein*, such as not winning *film awards*. This is where peer frequency makes a difference, i.e., most of *Einstein*'s peers are *not* actors. By relying on the property frequency for ranking, we can see that only universally absent statements get the highest scores. Even though it displays interesting negations (e.g., despite his status as famous researcher, *Einstein* truly never formally supervised any Ph.D. student), the top-k result set lacks grounded negative statements. Ensemble ranking, on the other hand, takes into consideration several features simultaneously, and covers both classes of negation. It returns interesting statements such as that *Einstein* notably refused to work on the *Manhattan* project, and was suspected of communist sympathies.

Correctness Evaluation. We used crowdsourcing to assess the correctness of results from the peer-based method. We collected 1k negative statements belonging to the three types, namely people, literature work, and organizations. Every statement was annotated 3 times as either correct, incorrect, or ambiguous. 63% of the statements were found to be correct, 31% were incorrect, and 6% were ambiguous. Most incorrect statements are due to KB completion issues. Interpreting the scores numerically (0/0.5/1), annotations showed a standard deviation of 0.23.

PCA (Partial Completeness Assumption) vs. CWA For a sample of 200 statements about people (10 each for 20 entities), half generated only relying on the CWA, half additionally filtered to satisfy the PCA (subject has at least one other object for that property [61]), we manually checked correctness. We observed 84% accuracy for PCA-based statements, and 57% for CWA-based statements. So the PCA yields significantly more correct negative statements, though losing the ability to predict universally negative statements.

Subject coverage. Our peer-based inference method offers a very high subject coverage and is able to discover negative statements about almost any existing entity in a given KB, whereas for

[7] https://www.mturk.com.

[8] Note that both models are not able to score statements about universal absence, a trait shared with the object popularity heuristic in our ensemble.

Table 4
Data samples for illustrating parameter tuning.

Statement	PEER	FRQ(p)	POP(o)	PIVO	Label
¬(Bruce Springsteen; award; Grammy Lifetime Achievement Award)	0.8	0.8	0.55	0.25	0.83
¬(Gordon Ramsay; lifestyle; mysticism)	0.3	0.8	0.8	0.65	0.33
¬∃x(Albert Einstein; doctoral student; x)	0.85	0.9	0.15	0.4	0.66
¬∃x(Celine Dion; educated at; x)	0.95	0.95	0.25	0.95	0.5

Table 5
Ranking metrics evaluation results for peer-based inference.

Ranking Model	Coverage(%)	$nDCG_3$	$nDCG_5$	$nDCG_{10}$	$nDCG_{20}$
Random	100	0.37	0.41	0.50	0.73
TransE [54]	31	0.43	0.47	0.55	0.76
HolE [59]	12	0.44	0.48	0.57	0.76
Property Frequency	11	**0.61**	**0.61**	**0.66**	**0.82**
Object Popularity	89	0.39	0.43	0.52	0.74
Pivoting Score	78	0.41	0.45	0.54	0.75
Peer Frequency	100	**0.54**	**0.57**	**0.63**	**0.80**
Ensemble	100	**0.60**	**0.61**	**0.67**	**0.82**

Table 6
Top-3 results for *Albert Einstein* using 3 ranking metrics.

Random rank	Property frequency	Ensemble
¬∃x(instagram; x)	¬∃x(doctoral student; x)	¬(occup.; astrophysicist)
¬(child; Tarek Sharif)	¬∃x(candidacy in election; x)	¬(party; Communist Party USA)
¬(award; BAFTA)	¬∃x(noble title; x)	¬∃x(doctoral student; x)

pre-trained embedding-based baselines, many subjects are out-of-vocabulary, or come with too little information to predict statements.

7.2. Inference with ordered peers

In the following, we used temporal order on specific roles, or on specific attribute values, to compute ordered peer sets. In particular, we used two common forms of temporal information in Wikidata to compute such peer groups:

- **Time-based Qualifiers (TQ)**: Temporal qualifiers are time signals associated with statements about entities. In Wikidata, some of those qualifiers are *point in time* (P585), *start time* (P580), and *end time* (P582). A few samples are shown in Table 7.
- **Time-based Properties (TP)**: Temporal properties are properties like *follows* (P155) and *followed by* (P156) indicating a chain of entities, ordered from oldest to newest, or from newest to oldest. For instance, `[The Cossacks; followed by; War and Peace; followed by; Anna Karenina; ..]`[9]

We created TQ groups from aggregating information about people sharing the same statements. For example, `position held; President of the U.S.` is one TQ group, where members will have a *start time* for this position, as well as an *end time*. In case of absence of an *end time*, this implies that the statement holds to this day (`Donald Trump`'s statement in Table 7). In other words, we aggregated entities sharing the same predicate–object pair, which will be treated as the peer group's title, and ranked them in ascending order of time qualifiers. For the *point in time* qualifier, we simply ranked the dates from oldest to newest, and for the *start/end date*, we ranked the end date from oldest to newest. If the *end date* is missing, the entity will be moved to the newest slot.

We collected a total of 19.6k TQ groups (13.6k using the *start/end date* qualifier and 6k using the *point in time* qualifier). Based on a manual analysis of a random sample of 100 groups of different sizes, we only considered time series with at least 10 entities.[10]

We created TP groups by first collecting all entities reachable by one of the transitive properties, *follows* (P155) and *followed by* (P156). Considering each of the collected entities as a source entity, we computed the longest possible path of entities with only transitive properties. This path consists in an ordered set of peers. To avoid the problem of double-branching (one entity followed by two entities), we considered the two directions separately. Again, one path will be chosen at the end; the one with maximum length. The total number of TP groups is 19.7k groups. We limited the size of the groups to at least 10 and at most 150.[11]

Setup and Baseline. We chose 100 entities, that belongs to at least one ordered set of peers, from Wikidata: 50 people and 50 literature works. We collected top-5 negative statements for each of those entities (for people, we consider TQ groups, and for literature works, TP groups). We made this choice because of the lack of entities of type person with transitive properties. In case an entity belongs to several groups, we merged all the results it is receiving from different groups, ranked them, and retrieved the top-5 statements. Similarly, as a baseline, using the peer-based inference method of Section 4, instantiated with cosine similarity on Wikipedia embeddings [56] as similarity function, we collected the top-5 negative statements for the same entities. We ended up with 1k statements, 500 inferred by each model.

Correctness Evaluation. We randomly retrieved 400 negative statements from the 1k statements collected above, 200 from each model (100 about people, and 100 about literature works). We then assessed the correctness of each method using crowdsourcing. We showed each statement to 3 annotators, asking them to choose whether this statement is correct, incorrect, or ambiguous. Results are shown in Table 8. Our order-oriented inference method clearly infers less incorrect statements by 9 percentage points for people, and 5 for literature works. It also

[9] Novels of by Leo Tolstoy.

[10] This variable can be easily adjusted depending on the preference of the developers and/or the purpose of the application.

[11] We did not truncate the groups, we simply disregarded any group smaller or larger than the thresholds.

Table 7
Samples of temporal information in Wikidata.

Statement	Time-based qualifier(s)
(Barack Obama; position held; U.S. senator)	*start time*: 3 January 2005; *end time*: 16 November 2008
(Maya Angelou; award received; Presidential Medal of Freedom)	*point in time*: 2010
(Donald Trump; spouse; Melania Trump)	*start time*: 22 January 2005

Table 8
Correctness of order-oriented and peer-based methods.

	People	Literature work
	Peer-based inference	
	%	%
Correct	81	88
Incorrect	18	12
Ambiguous	1	0
	Order-oriented inference	
	%	%
Correct	**91**	**91**
Incorrect	9	7
Ambiguous	0	2

produces more correct statements for people by 10 percentage points, and literature work by 3. The percentage of queries with full agreement in this task is 37%. Also, annotations show a standard deviation of 0.17.

Subject Coverage. To assess the subject coverage of the order-oriented method, we randomly sampled 1k entities from each dataset, and tested whether it is a member of at least one ordered set, thus the ability to infer useful negative statements about it. For TQ groups, we randomly sampled 1k people, which results in a coverage of 54%. And for TP groups, we randomly sampled 1k literature works, and also received a coverage of 54%. Although the order-oriented method produces better negative statements on both notions of correctness and usefulness (as we will see next), it does not outperform the baseline on subject coverage. However, using a different function to order peers might affect this drastically (e.g., using real-valued similarity functions like cosine distance of embedding vectors).

Usefulness. To assess the quality of our inferred statements from the order-oriented inference method against the baseline (the peer-based inference method), we presented to the annotators two sets of top-5 negative statements about a given entity, and asked them to choose the more interesting set. The total number of opinions collected, given 100 entities, 3 annotations each, is 300. To avoid biases, we repeatedly switched the position of the sets. Results are shown in Table 10. Overall results show that our method is preferred by 10% of the entities for both domains. The standard deviation of this task is 0.24 and the percentage of queries with full agreement is 18%. We observe two advantages of the ordered set of peers over the previous method: i) it gives better interpretations of what a peer is, by automatically producing labels for peer groups (e.g., Presidents of the U.S., Winners of the Best Actor Academy Award); and ii) it maximizes the *peerness* within a group. For instance, with Wikipedia embedding [56], closest peers to *Donald Trump* are *Hillary Clinton* and *Donald Trump Jr.*. While the peerness with the input entity is obvious, there is not much similarity between the peers themselves, hence, very sparse candidate negations. However, with the order-oriented peering, *Trump*'s peers include *Barack Obama* and *George W. Bush*, who are also peers of each other.

Evaluation of Verbalizations. One main contribution that our order-oriented inference method offers are *verbalizations* produced with every inferred negative statement. In other words, it can, unlike the peer-based inference method, produce more concrete explanations of the usefulness of the inferred negations. For example, the inferred negative statement ¬(`Abraham Lincoln; cause of death; natural causes`) was inferred by both of our methods. However, each method offers a different verbalization. For the peer-based method, the verbalization is "unlike 10 of 30 similar people", and for the order-oriented method is "unlike 12 of the previous 12 presidents of the U.S.". To assess the quality of the verbalizations more formally, we conducted a crowdsourcing task with 100 useful negations that were inferred by both methods from our previous experiment. For every negative statement, the annotator was shown two different verbalizations on "why is this negative statement noteworthy". We asked the annotator to choose the better verbalization, she can choose Verbalization1, Verbalization2, or Either/Neither. Results show that verbalizations produced by our order-oriented inference method were chosen 76% of the time, by the peer-based inference method 23% of the time, and the either or neither option only 1% of the time. The standard deviation is 0.23, and the percentage of queries with full agreement is 20%. Table 9 shows a number of examples, using different grouping functions for the peer-based method.

7.3. Conditional negative statements evaluation

We evaluated our lifting technique to retrieve useful conditional negative statements, based on three criteria: (i) compression, (ii) correctness, and (iii) usefulness. We collected the top-200 negative statements about 100 entities (people, organizations, and art work), and then applied lifting on them.

Compression. On average, 200 statements are reduced to 33, which means that lifting compresses the result set by a factor of 6.

Correctness. We asked the crowd to assess the correctness of 100 conditional negative statements (3 annotations per statement), chosen randomly. To make it easier for annotators who are unfamiliar with RDF triples,[12] we manually converted them into natural language statements, for example "*Bing Crosby did not play any keyboard instruments*". Results show that 57% were correct, 23% incorrect, and 20% were uncertain. The standard deviation of this task is 0.24 and the percentage of queries with full agreement is 18%.

Usefulness. For every entity, we showed 3 annotators 2 sets of top-3 negative statements: a grounded and universally negative statements set and a conditional negative statement set, and asked them to choose the one with more interesting information. Results are shown in Table 11. The conditional statements were chosen 45 percentage points more than the grounded and universally negative statements. The standard deviation of this task is 0.22 and the percentage of queries with full agreement is 21%. The significant out-performance of the conditional class over the other two classes is that it encapsulates them. Without losing the information from the original result set, lifting summarizes negations in meaningful manner, at the same time, allowing more diverse statements to be displayed in a top-k set. An example is shown in Table 12, with entity e =*Leonardo Dicaprio*, and its top-3 results. Even though he is one of the most accomplished actors in the world, unlike many of his peers, he never attempted directing any kind of creative work (films, plays, television shows, etc.).

[12] Especially because of the triple-pattern condition.

Table 9
Negative statements and their verbalizations using peer-based and order-oriented methods.

Statement	Order-oriented *Unlike..*	Peer-based *Unlike..*	Peering
¬(Emmanuel Macron; member; National Assembly)	29 of 36 members of La République En Marche party	70 of 100 similar people	WP embed. [56]
¬(Tim Berners-Lee; citizenship; U.S.)	101 of previous 115 winners of the MacArthur Fellowship	53 of 100 sim. comp. scientists	Structured facets
¬(Michael Jordan; occupation; basketball coach)	27 of prev. 49 winners of the NBA All-Defensive Team	31 of 100 sim. people	WP embed. [56]
¬(Theresa May; position; Opposition Leader)	11 of prev. 14 Leaders of the Conservative Party	10 of 100 sim. people	WP embed. [56]
¬(Cristiano Ronaldo; citizenship; Brazil)	4 of prev. 7 winners of the Ballon d'Or	20 of 100 sim. football players	Structured facets

Table 10
Usefulness of order-oriented and peer-based methods.

	People	Literature work
	%	%
Peer-based inference	42	44
Order-oriented inference	**52**	**54**
Both	6	2

Table 11
Usefulness of conditional negative statements.

Preferred	(%)
Conditional negative statements	**70**
Grounded and universally negative statements	25
Either or neither	5

8. Extrinsic evaluation

We highlight the relevance of negative statements for:

- Entity summarization on Wikidata.
- Decision support with hotel data from Booking.com.
- Question answering on various structured search engines.

8.1. Entity summarization

In this experiment we analyze whether mixed positive-negative statement set can compete with standard positive-only statement sets in the task of entity summarization. In particular, we want to show that the addition of negative statements will *increase the descriptive power* of structured summaries.

We collected 100 Wikidata entities from 3 diverse types: 40 people, 30 organizations (including publishers, financial institutions, academic institutions, cultural centers, businesses, and more), and 30 literary works (including creative work like poems, songs, novels, religious texts, theses, book reviews, and more). On top of the negative statements that we inferred, we collected relevant positive statements about those entities.[13] We then computed for each entity *e* a sample of 10 positive-only statements, and a mixed set of 7 positive and 3 *correct*[14] negative statements, produced by the peer-based method. We relied on peering using Wikipedia embeddings [56]. Annotators were then asked to decide which set contains more new or unexpected information about *e*. More particularly, for every entity, we asked workers to assess the sets (flipping the position of our set to avoid biases), leading to a total number of 100 tasks for 100 entities. We collected 3 opinions per task. Overall results show that mixed sets with negative information were preferred for 72% of the entities, sets with positive-only statements were preferred for 17% of the entities, and the option "both or neither" was chosen for 11% of the entities. Table 14 shows results per each considered type. The standard deviation is 0.24, and the percentage of queries with full agreement is 22%. Table 13 shows three diverse examples. The first one is *Daily Mirror*. One particular noteworthy negative statement in this case is that the newspaper is not owned by the "*News U.K.*" publisher which owns a number of other *British* newspapers like *The Times, The Sunday Times, and The Sun*. The second entity is *Peter the Great* who died in *Saint Petersburg* and not *Moscow*, and who did not receive the *Order of St Alexander Nevsky* which was first established by his wife, a few months after his death. And the third entity is *Twist and Shout*. Although it is a known song by *The Beatles*, they were *not* its composers, writers, nor original performers.

In this experiment, we showed that adding negative statements to a set of positive statements increases its quality. For that, we chose a split of 7 positive and 3 negative statements for top-10 results. One may wonder whether that is actually the best proportion. This motivates another analysis, *finding out the portion of negative statements to be added to a positive top-k set of statements that maximizes the relevance gain* (i.e., nDCG). We used the annotators' assessment of relevancy of individual positive and negative statements. We then compiled them as sets of top-k results with different k values and different portions of negative statements. The decision of adding a certain negative statement should respect the constraint of not decreasing the relevance gain (i.e., nDCG) of the currently chosen top-k results. We calculated the ideal ratio of positive to negative statements for k results. The ideal portion of negative statements within top-k statements about entity *e* was obtained for k=3, 5, 10, and 20. For a set of top-3 or top-5 statements, 1 negative statement is ideal, for 10 statements, 2 are ideal, and for 20, 5 are ideal.

8.2. Decision support

Negative statements are highly important also in specific domains. In online shopping, characteristics not possessed by a product, such as the *IPhone 7* not having a headphone jack, are a frequent topic highly relevant for decision making. The same applies to the hospitality domain: the absence of features such as free WiFi or gym rooms are important criteria for hotel bookers, although portals like Booking.com currently only show (sometimes overwhelming) positive feature sets.

To illustrate this, based on a comparison of 1.8k hotels in India, as per their listing on Booking.com, using the peer-based method, we inferred useful negative features. For peering, we considered all other hotels in India, and for ranking, we computed peer frequencies (*PEER*). We then used crowdsourcing over the results of 100 hotels. We asked annotators to check two sets of features about a given hotel, one set containing 5 random positive-only features, and one set containing a mix of 3 positive and 2 negative features. Their task was to choose which set of features will help them more in deciding whether to stay in this hotel or not. They

[13] We defined a number of common/useful properties to each of type, e.g., for people, "position held"is a relevant property for positive statements.

[14] We manually checked the correctness of these negative statements.

Table 12
Top-3 negative statements about *Leonardo Dicaprio*, before and after lifting.

Negative statements	Conditional negative statements
¬(occupation; film director)	¬∃*o* (occupation; *o*) (*o*; subclass of; director)
¬(occupation; theater director)	¬∃*x*(spouse; x)
¬(occupation; television director)	¬∃*x*(child; x)

Table 13
Results for the entities *Daily Mirror*, *Peter the Great*, and *Twist and Shout*.

Daily Mirror	
Pos-only	Pos-and-neg
(owned by; Reach plc)	***¬(newspaper format; broadsheet)***
(newspaper format; tabloid)	(newspaper format; tabloid)
(country; United Kingdom)	***¬(country; U.S.)***
(language of work or name; English)	(language of work or name; English)
(instance of; newspaper)	***¬(owned by; News U.K.)***
...	...
Peter the Great	
Pos-only	Pos-and-neg
(military rank; general officer)	(military rank; general officer)
(owner of; Kadriorg Palace)	(owner of; Kadriorg Palace)
(award; Order of the Elephant)	***¬(place of death; Moscow)***
(award; Order of St. Andrew)	(award; Order of St. Andrew)
(father; Alexis of Russia)	***¬(award; Knight of the Order of St. Alexander Nevsky)***
...	...
Twist And Shout	
Pos-only	Pos-and-neg
(composer; Phil Medley)	***¬(composer; Paul McCartney)***
(performer; The Beatles)	(performer; The Beatles)
(producer; George Martin)	***¬(composer; John Lennon)***
(instance of; musical composition)	(instance of; musical composition)
(lyrics by; Phil Medley)	***¬(lyrics by; Paul McCartney)***
...	...

Table 14
Positive-only vs. positive and negative statements.

Preferred Choice	Person (%)	Organization (%)	Literary work (%)
Pos-and-neg	**71**	**77**	**66**
Pos-only	22	10	17
Both or neither	7	13	17

Table 15
Usefulness of hotel features.

Preferred Choice	(%)
Pos-and-neg	**54**
Pos-only	38
Either or neither	8

can choose one of the sets, or both. For every hotel, we request 3 annotators.

Table 15 shows that sets with negative features were chosen 16 percentage points more than the positive-only sets. The standard deviation of this task is 0.22 and the percentage of queries with full agreement is 28%. Table 16 shows three hotels with useful negative features. Although the *Hotel Asia The Dawn* lists 64 positive features, negative information such as that it does not offer air conditioning and free Wifi may give important clues for decision making.

Moreover, we collected 20 pairs of hotels from the same dataset, and showed every pair's Booking.com pages to 3 annotators. We asked them to choose the better hotel for them. Then we showed them negative features about the pair, and asked them whether this new information would change their mind on their initial decision. A screenshot of the task is shown in Fig. 2. 42% changed their pick after negative features were revealed. The standard deviation on this task is 0.15. The full agreement of the 3 annotators on *changing the hotel after negative features were revealed* is 35%. The full agreement of annotators *choosing the same hotel at the end of the task* is 30%. The latter agreement measure disregard whether they have changed their decision or stayed with their initial choice.

8.3. Question answering

In this experiment, we compared the results to negative questions over a diverse set of sources. We manually compiled 20 questions that involve negation, such as *"Actors without Oscars"*[15]. We compared them over four highly diverse sources: Google Web Search (increasingly returning structured answers from the Google knowledge graph [4]), WDAqua [62] (an academic state-of-the-art KBQA system), the Wikidata SPARQL endpoint[16] (direct access to structured data), and our peer-based method. For Google Web Search and WDAqua, we submitted the queries in their textual form, and considered answers from Google if they come as structured knowledge panels. For Wikidata and peer-based inference, we transform the queries into SPARQL queries,[17] which we either fully executed over the Wikidata endpoint, or executed the positive part over the Wikidata endpoint, while evaluating the negative part over a dataset produced by our peer-based inference method. Note that all queries were safe, since they were designed to always asks for a class of entities (e.g., entities of occupation actor) that do not satisfy a certain

[15] Sample textual queries: "actors with no Oscars", "actors with no spouses", "film actors who are not film directors", "football players with no Ballon d'Or", "politicians who are not lawyers".

[16] https://query.wikidata.org/.

[17] sample SPARQL queries: https://w.wiki/A6r, https://w.wiki/9yk, https://w.wiki/9yn, https://w.wiki/9yp, https://w.wiki/9yq.

Table 16
Negative statements for hotels in India.

Hotel	Number of positive features	Top-3 negative features
The Sultan Resort	106	¬ Parking; ¬ Fan; ¬ Newspapers
Vista Rooms at Mount Road	28	¬ Room service; ¬ Food & Drink; ¬ 24-hour front desk
Hotel Asia The Dawn	64	¬ Air conditioning; ¬ Free Wifi; ¬ Free private parking

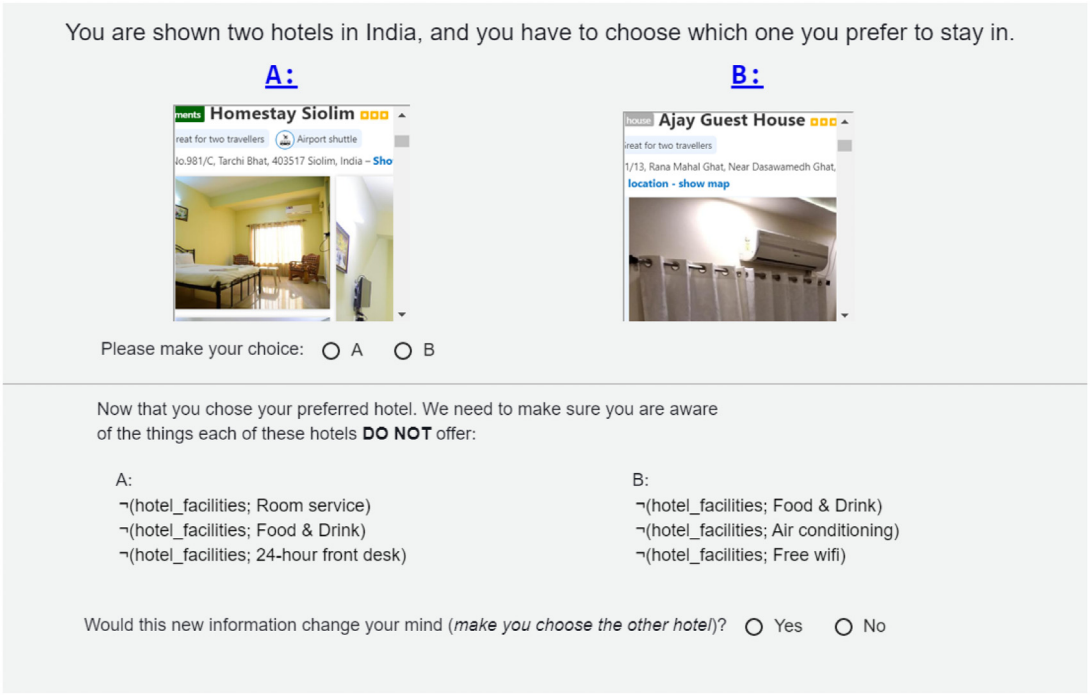

Fig. 2. Extrinsic use-case: decision support on hotel data.

property (e.g., having won the Oscar), which was captured via SPARQL MINUS with a shared variable. For each method, we then self-evaluated the number of results, the correctness and relevance of the (top-5) results. All methods were able to return highly correct statements, yet Google Web Search and WDAqua return no results for 18 and 16 of the queries, respectively.

We continued the assessment over a sample of 5 queries. Wikidata SPARQL returned by far the highest number of results, 250k on average, yet did not perform ranking, thus returned results that are hardly relevant (e.g., a local Latvian actor to the Oscar question). The peer-based inference outperforms it by far in terms of relevance (72% vs. 44% for Wikidata SPARQL). We point out that although Wikidata SPARQL results appear highly correct, this has no formal foundation, due to the absence of a stance of OWA KBs towards negative knowledge. For example, most actors or people did *not* win Oscars, which makes 99.99% of the entities returned by Wikidata's SPARQL query correct, even under the OWA.

9. Resources

Negative Statement Datasets for Wikidata. We publish the first datasets that contain dedicated *useful* negative statements about entities in Wikidata: (i) Peer-based and order-oriented inference data: 14 m negative statements about popular 600k entities from various types, (ii) release the mturk-annotated on the correctness of 1k negative statements of Section 7.1, and (iii) 40k ordered set of peers introduced in Section 7.2.

Open-source Code. We make our peer-based inference method available for users to try it on their own datasets.[18]

Demo. A web-based platform, Wikinegata [63] for browsing useful negations about Wikidata entities, is available at: https://d5demos.mpi-inf.mpg.de/negation/.

A screenshot is shown in Fig. 3.

All experimental material related to this paper can be found on a dedicated webpage.[19]

10. Discussion

10.1. Quality considerations

The CWA on the Semantic Web. Negation has traditionally been avoided on the Semantic Web, as it challenges the vision that anyone can state anything, without risking logical conflicts. In the present work, we showed that enriching KBs with useful negative statements is beneficial in use cases such as entity summarization and consumer decision making. In order to compile a set of likely correct negative statements about an entity, we assumed the CWA in parts of the KBs, namely within peer

18 https://github.com/HibaArnaout/usefulnegations.

19 https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/knowledge-base-recall/interesting-negations-in-kbs.

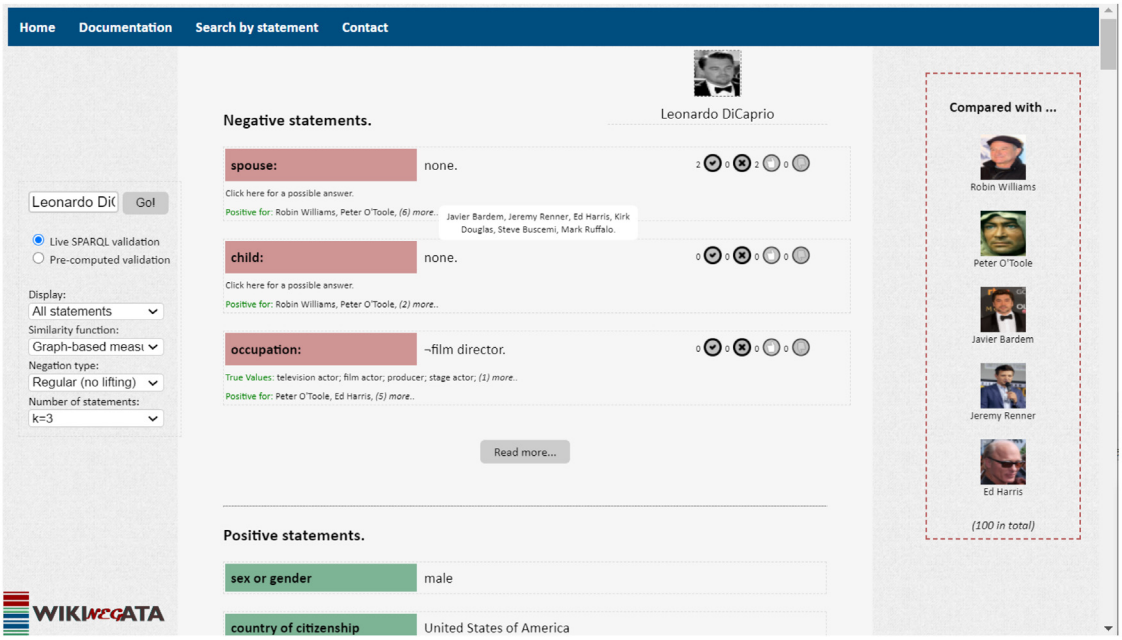

Fig. 3. Interface for Wikinegata — useful negative statements about *Leonardo DiCaprio*.

groups, and in the case of grounded negative statements, with the additional requirement that there is at least one other positive statement for the same entity-property pair. Although this approach outperforms other techniques, like embedding-based KB completion, inferences may still be incorrect. While correctness can be tuned to some extend by sacrificing recall (e.g., requiring very high thresholds on *PEER* and *PIVO*), errors are still possible. It is therefore advised to show candidate statements from automatic inference to KB curators for final assessment [52].

Real-world Changes and KB Maintenance. Due to real-world changes or new information added to the KB, some of the negative statements already inferred might become incorrect. For instance, *DiCaprio* has won his first *Oscar* in 2016. After the year 2016, the negative statement ¬(DiCaprio; award; Oscar) is no longer correct. Negative statements should therefore be timestamped, and ideally, additions of positive statements should automatically trigger updates of validity end-point timestamps.

Class Hierarchies. Some incorrect negations can be detected by help of subsumption checks (rdfs:subClassOf). For example, the presented method might incorrectly infer the statement ¬(Douglas Adams; occupation; author), which contradicts the two positive assertions that Douglas Adams is a writer, and writer is a subclass of author. One could detect such contradictions by use of a generic ontology reasoner like Protégé, or implement custom checks. For our specific use case of negative inference at scale, we found that checks focused on one or two hops in the class hierarchy capture a significant proportion of these errors. For KBs at the scale of Wikidata, one could precompute prominent subsumptions, and built these checks into the methodology (e.g., triggering a check for presence of "occupation-writer" whenever "occupation-not author" is inferred).

Subsumption similarly also affects properties: the relation *CEO* (between a company and a person) is a subproperty of *employee*, and as such, subject-object-pairs present for the former should not appear as negations for the latter.

Modeling and Constraint Enforcement. Some inferred negations are incorrect due to modeling issues, resulting in inconsistencies. An example is *Dijkstra* and the negative statement that his field of work is *not Computer Science*, and *not Information Technology*, while he has the positive value *Informatics*, which is arguably near-synonymous, yet in the Wikidata taxonomy, the two represent independent concepts, two hops apart. Some other incorrect negative statements could be due to a lack of constraints. For instance, for most businesses, the *headquarters location* property is completed using *cities*, but for *Siemens* in Wikidata, the *building* is added instead (*Palais Ludwig Ferdinand*), making our inferred statement ¬(Siemens; headquarters location; Munich) incorrect. Although Wikidata encourages editors to use cities for the *headquarters location* property and advise them to use another property for specific buildings, it has not been automatically enforced yet.[20]

10.2. Discovering relevant lifting aspects

For inferring conditional negative statements, the lifting aspects we used in this paper have been manually defined (see Table 3). For instance, if the grounded negative statements to be lifted describe educational institutions, then the aspects that make sense are the location of the institution (*U.S., Germany, Japan, etc.*) and its type (*public, private, research, etc.*). This does not scale well when the KB contains thousands of properties with thousands of possible aspects. Automatically discovering these aspects would improve the quality of conditional negative statements. A good start is the work in [57]. An aspect is described as an important characteristic of an entity. For example, for a book, the number of pages is not an important aspect, but genre is. This work introduces aspect ranking metrics such as object cardinality: a *good* predicate (e.g., genre) has a finite list of values to choose from (e.g., comedy, thriller, romance). Unsuitable predicates using this metric would be the predicate *number of pages* or *publication date*. In addition, the AMIE system [61,64] mines rules on millions of triples, and is specifically tailored to support open-world KBs. It can discover, for example, that *musicians that are influenced by each other often play the same instrument.* The *instrument* can be directly used as an aspect for lifting grounded negative statements (with predicate *influenced by*) about a *musician-entity*. In particular, *musician x* (a pianist), is

[20] https://www.wikidata.org/wiki/Property:P159.

Table 17
Negations across classes of Wikidata entities.

Class	Number of entities	3 most frequent negated properties	Sample entities
Book	8k	author, genre, publisher	Fahrenheit 451, Little Birds
Person	500k	spouse, child, occupation	Elon Musk, Oprah Winfrey
Country	199	diplomatic relation, member of, language used	Germany, China
Primary school	14k	instance of, heritage designation, country	Deutsche Schule Helsinki, Saint Joseph school
Film	26k	cast member, genre, screenwriter	Taxi Driver, Inception
Building	28k	architect, instance of, heritage designation	NY Times Building, White House
Organizations	22k	headquarters location, instance of, country	World Trade Organization, BBC
Musical group	8k	instance of, record label, genre	Coldplay, Jonas Brothers
Business	20k	parent organization, headquarters location, industry	Nokia, Facebook
Scientific journal	5k	main subject, editor, publisher	Journal of Web Semantics, Nature
Literary work	24k	author, composer, lyrics by	Diary of Anne Frank, Don Quixote

not influenced by anyone who plays the guitar, or more surprisingly not influenced by anyone who plays the piano (if that is the case). We consider this to be a promising research direction. It is worth exploring and improving the ideas in Section 6 further.

10.3. Entity prominence and class specificity

Negations in the Long Tail. Our method builds on the assumption that peer entities are available, for which we have sufficient data. For long-tail entities, both assumptions may be challenged. For entities with extremely little positive information (e.g., https://www.wikidata.org/wiki/Q97355589, for which only first name, last name, and gender are known), it is not possible to identify relevant peers, and hence, our method is not applicable. Low amounts of positive information on peers, in contrast, can be better compensated. Since our method is mainly concerned with finding the most interesting candidates for negation, absolute frequencies are not important, as long as it is possible to find a reasonable difference in frequencies among peers (i.e., not every positive statement appearing only once). If there is interest to put emphasis on specific facts, one could also adjust the ranking algorithms, e.g., giving "citizenship" negations a boost in the ranking.

Negations for Different Classes. In practice, we have applied our method on 11 diverse classes of entities: people, literary works, organizations, businesses, scientific journals, countries, buildings, musical groups, primary schools, books, and films. We have observed that within each class, interesting negations often cover the same properties. For instance, for people, interesting negative knowledge is mostly about awards, occupations, education, and family. We show statistics on frequent properties for every class of entities in Table 17. We do *not* filter nor assign weights for certain properties per class. The relative frequency metric takes care of prioritizing which property's negation *makes sense* in every class. For *people*, the reported properties are fairly general and not tied to specific subsets of this very large class. For instance, for sports figures, *member of sports team* is the most frequent property, and for politicians, *position held* is the dominating property.

We notice that negations for small classes, such as buildings and literary works, have a higher correctness ratio than larger classes, such as people. Entities of type *person* have 3 times more possible properties to fill than entities of type *book*. Given a book (e.g., *Orientalism*), a handful of properties and property–object pairs could be added and the information about the entity is considered near-complete (e.g., *main subject*, *author*, *genres*, *publisher*, and *language*). In contrast, for a person (e.g., *Joe Rogan*), the entity requires a greater effort and/or larger information sources to be considered complete (e.g., *occupation*, *education*, *residence*, *birth place*, *citizenship*, *sport*, *religion*, and many more). On the other hand, larger classes offer richer and more diverse possibilities for *interesting* negations. A result set for a person often covers a wider range of topics, such as personal information, professional achievements, relations with other people. A result set for a book is less diverse, often negating the same property repeatedly with different objects.

11. Conclusion & future directions

This article has made the first comprehensive case for explicitly materializing useful negative statements in KBs. We have introduced a statistical inference approach on retrieving and ranking candidate negative statements, based on expectations set by highly related peers. We have also released several resources to encourage further research.

In future work we would like to explore a number of research directions:

1. Missing vs negative statements: How to maximize tradeability between fewer highly correct statements, and larger sets of interesting negation candidates.
2. Mining complex negations: Our focus was on simple – grounded and universal – negation, with a hint at more complex conditional statements, but it is open to extend that to automatically finding aspects, further joins *"did not study at a university which was graduating any Nobel prize winner"*, negation on sets of entities instead of entity-centric *"no African country has hosted any Olympic games"*, etc.
3. Exploring textual information extraction for implicit negations, like *"Theresa May is an only child."* can be expressed using the KB statement $\neg\exists x$(sibling; x), and *"George Washington had no formal education."* as $\neg\exists x$(educated at; x).
4. Exploiting the ontology that comes with the KB to improve the correctness of inferred negations by making use of constraints like class and property subsumption.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

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

References

[1] D. Vrandečić, M. Krötzsch, Wikidata: A free collaborative knowledge base, CACM (2014).
[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, et al., DBpedia: A nucleus for a web of open data, in: ISWC, 2007.

[3] F. Suchanek, G. Kasneci, G. Weikum, Yago: A core of semantic knowledge, in: WWW, 2007.
[4] A. Singhal, Introducing the knowledge graph: things, not strings, 2012, https://www.blog.google/products/search/introducing-knowledge-graph-things-not.
[5] G. Flouris, Z. Huang, J. Pan, D. Plexousakis, H. Wache, Inconsistencies, negations and changes in ontologies, in: AAAI, 2006.
[6] A. Fader, Z. L., E. O., Open question answering over curated and extracted knowledge bases, in: KDD, 2014.
[7] Y. Yang, W. Yih, C. Meek, WikiQA: A challenge dataset for open-domain question answering, in: EMNLP, 2015.
[8] H. Arnaout, S. Razniewski, G. Weikum, Enriching knowledge bases with interesting negative statements, in: AKBC, 2020.
[9] T. Tanon, F. Suchanek, Querying the edit history of wikidata, in: ESWC, 2019.
[10] S. Malyshev, M. Krötzsch, L. González, L. Gonsior, A. Bielefeldt, Getting the most out of Wikidata: Semantic technology usage in Wikipedia's knowledge graph, in: ISWC, 2018.
[11] S. Ghosh, S. Razniewski, G. Weikum, Uncovering hidden semantics of set information in knowledge bases, JWS (2020).
[12] P. Ernst, A. Siu, G. Weikum, Knowlife: A versatile approach for constructing a large knowledge graph for biomedical sciences, BMC Bioinformatics (2015).
[13] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, D. Vrandečić, Introducing Wikidata to the linked data web, in: ISWC, 2014.
[14] R. Reiter, On closed world data bases, in: Logic and Data Bases, 1978.
[15] J. Minker, On indefinite databases and the closed world assumption, in: 6th Conference on Automated Deduction, 1982.
[16] Y. Ren, J.Z. Pan, Y. Zhao, Closed world reasoning for OWL2 with NBox, J. Tsinghua Sci. Technol. (2010).
[17] F. Baader, D. Calvanese, D. Mcguinness, D. Nardi, P. F. Patel-Schneider, The Description Logic Handbook, Cambridge University Press, 2007.
[18] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: The DL-lite family, J. Automat. Reason. (2007).
[19] D. McGuinness, F. Van Harmelen, et al., OWL web ontology language overview, W3C Recomm. (2004).
[20] J. Du, J.Z. Pan, Rewriting-based instance retrieval for negated concepts in description logic ontologies, in: ISWC, 2015.
[21] A. Analyti, G. Antoniou, C. Viegas Damásio, P. I., A framework for modular ERDF ontologies, AMAI (2013).
[22] A. Analyti, G. Antoniou, C. Viegas Damásio, G. Wagner, Negation and negative information in the W3C resource description framework, AMCT (2004).
[23] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison-Wesley, 1995.
[24] F. Darari, R. Prasojo, W. Nutt, Expressing no-value information in RDF, in: ISWC, 2015.
[25] L. Galárraga, S. Razniewski, A. Amarilli, F.M. Suchanek, Predicting completeness in knowledge bases, in: WSDM, 2017.
[26] S. Ortona, V. Meduri, P. Papotti, RuDiK: rule discovery in knowledge bases, VLDB (2018).
[27] R. Morante, C. Sporleder, Modality and negation: An introduction to the special issue, Comput. Linguist. (2012).
[28] W. Chapman, D. Hillert, S. Velupillai, M. Kvist, M. Skeppstedt, B. Chapman, M. Conway, M. Tharp, D. Mowery, L. Deleger, Extending the NegEx lexicon for multiple languages, Stud. Health Technol. Inform. (2013).
[29] S. Wu, T. Miller, J. Masanz, M. Coarr, S. Halgrim, D. Carrell, C. Clark, Negation's not solved: generalizability versus optimizability in clinical natural language processing, PLoS One (2014).
[30] J.Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, J. Liu, Content based fake news detection using knowledge graphs, in: ISWC, 2018.
[31] N. Cruz Díaz, Detecting negated and uncertain information in biomedical and review texts, in: RANLP, 2013.
[32] G. Szarvas, V. Vincze, R. Farkas, J. Csirik, The BioScope corpus: Annotation for negation, uncertainty and their scope in biomedical texts, in: BioNLP, 2008.
[33] I. Goldin, W. Chapman, Learning to detect negation with "not" in medical texts, in: SIGIR, 2003.
[34] I. Bärbäntan, R. Potolea, Towards knowledge extraction from electronic health records - automatic negation identification, in: IFMBE, 2014.
[35] W. Chapman, W. Bridewell, P. Hanbury, G. Cooper, B. Buchanan, A simple algorithm for identifying negated findings and diseases in discharge summaries, J. Biomed. Inform. (2001).
[36] I. Bärbäntan, R. Potolea, Exploiting word meaning for negation identification in electronic health records, in: AQTR, 2014.
[37] G. Karagiannis, I. Trummer, S. Jo, S. Khandelwal, X. Wang, C. Yu, Mining an "anti-knowledge base" from Wikipedia updates with applications to fact checking and beyond, PVLDB (2019).
[38] T. Safavi, D. Koutra, Generating negative commonsense knowledge, 2020, ArXiv.
[39] Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: A survey of approaches and applications, IEEE TKDE (2017).
[40] J. Lajus, L. Galárraga, F. Suchanek, Fast and exact rule mining with AMIE 3, in: ESWC, 2020.
[41] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014.
[42] K. Wiharja, J.Z. Pan, M.J. Kollingbaum, Y. Deng, Schema aware iterative knowledge graph completion, J. Web Semant. (2020).
[43] L. Jiang, A. Bosselut, C. Bhagavatula, Y. Choi, "I'm Not Mad": Commonsense implications of negation and contradiction, in: NAACL-HLT, 2021.
[44] G. Kasneci, F. Suchanek, G. Ifrim, M. Ramanath, G. Weikum, NAGA: Searching and ranking knowledge, in: ICDE, 2008.
[45] M. Yahya, D. Barbosa, K. Berberich, Q. Wang, G. Weikum, Relationship queries on extended knowledge graphs, in: WSDM, 2016.
[46] H. Arnaout, S. Elbassuoni, Effective searching of RDF knowledge graphs, JWS (2018).
[47] H. Bast, B. Buchhold, E. Haussmann, Relevance scores for triples from type-like relations, in: SIGIR, 2015.
[48] S. Huang, J. Liu, F. Korn, X. Wang, Y. Wu, D. Markowitz, C. Yu, Contextual fact ranking and its applications in table synthesis and compression, in: KDD, 2019.
[49] S. Razniewski, W. Nutt, Completeness of queries over incomplete Databases, VLDB, 2011.
[50] J.Z. Pan, D. Calvanese, T. Eiter, I. Horrocks, M. Kifer, F. Lin, Y. Zhao (Eds.), Reasoning Web: Logical foundation of knowledge graph construction and query answering, Springer, 2017.
[51] T. Tran, M. Gad-Elrab, D. Stepanova, E. Kharlamov, J. Strötgen, Fast computation of explanations for inconsistency in large-scale knowledge graphs, in: WWW, 2020.
[52] V. Balaraman, S. Razniewski, W. Nutt, Recoin: Relative completeness in Wikidata, in: Wiki Workshop At WWW, 2018.
[53] M. Ponza, P. Ferragina, S. Chakrabarti, A two-stage framework for computing entity relatedness in Wikipedia, in: CIKM, 2017.
[54] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: NIPS, 2013.
[55] S. Razniewski, V. Balaraman, W. Nutt, Doctoral advisor or medical condition: Towards entity-specific rankings of knowledge base properties, in: ADMA, 2017.
[56] I. Yamada, A. Asai, H. Shindo, H. Takeda, Y. Takefuji, Wikipedia2Vec: An optimized tool for learning embeddings of words and entities from wikipedia, 2018, ArXiv.
[57] E. Oren, R. Delbru, S. Decker, Extending faceted navigation for RDF data, in: ISWC, 2006.
[58] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, Trans. Inf. Syst. (2002).
[59] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, in: AAAI, 2016.
[60] V. Ho, D. Stepanova, M. Gad-Elrab, E. Kharlamov, G. Weikum, Rule learning from knowledge graphs guided by embedding models, in: ISWC, 2018.
[61] L. Galárraga, C. Teflioudi, K. Hose, F. Suchanek, Fast rule mining in ontological knowledge bases with AMIE+, VLDB (2015).
[62] D. Diefenbach, K. Singh, P. Maret, WDAqua-core0: A question answering component for the research community, in: ESWC, 2017.
[63] H. Arnaout, S. Razniewski, G. Weikum, J. Pan, Wikinegata: A knowledge base with interesting negative statements, in: PVLDB, 2021.
[64] L. Galárraga, C. Teflioudi, K. Hose, F.M. Suchanek, AMIE: association rule mining under incomplete evidence in ontological knowledge bases, in: WWW, 2013.