Enriching Open-world Knowledge Graphs with Expressive Negative Statements

Hiba Arnaout

A dissertation submitted towards the degree

Doctor of Engineering Science (Dr.-Ing.)

of the Faculty of Mathematics and Computer Science
of Saarland University

Saarbrücken, 2023.

Date of Colloquium:	27.10.2023
Dean of the Faculty:	Prof. Dr. Jürgen Steimle
Chair of the Committee:	Prof. Dr. Kurt Mehlhorn
Reviewers:	Dr. Simon Razniewski
	Prof. Dr. Gerhard Weikum
	Prof. Dr. Ralf Schenkel
Academic Assistant:	Dr. Anna Tigunova

Abstract

Knowledge Graphs (KGs) about real-world entities and their properties are an important asset in many AI applications. Web-scale KGs store almost only positive statements, and miss out on negative statements. Due to the incompleteness of open-world KGs, absent statements are considered *unknown*, rather than *false*. This dissertation makes the case for enriching KGs with informative statements that do *not* hold, and thus enhancing their usability in applications such as question answering and entity summarization. With potentially billions of candidate negative statements, we tackle four main challenges.

1. Correctness (or plausibility) of negative statements: operating under the Open-World Assumption (OWA), it is not sufficient to check if a candidate negative is not explicitly stated as positive in the KG, since it might be a missing positive. Methods to scrutinize large sets of candidates and prune false positives are crucial.

2. Salience of negative statements: the set of correct negative statements is very large but full of trivial or nonsensical statements, e.g., "*A cat cannot store data.*". Methods to quantify the informativeness of negatives are necessary.

3. Coverage of subjects: depending on the source of data and methods for retrieving candidates, some subjects or entities in the KG might receive zero candidate negatives. Methods must ensure the ability to discover negatives about almost any existing entity.

4. Complex negative statements: in some cases, expressing a negation requires more than one KG triple. For instance, "*Einstein did not receive an education*" is a false negative, but "*Einstein did not receive an education at a U.S. university*" is a true negative. Methods to generate conditioned negatives are needed.

This dissertation tackles these challenges as follows.

1. We first make the case for *selective materialization* of negative statements about entities in *encyclopedic* (well-canonicalized) open-world KGs, and formally define three types of negative statements: grounded, universally absent, and conditional negative statements. We present the *peer-based negation inference* method to compile lists of salient negatives about entities. The method computes relevant peers for a given input entity, and uses their positives to set expectations for the input entity. An expectation that does *not* hold is an immediate candidate negative, and is then scored using frequency, importance, and unexpectedness metrics.

2. We propose the *pattern-based query log extraction* method to extract salient negatives from rich *textual sources*. This method extracts salient negatives about an entity by harvesting large corpora, i.e., search engine's query logs, using a few handcrafted patterns with negative keywords.

3. We introduce the *UnCommonsense* method to generate salient negative phrases about everyday concepts in less-canonicalized *commonsense* KGs. This method is designed to handle negation inference, scrutiny, and ranking of short natural language phrases. It computes comparable concepts for a given target concept, infers candidate negatives from comparing their positives, and scrutinizes these candidates against the KG itself, as

well as Language Models (LMs) as an external source of knowledge. Finally, candidates are ranked using semantic-similarity-aware frequency measures.

4. To facilitate exploring our methods and their results, we implement *two prototype systems*. In *Wikinegata*, a system to showcase the peer-based method is developed where users can explore negative statements about 500K entities of 11 classes, and adjust different parameters of the peer-based inference method. They can also query the KG using triple patterns with negated predicates. In the *UnCommonsense* system, users can closely inspect what the method produces at every step, as well as browse negatives about 8K everyday concepts. Moreover, using the peer-based negation inference method, we create the first large-scale dataset on demographics and outliers in communities of interest, and show its usefulness in use cases such as identifying under-represented groups.

5. We release all datasets and code produced in these projects at https://www.mpi-inf.mpg.de/negation-in-kbs and https://www.mpi-inf.mpg.de/Uncommonsense.

Zusammenfassung

Wissensgraphen über Entitäten und ihre Attribute sind eine wichtige Komponente vieler KI-Anwendungen. Wissensgraphen im Webmaßstab speichern fast nur positive Aussagen und übersehen negative Aussagen. Aufgrund der Unvollständigkeit von Open-World-Wissensgraphen werden fehlende Aussagen als *unbekannt* und nicht als *falsch* betrachtet. Diese Dissertation plädiert dafür, Wissensgraphen mit informativen Aussagen anzureichern, die *nicht* gelten, und so ihren Mehrwert für Anwendungen wie die Beantwortung von Fragen und die Zusammenfassung von Entitäten zu verbessern. Mit potenziell Milliarden negativer Aussagen von Kandidaten bewältigen wir vier Hauptherausforderungen.

1. Korrektheit (oder Plausibilität) negativer Aussagen: Unter der Open-World-Annahme (OWA) reicht es nicht aus, zu prüfen, ob ein negativer Kandidat im Wissensgraphen nicht explizit als positiv angegeben ist, da es sich möglicherweise um eine fehlende Aussage handeln kann. Von entscheidender Bedeutung sind Methoden zur Prüfung großer Kandidatengruppen, und zur Beseitigung falsch positiver Ergebnisse.

2. Bedeutung negativer Aussagen: Die Menge korrekter negativer Aussagen ist sehr groß, aber voller trivialer oder unsinniger Aussagen, z. B. "*Eine Katze kann keine Daten speichern.*". Es sind Methoden zur Quantifizierung der Aussagekraft von Negativen erforderlich.

3. Abdeckung der Themen: Abhängig von der Datenquelle und den Methoden zum Abrufen von Kandidaten erhalten einige Themen oder Entitäten in dem Wissensgraphen möglicherweise keine negativen Kandidaten. Methoden müssen die Fähigkeit gewährleisten, Negative über fast jede bestehende Entität zu entdecken.

4. Komplexe negative Aussagen: In manchen Fällen erfordert das Ausdrücken einer Negation mehr als ein Wissensgraphen-Tripel. Beispielsweise ist "*Einstein hat keine Ausbildung erhalten*" eine inkorrekte Negation, aber "*Einstein hat keine Ausbildung an einer US-amerikanischen Universität erhalten*" ist korrekt. Es werden Methoden zur Erzeugung komplexer Negationen benötigt.

Diese Dissertation geht diese Herausforderungen wie folgt an.

1. Wir plädieren zunächst für die *selektive Materialisierung* negativer Aussagen über Entitäten in *enzyklopädischen* (gut kanonisierten) Open-World-Wissensgraphen, und definieren formal drei Arten negativer Aussagen: fundiert, universell abwesend und konditionierte negative Aussagen. Wir stellen die *Peer-basierte Negationsinferenz*-Methode vor, um Listen hervorstechender Negationen über Entitäten zu erstellen. Die Methode berechnet relevante Peers für eine bestimmte Eingabeentität und verwendet ihre positiven Eigenschaften, um Erwartungen für die Eingabeentität festzulegen. Eine Erwartung, die nicht erfüllt ist, ist ein unmittelbar negativer Kandidat und wird dann anhand von Häufigkeits-, Wichtigkeits- und Unerwartetheitsmetriken bewertet.

2. Wir schlagen die Methode *musterbasierte Abfrageprotokollextraktion* vor, um hervorstechende Negationen aus umfangreichen *Textquellen* zu extrahieren. Diese Methode extrahiert hervorstechende Negationen über eine Entität, indem sie große Korpora, z.B., die Anfrageprotokolle von Suchmaschinen, unter Verwendung einiger handgefertigter Muster mit negativen Schlüsselwörtern sammelt.

3. Wir führen die *UnCommonsense*-Methode ein, um hervorstechende negative Phrasen über alltägliche Konzepte in weniger kanonisierten *commonsense*-KGs zu generieren. Diese Methode ist für die Negationsinferenz, Prüfung und Einstufung kurzer Phrasen in natürlicher Sprache konzipiert. Sie berechnet vergleichbare Konzepte für ein bestimmtes Zielkonzept, leitet aus dem Vergleich ihrer positiven Kandidaten Negationen ab, und prüft diese Kandidaten im Vergleich zum Wissensgraphen selbst, sowie mit Sprachmodellen (LMs) als externer Wissensquelle. Schließlich werden die Kandidaten mithilfe semantischer Ähnlichkeitserkennungshäufigkeitsmaßen eingestuft.

4. Um die Exploration unserer Methoden und ihrer Ergebnisse zu erleichtern, implementieren wir *zwei Prototypensysteme*. In *Wikinegata* wird ein System zur Präsentation der Peer-basierten Methode entwickelt, mit dem Benutzer negative Aussagen über 500K Entitäten aus 11 Klassen untersuchen und verschiedene Parameter der Peer-basierten Inferenzmethode anpassen können. Sie können den Wissensgraphen auch mithilfe einer Suchmaske mit negierten Prädikaten befragen. Im *UnCommonsense*-System können Benutzer genau prüfen, was die Methode bei jedem Schritt hervorbringt, sowie Negationen zu 8K alltäglichen Konzepten durchsuchen. Darüber hinaus erstellen wir mithilfe der Peer-basierten Negationsinferenzmethode den ersten groß angelegten Datensatz zu Demografie und Ausreißern in Interessengemeinschaften und zeigen dessen Nützlichkeit in Anwendungsfällen wie der Identifizierung unterrepräsentierter Gruppen.

5. Wir veröffentlichen alle in diesen Projekten erstellten Datensätze und Quellcodes unter https://www.mpi-inf.mpg.de/negation-in-kbs und https://www.mpi-inf.mpg.de/Uncommonsense.

Acknowledgements

I would like to thank my supervisors, committee members, collaborators, colleagues, and family, for their immense support.

Contents

Introduction 1

Contents

1.1 Motivation and Problem

Structured knowledge is an important asset in many knowledge-intensive AI applications such as question-answering, dialogue agents, and recommendation systems. The knowledge is often stored in Knowledge graphs (KGs), aka Knowledge bases (KBs). Over the last decade, we have seen a rise of interest in KG construction, completion, and querying, resulting in notable public projects such as Wikidata [VK14] and Yago [SKW07], and commercial projects, such as the Amazon Product Graph [DHK+20] and the Google Knowledge Graph [Sin12]. These KGs store information such as "*Halle Berry won the 2002 Oscar for Best Actress*" as a `(subject, predicate, object)` triple `(halle berry, Won, academy award for best actress)`. One major limitation in these web-scale KGs is their inability to deal with negative information [FHP+06]. At present, popular KGs focus on obtaining positive statements, whereas statements such as "*Michelle Pfeiffer did not win an Oscar*" could only be inferred with the assumption that the KG is complete, i.e., the Closed-World Assumption (CWA). Under this assumption, absent information is considered *false*, which is not realistic in many cases, given that KGs are only pragmatic collections of positive statements. In reality, absent statements can be *true* but merely *unknown* to the KG.

Not being able to precisely 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, the 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. In general-use question-answering systems, asserting that *Switzerland* is *not* a member of the *EU*, for instance, ensures that the system will return a definite "no" when asked, instead of a list of relevant documents or text passages for users to scan.

1.2 State of the Art and its Limitations

The focus in constructing KGs is mostly on obtaining positive knowledge; nevertheless, some KGs contain a small number of negatives, e.g., Wikidata has almost 900K negative and 1.5B positive statements, about 100M entities. Negative statements are expressed using negated relations such as `NotIsA` in the commonsense KG ConceptNet [SCH17] and `DoesNotHaveQuality` in the general-purpose KG Wikidata [VK14]. Despite these efforts, the majority of existing negatives are trivial, e.g., `(person, NotIsA, tree)`, or cover specific domains, e.g., `NeverExceedAltitude` for airplanes in DBpedia [ABK+07] and `NotCausedBy` in the medical KG Knowlife [ESW15]. On actively compiling lists of negatives, existing approaches include extractive [RRP+19, RR20, KTJ+19] and generative [SK20, SZK21, KS20] methods. While extractive methods resort to textual corpora, including query logs and edit history, generative methods rely on pre-trained Language Models (LMs) [DCLT19, RWC+19] to generate negative statements, and in some cases, compute their salience. Extractive methods mainly suffer from low subject coverage, due to the limited resources or limited access to them. Generative methods, on the other hand, struggle to understand negation [KS20] which results in many false negatives. A more comprehensive review of related work can be found in Chapter 2.

1.3 Challenges

Given a KG, compiling lists of salient negatives about their entities requires overcoming the following challenges.
C1: Generate True Negatives. Web-scale open-world KGs are highly incomplete. This means that an absent statement does not mean its *falseness*. In the commonsense KG Ascent [NRW21, NRRW22], the absence of `(elephant, HasA,` *eye*`)` is clearly a missing positive. Therefore, relying only on the Closed-World Assumption (CWA) or even the Local Closed-World Assumption (LCWA) [Mot89, DCCBA08], where *some parts* of the KG are treated under CWA, is not sufficient. The task is then to *find internal and external sources of knowledge to support or contradict generated candidates*.

C2: Generate Salient Negatives. The goal of this work is to identify salient negative statements about entities. A salient negative statement is a statement that is noteworthy and informative.

Sample encyclopedic statements:

- `(stephen hawking, NotWon, nobel prize in physics)` - *salient*
- `(stephen hawking, NotWon, academy award for best director)` - *nonsalient*

Sample commonsense statements:

- `(peanut, NotIsA,` *nut*`)` - *salient*
- `(peanut, NotMadeOf,` *beer*`)` - *nonsalient*

For instance, both statements about `peanut` are related to *food*, but that `peanut` is not, in fact, a *nut* but a *legume* is more noteworthy than the obvious *not made of beer*.

It is important to emphasize the difference between *salience* and *type consistency*. While the latter is an easily checkable condition using the type system or taxonomy of the given KG, the former is more challenging. Enforcing type consistency will discard inconsistent statements such as `(stephen hawking, NotHasCapital, paris)`, since only an entity of type `Country` is allowed to have the relation `HasCapital`. On the other hand, salience requires identifying relevant subsets of entities and statements. For instance, entity `stephen hawking` under class `Person` is eligible for winning `Award`-type entities, including `oscar` and `nobel prize in physics`. Given his status as a famous physicist and his non-existing film career, not winning a

Nobel Prize in Physics is more informative than not winning an *Oscar*. The task is then to *infer sets of candidate informative negatives and propose ranking metrics to sort these potentially large sets.*

C3: Subject Coverage. Being able to compile a list of informative negatives about *any* existing KG entity is not obvious, especially for long-tail entities that we know very little about. For instance, there is even hardly any positive information about the Lebanese professional Basketball Player *Walid Doumiati*'s childhood or personal life in his Wikipedia article. The task here is to *choose rich sources for positive and/or negative information for higher coverage.*

C4: Beyond Simple Negative Statements. Simple negatives are statements that can be expressed using 1 triple with negated predicate `(albert einstein, NotEducatedAt, harvard)`, or 1 triple with an empty object `(albert einstein, HasTwitterAccount, no-value)`, expressing the facts that *"Einstein did not study at Harvard"* and *"is not on Twitter"*, respectively. What about *Einstein did not study at any U.S. university*? This can be observed by negating, for each *U.S.* university in the KG, that he did not study there, resulting in potentially hundreds of simple negatives that can only make the point when joined together. We cannot, on the other hand, use `(albert einstein, EducatedAt, no-value)` since he did study in other places outside the *U.S.* A more practical way of expressing this is to allow *conditional negatives* that can summarize and lift such cases. The task is then to *formally define the notion of multi-triple negatives, and propose a method to lift them from simple negatives*.

1.4 Contributions

We address these challenges in discovering salient negatives from encyclopedic KGs, query logs, and commonsense KGs, as follows.

- Encyclopedic KGs: we define the notion of negative statements in open-world encyclopedic KGs, and propose a method, the *peer-based negation inference* [ARW20, ARWP21c, ARWP21a] to compile salient ones. At first, we propose several measures to compute highly related entities for almost any existing input entity. These are later used to define parts of the KG where completeness is postulated, i.e. the LCWA (Local Closed-world Assumption). To ensure the correctness of candidates, we exploit the PCA (Partial Completeness Assumption [GTHS13, GTHS15]) for higher correctness. The PCA is one instantiation of the LCWA, which asserts that if a subject has *at least one* object for a given predicate, then there are no other objects beyond those that are in the KG. We finally propose four ranking metrics, combining frequency signals with popularity and probabilistic likelihoods, in a learn-to-rank model, to measure the salience of candidate negative statements. In an extension of this method, we explore the usage of ordered lists of peers, the *order-oriented peer-based negation inference* [ARWP21b], which shows an improvement in salience. Moreover, we define the notion of *conditional negative statements* to express complex negation, and present a method to lift them from previously inferred simple ones.

- Search Engine Query logs: we present a method for extracting interesting negatives about entities from query logs, the *pattern-based query log extraction* [ARW20]. We create a few handcrafted patterns, which we instantiate in the second step with entity mentions to retrieve textual occurrences. These patterns contain negative keywords to ensure the true negativeness of returned completions. The choice of the textual corpus is crucial for computing salient negatives, in this case, rich query logs of search engines.

- Commonsense KGs: we present *UnCommonsense* [ARWP22, ANRW23], a method for compiling lists of interesting negatives about everyday concepts from commonsense KGs. We revisit the challenges addressed in encyclopedic KGs and propose more fitting solutions for this new setting, i.e., less-canonicalized KGs. First, due to the lack of type systems (or taxonomies) in commonsense KGs, we propose using large collections of hypernymy relations to identify highly related concepts. Second, due to potential ambiguities of object phrases, using PCA has very little effect on ensuring the correctness of statements. Instead, we opt for internal and external measures for semantic similarity. We also integrate semantic similarity in the last ranking step to boost the score of near-synonymous candidates.

1.5 Publications

The research papers published towards constructing this dissertation are:

Chapter 3 (Negation Inference from Encyclopedic KGs) is based on:

- [ARW20] **Enriching Knowledge Bases with Interesting Negative Statements.**
  - except Section 5 of the paper
  Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. Proceedings of the 2nd Conference on Automated Knowledge Base Construction, AKBC 2020.

- [ARWP21a] **Negative Knowledge for Open-World Wikidata.**
  Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Proceedings of the Companion Proceedings of the 30th Web Conference, WWW Companion 2021.

- [ARWP21b] **Negative Statements Considered Useful.**
  Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Journal of Web Semantics, JWS 2021.

Chapter 4 (Negation Inference from Query Logs) is based on:

- [ARW20] **Enriching Knowledge Bases with Interesting Negative Statements.**
  - Section 5 of the paper
  Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. Proceedings of the 2nd Conference on Automated Knowledge Base Construction, AKBC 2020.

Chapter 5 (Negation Inference from Commonsense KGs) is based on:

- [ARWP22] **UnCommonsense: Informative Negative Knowledge about Everyday Concepts.**
  Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Proceedings of The 31st ACM International Conference on Information & Knowledge Management, CIKM 2022.

Chapter 6 (Systems and Resources) is based on:

- [ARWP21c] **Wikinegata: A Knowledge Base with Interesting Negative Statements.**
  Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Proceedings of the 47th International Conference on Very Large Databases Endowment, PVLDB 2021.

- [ANRW23] **UnCommonsense in Action! Informative Negations for Commonsense Knowledge Bases.**
  Hiba Arnaout, Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. Proceedings of The 16th ACM International Conference on Web Search & Data Mining, WSDM 2023.

- [ARP23] **Wiki-based Communities of Interest: Demographics and Outliers.**
  Hiba Arnaout, Simon Razniewski, and Jeff Z. Pan. Proceedings of The 17th International Aaai Conference On Web And Social Media, ICWSM 2023.

Furthermore, **Chapter 2 (Background)** and **Chapter 7 (Conclusion)** partially covers the following two publications, respectively:

- [RAGS21] **On the Limits of Machine Knowledge: Completeness, Recall and Negation in Web-scale Knowledge Bases.**
  Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, and Fabian Suchanek. Proceedings of the 47th International Conference on Very Large Databases Endowment, PVLDB 2021.

- [AR23] **Can Large Language Models Generate Salient Negative Statements?**
  Hiba Arnaout, Simon Razniewski. arXiv preprint arXiv:2305.16755.
  (Submitted to) Proceedings of The 32nd ACM International Conference on Information & Knowledge Management, CIKM 2023.

Resources including datasets and demonstrations are available at https://www.mpi-inf.mpg.de/negation-in-kbs and https://www.mpi-inf.mpg.de/Uncommonsense.

1.6 Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we present an overview of related work on negative knowledge in KGs. The three following chapters discuss our proposed methods for discovering salient negatives from encyclopedic KGs, query logs, and commonsense KGs, in Chapters 3, 4, and 5, respectively. In Chapter 6, we present the systems and resources created to showcase previously discussed methods. Finally, we summarize our findings in Chapter 7, and discuss open opportunities for future research.

Background 2

Contents

2.1 Knowledge Graphs

2.1.1 Encyclopedic Knowledge Graphs

Encyclopedic knowledge consists of facts about notable real-world entities, such as person (*Stephen Hawking*), location (*London*), and organization (*BBC*). Information about these entities is stored as `(subject, predicate, object)` triples, where `subject` is an entity, `object` can be either an entity or a literal (date, textual quote), and `predicate` is a pre-defined relation that links a `subject` and an `object`. For instance, the fact *Stephen Hawking studied at The University of Oxford* can be expressed as `(stephen hawking, EducatedAt, the university of oxford)`.

Notable web-scale projects include Wikidata [VK14], Yago [SKW07, HSBW13, RSH+16, TWS20], and DBpedia [ABK+07]. These KGs were constructed from encyclopedic sources of information, mainly Wikipedia, and later enriched using other web sources, such as news articles. Some of these projects go beyond the simple triple and include additional information, such as time and place, e.g., in Yago2, the triple `(barack obama, WasInauguratedAs, president of the united states)` with triple id #2, which is associated with the temporal triple `(#2, OccursOnDate, 2009-01-20)`.

In addition, many commercial search engines have access to their in-house knowledge graphs. For instance, Google uses the Google Knowledge Graph [Sin12] and Bing uses Microsoft Satori.

2.1.2 Commonsense Knowledge Graphs

Commonsense knowledge represents information about everyday concepts such as *gorilla*, *pancake*, and *newspaper*, which is shared by the majority of people. It is normally stored in commonsense KGs as `(subject, predicate,` *object phrase*`)` triples. Similar to encyclopedic

KGs, `subject` is a canonicalized entity, and `predicate` is a pre-defined relation. Instead of canonicalized objects, commonsense KGs often have *object phrases*. The list of predicates in these KGs is much shorter than in an encyclopedic KG (10812 predicates in Wikidata v. 19 in ConceptNet). Moreover, they lack crisp definitions, e.g., `CapableOf` or `HasProperty` in ConceptNet. This is remedied in the object component of the triple as a short expressive phrase, e.g., `(gorilla, CapableOf,` *inhabit the forests of central africa*`)`.

Notable projects include Ascent [NRW21, NRRW22], ConceptNet [SCH17], CYC [Len95], Webchild [TMSW14], ATOMIC [SBA+19], COMET [BRS+19], and Quasimodo [RRP+19].

ConceptNet, the most prominent of these projects, was mainly constructed using human crowdsourcing. CYC, the oldest of these KGs was built using handcrafted assertions by a team of knowledge engineers. Ascent, WebChild , TupleKB, and Quasimodo used fully automated triple extraction methods over selected text corpora, such as books, image tags, QA forums, and the C4 crawl [RSR+20]. WebChild relies on handcrafted extraction patterns, and TupleKB and Quasimodo use open information extraction with subsequent cleaning and ranking. Except for Ascent and Quasimodo, which stores additional informative facets, all these KGs are limited to `subject-predicate-object` triples.

ATOMIC is entirely based on a large-scale human compilation that focuses on inferential knowledge, i.e., if-then assertions. COMET is an autoregressive language model, fine-tuned on existing commonsense KGs, such as ATOMIC, that is used to predict `objects` for given `subject-predicate` pairs

Recently, an inference model has been proposed to build a KG specifically for *commonsense behavioral contradictions* [JBBC21], mainly through crowdsourcing, such as "*Wearing a mask is seen as responsible*" is the contradiction of "*Not wearing a mask is seen as carefree*".

2.1.3 Other Knowledge Graphs

Corporate KGs contain information about products or services, such as their types, from large E-commerce businesses. They are built to help manage their internal data and to provide a better customer service experience. For instance, Amazon has the Amazon Product Graph [DHK+20], Alibaba has the E-commerce Graph [LLY+20], and Bloomberg has the Bloomberg Knowledge Graph [Mei19].

Other domain-specific KGs cover other fields, such as the medical domain [EMSW14, ESW15], fictional universes [CRW21, CRW19, CRW20b], and music [OEAS+16].

2.1.4 Applications

Entity Search and Question Answering. KGs are a good source for compiling concise lists of salient short statements about given entities. Users can query general facts about *Stephen Hawking* or more precise ones about *Awards of Stephen Hawking*. Ranking and diversification models are sometimes incorporated to retrieve top-k triples about an entity [AE18]. Moreover, commercial search engines now incorporate KGs to improve their search results. For instance, the Google search engine uses the Google Knowledge Graph [Sin12], and Bing uses Microsoft Satori. Using these structured data stores, modern search engines are able to give direct answers to questions such as *Who voiced Woody in the Toy Story movie?*, which returns the entity `tom hanks` instead of the traditional search experience, where users had to scan several web pages to find their answers.

Recommender Systems and Chatbots. Latest advances in AI have made digital assistants, such as recommendation systems and chatbots more popular. Whether to find a good book

KG triples = {(tom hanks, Occupation, actor),(tom hanks, Won, oscar)}			
Query	**OWA**	**CWA**	**LCWA**
(tom hanks, Won, Oscar)?	True	True	True
(tom hanks, Won, nobel prize in physics)?	Unknown	False	Unknown
(tom hanks, Occupation, dentist)?	Unknown	False	False

Table 2.1: Querying under the OWA, CWA, and LCWA (CWA for occupations).

to read or a step-by-step recipe for a famous stew, users can interact with these services for relevant answers. Web-scale KGs, such as Wikidata can provide background information for general knowledge [GLH^{+}21, JMCC21], helping to improve the quality of answers. For instance, looking to watch *comedy* movies starring *Tom Hanks* is easier when the recommender system has access to the KG relations `Genre` and `CastMember`.

2.1.5 Completeness and Closed/Open-world Assumptions

The Closed-world assumption (CWA) is widely postulated for structured databases. It assumes that all statements not stated in the database to be true are *false*. In KGs, 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 KG are false, nor is it useful to consider the truth of all of them as unknown, since in many cases statements not contained in KGs are indeed not there because they are known to be false [RAGS21].

Between these two assumptions, there is also the so-called Local Closed-world assumption (LCWA) [Mot89, DCCBA08], sometimes referred to as the Partial Closed-world assumption (PCWA), where the OWA is used in general, while the CWA is applied to some parts of the KG, e.g., specific classes or predicates.

We highlight these different assumptions in Table 2.1, where only the predicate `Occupation` is considered complete under the LCWA. Given a toy KG, where all we know is that *Tom Hanks is an actor who won an Oscar*, under all assumptions, querying whether he did win an *Oscar* returns True, since the information is present in the KG. Querying whether he won a *Nobel Prize in Physics* returns False under the CWA. Under the OWA and LCWA, this is considered unknown due to the absence of information. Finally, querying whether he is a *dentist* returns unknown under the OWA, but False under the LCWA, since information about his professions is considered to be complete.

2.2 Negation in Existing Knowledge Graphs

In the following, we study two main types of negative knowledge in existing KGs, namely implicit and explicit negatives.

2.2.1 Implicit Negation

Deleted Statements. Triples that were once part of a KG but got subsequently deleted are promising candidates for negative knowledge [TS19]. For instance, the wrong statement

Statement	Date of removal	Reason of removal
(EthnicGroup, british people)	29.09.22	unsourced claim
(Movement, atheism)	12.07.22	reverted
(GreatRussianEncyclopediaID, 4695308)	08.07.22	no reason given
(PASMemberID, deceased/hawking)	13.06.22	reverted
(C-SPANPersonID, 53930)	10.05.22	ids migrated to P10660
(MemberOf, order of the companions of honour)	05.05.22	reverted
(AwardReceived, order of the british empire)	14.02.22	reverted
(CANTICID, a1038456x)	21.12.21	cleanup
(Occupation, writer)	11.12.21	cleanup
(Occupation, mathematician)	11.12.21	cleanup

Table 2.2: 10 most recent deletions for Wikidata's entity stephen hawking.

(joseph o'mahoney, DeathPlace, suffolk county) was once part of Wikidata, but was deleted in October 2017 and replaced with the correction (joseph o'mahoney, DeathPlace, bethesda). To study this more systematically, we identified deleted statements between two Wikidata versions from 1/2017 and 1/2018, focusing in particular on statements about people (0.5M deleted statements). We studied a random sample of 1K deletions, and we found that over 82% were just caused by ontology modifications, granularity changes, rewordings, or prefix modifications. Samples such as (gandhi, Lifestyle, vegetarian) was changed to (gandhi, Lifestyle, vegetarianism), (gandhi, DeathPlace, new delhi) changed to (gandhi, DeathPlace, gandhi smriti), and (james green, OxfordID, 101011386) changed to (james green, OxfordID, 11386). 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. They present mostly corrections of birth dates or location updates reflecting geopolitical changes. In Table 2.2, we show the 10 most recent deletions in stephen hawking's Wikidata page.

Count Predicates. A subtle way to express negative information is by matching count with enumeration predicates [GRW20, MRN16]. For instance, children-related information in Wikidata can be expressed using Child and NumberOfChildren. tom hanks has the values colin hanks, elizabeth hanks, chet hanks, and truman hanks for the former predicate and the value 4 for the latter. This means that Wikidata contains *all* of *Hank*'s children, and that any new inference or extraction is most likely to be false. These kinds of conclusions can be derived only over high-quality KGs, i.e., high precision/correctness. Otherwise, new findings could be used to refute old ones, which goes under the umbrella of the *research area on KG repair*. Moreover, while such count predicates exist in popular KGs, none of them has a formal way of dealing with all instance-based predicates, e.g., NumberOfAwards, NumberOfPositionsHeld, etc.

2.2.2 Explicit Negation

Negated Predicates. A few prominent KGs allow predicates that express explicit negative meaning. Predicates such as NeverExceedAltitude for airplanes in DBpedia [1], and IsNotCausedBy in the medical KG Knowlife [ESW15] are considered. KnowLife contains 0.002% negated statements. Moreover, Wikidata allows a few type-agnostic predicates, in-

[1] DBpedia dropped these predicates in later versions. It its latest version, as of 2022, no negated predicates are found.

cluding `DoesNotHavePart` and `DoesNotHaveQuality`, with a total of 869K statements. A more systematic case for negated predicates is found in ConceptNet, where the 6 main predicates have negated counterparts, namely `NotIsA`, `NotCapableOf`, `NotDesires`, `NotHasA`, `NotHasProperty`, and `NotMadeOf`, with a total of 14.1K negated statements in ConceptNet v5.5. We use these to construct a ground-truth dataset (See Chapter 5).

No-value Objects. Though not widely used, to assert the falseness of *all* possible objects for a certain `subject-predicate` pair, KGs such as Wikidata allow the triple to have an empty object, e.g., `(ludwig van beethoven, Child, no-value)`[2]. The total number of `no-value` statements in Wikidata is 20.6K.

Deprecation of Statements. KGs, like Wikidata, encourage flagging certain statements as incorrect as opposed to removing them. These are usually outdated statements or statements that are known to be false[3]. Wikidata also encourages editors to enter a reason for such cases. For instance, `(philip de laszlo, BirthPlace, budapest)`, who was born in 1869 has been flagged as incorrect and the reason given is *"at the specified time, this entity* (`budapest`) *may or may not have existed"*. This helps in spotting mistakes and prevents editors from repeatedly adding false information as positives. Yet we found that this mostly relates to errors coming from various import sources, and did not concern the active collection of *informative* negatives, as advocated in this dissertation.

Statements with Negative Polarity. In the commonsense KG Quasimodo [RRP+19], every statement is extended by a polarity value to express whether it is a positive or a negative statement, e.g., (`scientist, Has,` *academic degree*) with *polarity=positive* and (`baby, HasBodyPart,` *hair*) with *polarity=negative*. 5.6% (350K) of Quasimodo's statements are negative. Many of these negatives, however, are either inaccurate or nonsensical, e.g., (`show, ShowUpOn,` *netflix*), (`fish, HasProperty,` *halal*) (both with negative polarity). We collect statements with negative polarity and use them as a baseline in our experiments (see Chapter 5).

2.3 Related Areas

2.3.1 Negation in Logics and Data Management

In limited domains, logical rules and constraints, such as Description Logics [BCM+07, CGL+07] (or OWL) can be used to derive negative statements. For instance, the rule that every person has only one birthplace allows us 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 [MvH04], 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).

In [AADP13, AADW04], a thorough study on negative information in the Resource Description Framework (RDF) argues in favor of explicit negation. The study proposes 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"*).

Similar to the no-value triple-objects allowed in Wikidata, the notion of no-value in RDF is introduced in [AHV95]. It has been recently adapted in [DPN15] 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 a lack of

[2] https://www.wikidata.org/wiki/Q255

[3] https://www.wikidata.org/wiki/Help:Deprecation

information or actual negation.

The AMIE framework [GRAS17] employs 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 KG, inferring that all absent values for a `subject-predicate` pair are false. For our task, this could be a building block, namely when scrutinizing candidates, but it does not address the inference of *salient* negative statements.

RuDiK [OMP18] 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. Moreover, the mining also discovers many convoluted and exotic rules, e.g., *people whose body weight is less than their birth year cannot win the Nobel prize*, often with a large number of atoms in the rule body, and such rules are among the top-ranked ones.

2.3.2 Textual Information Extraction of Negation

Negation is an important feature of human language [MS12]. 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 [CHV$^+$13, WMM$^+$14]. Yet theories of conversational schemes indicate that negative statements can also be inferred from sentences that do not contain explicit negation. For instance, following Grice's maxims of cooperative communication [Gri75], a reasonable conclusion from the sentence "*John has two children, Mary and Bob*" is that nobody else is a child of *John*. Such inferences are called scalar implicatures, and they play a considerable role in language pragmatics [Car98].

A body of work targets negative knowledge in medical data and health records. In [Día13], a supervised system for detecting negation, speculation and their scope in biomedical data is developed, based on the annotated BioScope corpus [SVFC08]. In [GC03], 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 [BP14b], a rule-based approach based on NegEx [CBCB01], and a vocabulary-based approach for prefix detection were introduced. PreNex [BP14a] also deals with negation prefixes, such as *asymptomatic*. The work proposes to break terms into prefixes and root words to identify this kind of negation and rely on a pattern-matching approach over medical documents.

The work in [KTJ$^+$19] exploits the edit history of collaborative encyclopedias such as Wikipedia as a rich source of implicit negations. Editors make thousands of changes every day for various reasons, including fixing spelling mistakes, rephrasing sentences, updating information on controversial topics, and *fixing factual mistakes*. The work focuses on mining data from the last category. In particular, it looks at sentence edits in Wikipedia where only one entity or one number is changed, e.g., the sentence "*Heineken is Danish*" is updated to "*Heineken is Dutch*". To decide whether this update is in fact a factual mistake, several heuristics are applied, including monitoring how often a sentence is being updated (to exclude controversial topics where different editors have different opinions) and computing the edit distance between the entities (to exclude spelling corrections). It remains to be checked whether the edit removes or introduces a false statement. This is done by counting the number of supporting statements, e.g., web hits of each.

The work in [YBB$^+$16] proposes extending existing KGs with additional knowledge from textual web content. Extracted statements, using OpenIE methods [MSB$^+$], mostly compile lists of trial negatives, such as `(iran, NotAs, lebanon)`, `(sudan, NotIn, china)`.

Overall, the main limitation of the text-based methods is subject coverage. Most extractive models remain at the mercy of how rich the corpora used are. In the edit history work [KTJ+19], for instance, where Wikipedia is considered, the articles that are often updated are about prominent entities. The other challenge is linking extracted statements to the original KG. This requires canonicalization of entities and predicates, which is not a trivial task [WWKY18].

2.3.3 Pre-trained Language Models for Negation Generation

In recent years, Language Models (LMs) showed their ability to store factual knowledge, learned from pre-training data [PRR+19, SRI+20]. Via LM-probing, they can well predict positive facts with high accuracy, e.g., for e.g. *birds can [MASK]*, top predictions are *fly, sing, talk*. On the other hand, LMs, such as BERT [DCLT19], have been also repeatedly shown to struggle with explicit negation [KS20, TEGB20], e.g., for *birds cannot [MASK]*, top predictions are *fly, sing, speak*. More recent models, such as GPT-3 [RWC+19] and ChatGPT [Ope22] show more promising results when it comes to negation generation. We compare our proposed models with the former in Chapter 5, and share insights of our experiments on the latter, in Chapter 7.

The NegatER framework [SZK21, SK20] proposes a corruption-based negation inference model, which uses LMs to score their salience. Given a commonsense KG, e.g., ConceptNet, and a pre-trained language model (LM), e.g., BERT, the LM is fine-tuned using the KG's positives. This strengthens its ability to classify unseen true and false statements. Next, plausible candidate negatives are generated using dense k-nearest-neighbors retrieval, by either replacing the `subject` or the `object` with a neighboring phrase. In a final ranking step, the set of candidates is scored, using the fine-tuned LM, by descending order of *negativeness*, measured as the candidate's proximity to the decision threshold, or the model's gradient. Even though NegatER compiles lists of thematically-relevant negatives, one major limitation is that it generates many type inconsistent statements, due to the absence of a taxonomy, e.g., `(horse rider, NotIsA, expensive pet)`.

3 Negation Inference from Encyclopedic Knowledge Graphs

Contents

3.1 Introduction

Motivation. Notable web-scale encyclopedic KGs like Wikidata [VK14], DBpedia [ABK$^+$07], and Yago [SKW07], mostly store positive statements, e.g., (renée zellweger, Won, academy award for best actress), and are a key asset for many knowledge-intensive AI applications, such as question answering and recommendation systems. A major limitation of these KGs is their inability to deal with negative information [FHP$^+$06]. For example, (tom cruise, NotWon, academy award for best actor) could only be inferred with the major assumption that the KG is complete - the so-called *closed-world assumption* (CWA). Yet as KGs are only pragmatic collections of positive statements, the CWA is not realistic to assume, and there remains uncertainty whether statements not contained in a KG are false, or truth is merely unknown to the KG. This has direct consequences for the usage of KGs. For example, question-answering systems over KGs too often still return best-effort answers for queries such as *"Actors without Oscar"* or *"Children of Emmanuel Macron"*.

More generally, distinguishing whether an absent statement is true or false can boost the robustness of many applications: It is important in medicine, for instance, 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, the 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.

Approach. In this chapter, we make the case that important negative knowledge should be explicitly materialized. We motivate this selective materialization with the challenge of overseeing a huge space of false statements[1], and with the importance of explicit negation in search and question answering. To this end, we propose the peer-based negation inference method to compile lists of salient negative statements about encyclopedic entities. First, we select highly related entities to `e`, or *peers*. We then use these peers to derive positive expectations about `e`, where the absence of these expectations might be interesting. In this approach, we are assuming completeness only within the group of peers. This is followed by a ranking step where we use `predicate` and `object` prominence, frequency, and textual context in a learning-to-rank model.

3.2 Problem and Design Space

A KG is a set of statements, each being a triple `(s, p, o)`, where `s` stands for `subject`, `p` for `predicate`, and `o` for `object`.

Let K^i be an (imaginary) ideal KG that perfectly represents reality, i.e., contains exactly those statements that hold in reality. Under the OWA, (practically) available KGs, K^a contains correct statements, but may be *incomplete*, so the condition $K^a \subseteq K^i$ holds, but not the converse [RN11].

We, initially, distinguish two forms of negative statements, grounded and universally negative statements.

Definition 3.2.1 (Grounded Negative Statement)**.** ¬`(s, p, o)` is satisfied if `(s, p, o)` $\notin K^i$.

Definition 3.2.2 (Universally Negative Statement)**.** ¬∃`o: (s, p, o)` is satisfied if there exists no `o` such that `(s, p, o)` $\in K^i$.

Sample grounded and universally negative statements are displayed in Table 3.1. 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 (:BirthPlace :bruce willis :united states of america)`, while universally negative statements could be expressed via `ObjectAllValuesFrom` or `owl:complementOf` [EGK+14], e.g., `ClassAssertion (ObjectAllValuesFrom (:MarriedTo owl:Nothing) :leonardo dicaprio)`. 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 [BCM+07, PCE+17, TGES+20]. Yet compiling negative statements faces **two other challenges**:

1. 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 couple it with KG-based negative evidence signals using PCA [GTHS13].

2. The set of correct negative statements is *very* large, especially for grounded negative statements. Thus, unlike positive statements, negative statement construction/extraction needs a tight coupling with ranking methods.

[1] This means adding around 150,000,000,000 new negated statements to Wikidata, for instance.

Statement	Negation Type
¬(bruce willis, BirthPlace, united states of america)	grounded negative
¬∃o: (leonardo dicaprio, MarriedTo, o)	universally negative
¬(dubai, CapitalOf, united arab emirates)	grounded negative
¬(denmark, Currency, euro)	grounded negative
¬∃o: (george washington, EducatedAt, o)	universally negative

Table 3.1: Sample grounded and universally negative statements.

Research Problem. Given an entity e in encyclopedic open KG K^a, compile a ranked list of salient grounded and universally negative statements.

3.3 Peer-based Negation Inference

The peer-based negation inference method derives salient 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, namely other famous physicists, have won the nobel in physics. We may thus conclude that his not winning this prize could be an especially salient statement. Yet related entities also share other traits, e.g., many of them are u.s. citizens, while stephen hawking is british, and unlike him, a few are politicians, or can speak german. We thus need to devise ranking methods that take into account various clues such as frequency, importance, unexpectedness, etc.

3.3.1 Peer-based Candidate Retrieval

In the first stage, we compute a candidate set of negative statements using the CWA *on certain identified parts* of the KG, i.e., LCWA, to be ranked in the second stage. Given an input entity 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 computed using:
   - *Structured facets* of e [BRN18], such as Occupation, or Nationality for people, or Type for other entities.
   - *Graph-based measures* such as distance or connectivity [PFC17]. For instance, how many predicate-object pairs an input entity and a candidate peer share.
   - *Vector space embeddings* to reflect latent similarity between entities. For instance, for a given input entity, we identify the closest entities by measuring the cosine similarity between their pre-trained embedding vectors [YAS+20].
2. **Count statements:** We count the relative frequency of all predicate-object pairs (i.e., (_, p, o)) for grounded negatives and predicates (i.e., (_,p,_)) for universally negatives within the peer groups, and retain the maxima, in the case where candidate

negatives occur in several groups [2]. 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 hold for e in K^a.

The peer-based candidate retrieval is shown in Algorithm 1. In line 1, groups of peers $P[]$ are selected based on some blackbox function *peer_groups*, e.g., peers are entities that share the same `profession` as the input entity.

$$P = [P_1, ...P_n], \text{ with } n >= 1.$$

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

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

Subsequently, for each peer group, all the positive statements, that these peers have, are retrieved from the KG (lines 6 and 7), and stored as a list of candidate statements.

$$candidates = [st_1, ..., st_w].$$

A statement st_j in *candidates* is either a `predicate` P or a `predicate-object` pair PO.

The loop at line 10 iterates over the list of unique positive statements *ucandidates*, computes their relative frequency, and stores them in the final list of negatives N. Every negative statement in N is associated with its pre-computed relative score.

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

Across peer groups, it retains the maximum relative frequencies (hence, lines 12-13), in case a statement occurs across several. Before returning the top k results as output (line 19), it subtracts those already possessed by entity `e` (lines 17-18).

Scrutinize final candidates (*optional step*). To improve the correctness of final negatives, we drop those which do not satisfy the PCA assumption [GTHS13]. In particular, given a candidate grounded negative statement ¬`(s, p, o)` $\in K$, the statement is considered correct, i.e., truly negative, if and only if:

$$\exists \texttt{o}\prime : (\texttt{s, p, o}\prime) \in K, \text{ such that } \texttt{o} \neq \texttt{o}\prime$$

Suppose that we are given the candidate ¬`(stephen hawking, CitizenOf, australia)`. This statement satisfies the PCA assumption due to the presence of the positive statement `(stephen hawking, CitizenOf, u.k.` in K. The rationale behind this assumption is that salient `predicates`, such as `HasChild` or `CitizenOf`, especially for prominent entities, will either be covered completely, throughout the KG construction and maintenance process, or not at all. On the other hand, the candidate negative statement ¬`(stephen hawking, HasHobby, cooking)` does not satisfy the PCA rule, since the K knows nothing about `hawking`'s hobbies.

In our method, we make this step optional, because when applied, *universally negative statements* are no longer possible to infer. We show the effect of including this step on the correctness of the results in Section 3.3.3.

[2] This is not possible for the numerical-based similarity functions, like the graph-based measures and vector space embeddings, where only one group is constructed.

Algorithm 1: Peer-based candidate retrieval algorithm.

```
Input  : knowledge base KG, 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
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.
N[]= ∅                                                  ▷ Ranked list of negative statements about e.
for $P_i \in P$ do
    candidates = []                                     ▷ To store positives about entities in $P_i$.
    for $pe \in P_i$ do
        candidates+=collectP(pe)                        ▷ Collecting predicates that hold for peer pe.
        candidates+=collectPO(pe)                       ▷ Collecting predicate-object pairs that hold for pe.
    end
    ucandidates = unique(candidates)                    ▷ List of unique statements in candidates.
    for $st \in ucandidates$ do
        $sc = \frac{count(st,candidates)}{s}$   ▷ sc computes how many peers share the statement st (normalized).
        if $N[st] < sc$ then
            $N[st] = sc$
        end
    end
end
N-=collectP(e)                                          ▷ Remove statements e already has.
N-=collectPO(e)
return top(N, k)
```

Example 3.3.1. Consider the input entity e = brad pitt. Table 3.2 shows two examples of his peers and candidate negative statements about him. We instantiate the peering function to be based on shared objects for a certain type-like predicate. In particular, entities sharing the same profession with pitt, as in Recoin [BRN18]. In Wikidata, he has 8 professions, thus we would obtain 8 peer groups.

$$P = [\texttt{film actors, film directors, ..., models}], \text{ with } n = 8.$$

For readability, let us consider statements derived from only one of these peer groups, film actors. Let us assume 2 entities in that peer group:

$$P_{\text{film actor}} = \{\texttt{russel crowe, tom hanks}\}$$

The list of negative candidates, *candidates* as per the algorithm, are all the predicate and predicate-object pairs shown in the columns of the 2 actors. In this particular example, N is just *ucandidates* with scores from only the film actors group.

$$\begin{aligned} N = [(&\neg(\texttt{Award, oscar for best actor}), 1.0),\\ &(\neg\exists\texttt{o}(\texttt{Instagram, o}), 1.0),\\ &(\neg\exists\texttt{o}(\texttt{Convicted, o}), 0.5),\\ &(\neg\exists\texttt{o}(\texttt{child, o}), 1.0),\\ &(\neg(\texttt{Occupation, screenwriter}), 1.0),\\ &(\neg(\texttt{Citizen, u.s.}), 0.5)]. \end{aligned}$$

In this running example, we reduce the list of candidates, to the ones in the table, for readability, but statements such as (Instagram, russellcrowe) and (HasChild, colin hanks) are also considered.

Russell Crowe	**Tom Hanks**	**Brad Pitt**	**Candidates**
(Award, best actor)	(Award, best actor)	(Citizen, u.s.)	¬(Award, best actor), 1.0
(HasChild)	(Citizen, u.s.)	(HasChild)	¬(Occup., screenwriter), 1.0
(Occup., screenwriter)	(HasChild)		¬∃o(Instagram, o), 1.0
(Convicted)	(Occup., screenwriter)		¬∃o(Convicted, o), 0.5
(Instagram)	(Instagram)		

Table 3.2: Inferring candidate negatives for pitt using one peer group with 2 *actors*.

The statements that *hold* for pitt are then dropped. The updated list of candidates is:

$$\begin{aligned} N = [(\neg(\texttt{Award, oscar for best actor}), 1.0),\\ (\neg\exists\texttt{o}(\texttt{Instagram, o}), 1.0),\\ (\neg\exists\texttt{o}(\texttt{Convicted, o}), 0.5),\\ (\neg(\texttt{Occupation, screenwriter}), 1.0)]. \end{aligned}$$

Note that if we were to enable further scrutiny using the PCA, the statements ¬∃o(Convicted, o) and ¬∃o(Instagram, o) will be dropped, since the KG knows nothing about pitt's social media participation, nor his criminal record. The top-k of the rest of the candidates in N is finally returned. The top-3 negative statements, for this example, are ¬(Award, oscar for best actor), ¬(occupation, screenwriter), and ¬∃*o*(Instagram, o).

Also note that the "if" statement at line 12 is only necessary when multiple peer groups are considered: in this case, only the highest score is retained. In the original *full* example, pitt is also a member of the peer group models. The statement ¬(Occupation, screenwriter) was inferred twice, once from the actors group and once from the models group, with a relative frequency of 0.9 and 0.1, respectively. Here, we would retrain the candidate inferred from the actors group with 90% relative frequency. An alternative to picking the maximum score would be to compute the average of the scores across groups.

3.3.2 Ranking Negative Statements

Often, the final set of candidates is large. For example, using only 30 peers, the candidate set for pitt on Wikidata is already about 1500 statements, many of them nonsalient, e.g., ¬(brad pitt, HasChild, colin hanks). Therefore, ranking methods are necessary.

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 negation inference step 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 ¬(subject, predicate, object), its relevance might be reflected by the popularity[3] 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 former compared to the latter. In particular, on average, the monthly

[3] Wikipedia page views, for instance.

page views for `oscar for best actor` in 2022 is 143K, but only 1.3K for `london film critics' circle award`.

3. ***Predicate frequency (FRQ):*** When the inferred statement is universally negative, i.e., a `predicate` with an empty `object`, the frequency of the `predicate` can reflect the authority of the statement. To compute the importance of a `predicate`, we refer to its frequency in the KG. For example, ¬∃`o(Joel Slater, CitizenOf, o)` will receive a higher score (4.4M citizenships in Wikidata) than ¬∃`o(Joel Slater, Twitter, o)` (347K Twitter usernames in Wikidata).

4. ***Pivoting likelihood (PIVO):*** In addition to these frequency and 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 classifiers [RBN17], i.e., classifiers that decide whether an entity has a certain statement, each trained on the Wikipedia embeddings [YAS$^+$20] of 100 entities for which the statement holds (numerical label=1), and 100 for which it does not (numerical label =0) [4]. To classify a new (unseen) candidate statement, we then use the pivoting score of the respective classifier, i.e., the likelihood of the classifier to assign `e` to the group of entities having that statement. Note that this is a ranking, and not a correctness, metric. Assuming the inferred candidate is truly negative, we interpret the model's *misclassification* of a negative as a positive as an indicator of unexpectedness.

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

Definition 3.3.1 (Ensemble Ranking Score)**.**

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

We train a linear regression model where λ_1, λ_2, λ_3, and λ_4, are tuned using k-fold cross validation.

3.3.3 Evaluation

Setup. We instantiate the peer-based negation inference method with:

- Knowledge base: Wikidata.
- Peering function: the input entity (person)'s occupation, i.e., `predicate=P106`. The choice of this simple peering function was inspired by Recoin [BRN18].
- Number of peers: 30. In order to further ensure relevant peering, we also only considered entities as candidates for peers if their Wikipedia view count was at least a quarter of that of the input entity.
- Popularity metric POP in Equation 3.3.1: Wikipedia page views.

[4] On withheld data, logistic regression classifiers achieve 74% average accuracy on this task.

We randomly sample 100 popular Wikidata people. For each of them, we collect 20 candidate negatives: 10 with the highest PEER score, and 10 chosen at random from the rest of the retrieved candidates. We then use crowdsourcing [5] to annotate each of the total 2000 statements on *salience*, i.e., whether it was interesting enough to be added to a biographic summary about the input entity, with answer options: Yes, Maybe, and No. Each statement is shown to 3 annotators. Interpreting the answers as numeric scores 1, 0.5, and 0, for Yes, Maybe, and No respectively, we found a standard deviation of 0.29 and full agreement of all the annotators on 25% of the questions. Our final labels are the numeric averages among the 3 annotations.

The point of this task is to collect ground-truth salience labels to tune the parameters of our ensemble ranking scores.

Parameter Tuning. To learn optimal parameters for the ensemble ranking function 3.3.1, we trained a linear regression model using 5-fold cross-validation on the 2000 labels for salience. 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. We compute the quality of the top results of different variations of our ranking model using the Discounted Cumulative Gain (DCG) [JK02]. It is a measure that takes into consideration the rank of relevant statements and can incorporate different relevance levels. DCG is defined as follows:

Definition 3.3.2 (Discounted Cumulative Gain)**.**

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

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 relevance value $\in$ [0, 0.5, 1], depending on the annotator's assessment. We then average, for each 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 (IDCG), we obtained the normalized DCG (nDCG), which accounts for the variance in performance among queries, i.e., input entities.

Baselines and Peer-based Method Variants. After retrieving the initial set of candidates using the peer-based inference, with the above setup, we use the following baselines and configurations for ranking by salience:

1. **Naïve baseline (randomly ordered results)**: this baseline gives a lower bound on what any ranking model should exceed.

2. **Embedding-based baselines (link prediction)**: we experiment with TransE [BUGD+13] and HolE [NRP16]. For these two, we used pre-trained models [HSGE+18], on Wikidata (300K statements), covering prominent entities of different types, which we also enrich with all the statements about the sampled entities. We utilize their prediction score as a relevance score for each candidate grounded negative statement.[6]

3. **Frequency of `predicates`**: statements are ranked by descending order of their `predicates` frequency in Wikidata.

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

[6] Note that both models are not able to score statements about universal negation, i.e., empty `objects`, a trait shared with the object popularity heuristic in our ensemble.

4. **Popularity of `objects`**: statements are ranked by descending order of their `objects` view counts in Wikipedia.

5. **Pivoting score**: statements are ranked by descending order of their pivoting classifiers' prediction.

6. **Peer frequency**: statements are ranked by their relative frequency within the group of peers.

7. **Ensemble**: statements are ranked using the combination of metrics in Equation 3.3.1.

Results. Table 3.4 shows the average *nDCG* over the 100 entities for top-*k* negative statements, for *k* equals 3, 5, 10, and 20. 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 cannot be scored. The same applies to TransE and HolE. Ranking using the *Ensemble* metrics and the *Frequency of Property* outperform all other ranking metrics and the three baselines in *nDCG*, with an improvement over the random baseline of 20% for *k*=3 and *k*=5.

Examples of top-3 negatives for `albert einstein` are shown in Table 3.5. The random rank basically displays any candidate negation if it holds for at least one peer. For instance, `omar sharif` is `einstein`'s fellow *non-fiction writer*. This makes the negation *"Tarek Sharif, the child of Omar Sharif, not the child of Albert Einstein"* a valid candidate, hence, proving our argument of the necessity for a ranking step. Moreover, `omar sharif` is also an actor, which brings other topics to the candidates set of `einstein`, such as not winning some *entertainment awards*. This is where peer frequency makes a difference, as the majority of `einstein`'s peers are *not* actors. Even though it displays interesting negations, e.g., *despite his status as a famous researcher, `einstein` truly never formally supervised any PhD students*, the top-*k* result set, for the `predicate` frequency metric, lacks grounded negative statements. This is also reflected in the coverage column of Table 3.4. 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. We use crowdsourcing to assess the correctness of results. We collect 1K negatives about entities of type people, literature work, and organizations. Every statement is annotated 3 times, as either correct, incorrect, or ambiguous (evidence not found/unsure). 63% of the statements were found to be correct, 31% were incorrect, and 6% were ambiguous. We notice that most incorrect statements are due to KG completion issues. This task has a standard deviation of 0.23. Samples are shown in Table 3.3. For example, questions have been posed as to whether medical malpractice played a part in `franz liszt`'s death, hence the uncertainty about its real cause.

Statement	Correctness
¬(stephen king, Occupation, novelist)	Incorrect
¬(goldman sachs, Headquarters, new york city)	Incorrect
¬(jimmy carter, Occupation, lawyer)	Correct
¬(unesco, Headquarters, geneva)	Correct
¬(franz liszt, MannerOfDeath, natural causes)	Ambiguous

Table 3.3: Sample correctness annotations.

Ranking Model	Coverage(%)	$nDCG_3$	$nDCG_5$	$nDCG_{10}$	$nDCG_{20}$
Random	100	0.37	0.41	0.50	0.73
TransE [BUGD+13]	31	0.43	0.47	0.55	0.76
HolE [NRP16]	12	0.44	0.48	0.57	0.76
Predicate 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 3.4: Evaluation of different ranking metrics and baselines.

Random rank	Predicate frequency	Ensemble
`¬∃o(Instagram, o)`	`¬∃o(DoctoralStudent, o)`	`¬(Occupation, astrophysicist)`
`¬(Haschild, tarek sharif)`	`¬∃o(Candidacy, o)`	`¬ (Affiliation, communism)`
`¬(Award, bafta)`	`¬∃o(NobleTitle, o)`	`¬∃o(DoctoralStudent, o)`

Table 3.5: Top-3 results for `einstein` using 3 different ranking metrics.

Effect of PCA on Correctness. For a sample of 200 statements about 20 entities, half generated only relying on the LCWA within the group of peers, the other half additionally filtered to satisfy the PCA (the subject has at least one other object for that property [GTHS15]), we manually check for correctness, and found that 84% of PCA filtered statements are correct, and 57% for LCWA-based statements. This shows that enforcing the PCA step yields significantly more correct negatives, though the drawback brings losing the ability to predict universally negative statements.

3.4 Order-oriented Peer-based Negation Inference

Motivation. In the peer-based inference method, we assume a binary peer relation as the basis of peer group computation. For example, using structured facets, such as `Occupation`, to collect peers of `barack obama`, a highly relevant peer is `donald trump` (another contemporary politician, from the same country, who held the same position) and `anwar sadat` (a politician, from a different country, and a different era). Even peering functions that consider the order of peers, such as entity embeddings [YAS+20], are restricted to *one unlabeled* peer group. Moreover, the rank of the peer is not considered when ranking a statement. In expressive KGs, relatedness is typically graded and multifaceted, thus we show that introducing the notion of ordered peer sets and order-aware ranking to the peer-based inference method improves the quality of inferred negative statements and contributes to their explainability.

3.4.1 Ordered-peers Retrieval

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 `joe biden` are `donald trump` (predecessor), *barack obama* (pre-predecessor), *george w. bush* (pre-pre-predecessor), etc.

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

Example 3.4.1. Let us consider temporal recency of having a time-augmented `predicate-object`, e.g., `(Award-oscar for best acting)` as a similarity function w.r.t. the input entity `olivia colman`, who won the award in 2018. The ordered list of closest peer sets is [2017:{`frances mcdormand, gary oldman`}, 2016:{`emma stone, casey affleck`}, 2015:{`brie larson, leonardo diCaprio`}, 2014:{`julianne moore, eddie redmayne`}.., 1927:{`janet gaynor, emil jannings`}].

3.4.2 Order-aware Ranking

Ranking Metrics. Given an index of interest m $(m \leq n)$, we have a prefix list $S_{[1,m]}$ of ordered peers. For any negative statement candidate *stmt*, we 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., $VOL = |S_1 \cup ... \cup S_m|$. 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)*:** *PEER* denotes the fraction of entities in $S_1 \cup ... \cup S_m$ 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 change.

Example 3.4.2. Consider the input entity `e=olivia colman` from our example, with prefix length 3. For the statement `(Citizen, 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.

Research Problem. Given an entity `e` and an ordered group of peers, from an encyclopedic open KG K^a, compile a ranked list of salient negative statements about `e`.

Peer-order-aware Ranking. What makes a negative statement from an ordered peer set *salient*? 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, u.s.)` above is preferable over ¬`(Occupation, director)`, due to it being both reported on a larger set of peers, and with higher 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 equation as follows:

Definition 3.4.1 (Statement Salience Score)**.**

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

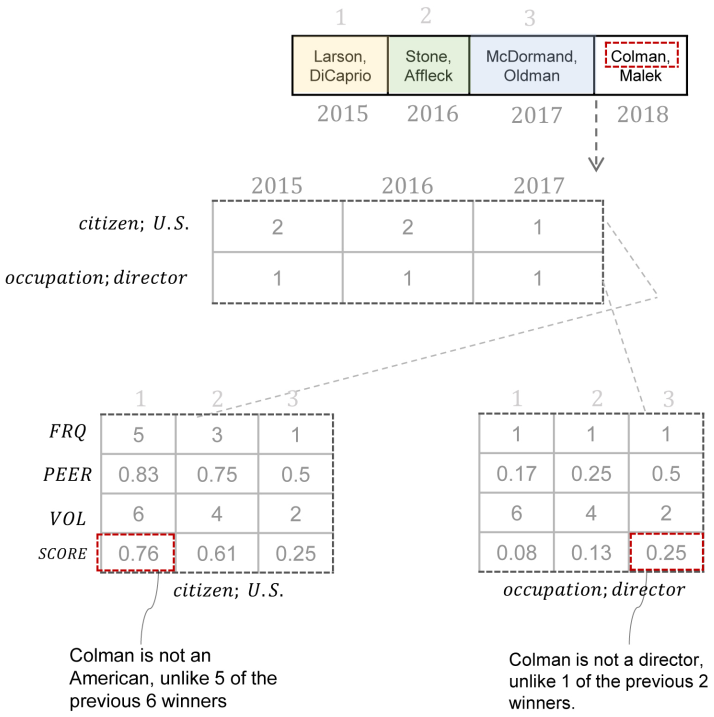


Figure 3.1: Retrieving salient negatives about `olivia colman`, using ordered peers (Oscar winners).

where α is 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 a larger number of peers is preferable. For example, for the same *PEER* value 0.5, we can have a statement that holds for 5 out of 10 peers, and a statement with a statement that holds for 1 out of 2 peers.

Example 3.4.3. Given the same example, the score for `olivia colman`'s negative candidate $\neg$(`Citizen, u.s.`) at prefix length 3 and $\alpha = 0.5$ is 0.76, with verbalization as "unlike 5 of the previous 6 `oscar` 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 $\neg$(`Occupation, director`) at prefix length 3 and $\alpha = 0.5$ is 0.08, with verbalization "unlike 1 of the previous 6 winners". The same statement with prefix length 2 will receive a score of 0.13, with verbalization "unlike 1 of the previous 4 winners". This example is illustrated in Figure 3.1.

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 candidate statement, based on a single pass over the prefix.

Example 3.4.4. Given the entity `e=olivia colman`, ordered groups of peers are identified in line 1.

$$L = [\texttt{Award-oscar, Award-bafta,..., Recipient-c.b.e}].$$

For readability, we proceed with one group, namely the *winners of Oscar for Best Acting*. It consists of ordered winners prior to e's win.

$L_{\texttt{Award-oscar}}$ = [{`frances mcdormand, gary oldman`} {`emma stone, casey affleck`}, {`brie larson, leonardo dicaprio`}, {`julianne moore, eddie redmayne`} .., {`janet Gaynor, emil jannings`}].

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

```
   Input  : knowledge base $KG$, entity $e$, ordered peer function $ordered\_peers$, number of results $k$,
            hypeparameter of salience scoring function $\alpha$
   Output : top-$k$ negative statement candidates for $e$
1  $\boldsymbol{L}[] = ordered\_peers(e)$   ▷ List of ordered peer group(s); Group $L_i$ at position $i$ is one ordered group.
2  $\boldsymbol{N}[] = \emptyset$   ▷ Ranked list of negative statements about $e$.
3  for $L_i \in \boldsymbol{L}$ do
4      $candidates$ = []
5      $pos$=position($L_i$, $e$)   ▷ Position of $e$ in the ordered group.
6      for $pe \in L_i$ do
7          if $pe == e$ then
8              continue
9          end
10         $candidates$+=$collectP(pe)$
11         $candidates$+=$collectPO(pe)$
12     end
13     $ucandidates = unique(candidates)$
14     for $st \in ucandidates$ do
15         $sc = scoring(st, L_i, e, pos, \alpha)$   ▷ Dynamic scoring of statement $st$ with different prefix lengths.
16         if $N[st] < sc$ then
17             $N[st] = sc$
18         end
19     end
20 end
21 $N$-=$collectP(e)$   ▷ Remove statements $e$ already has.
22 $N$-=$collectPO(e)$
23 return $top(N, k)$
24
25 Function scoring($st$, $S$, $e$, $pos$, $\alpha$):
26     max_sc = - inf; max_frq = - inf; max_vol = - inf;   ▷ Initializing the maximum score, FRQ, and VOL for statement $st$.
27     frq = 0; vol=0;   ▷ Initializing the FRQ and VOL of statement $st$.
28     for $j$ = $pos$; $j >= 1$; $j--$ do
29         vol += countentities($S[j]$)   ▷ Computing number of entities at position $j$.
30         frq += count($st$, $candidates$, $S[j]$)   ▷ Computing number of entities at position $j$ for which $st$ holds.
31         sc = $\alpha * \frac{frq}{vol} + (1 - \alpha) * log(frq)$   ▷ Computing the score of $st$ at position $j$.
32         if $sc > max\_sc$ then
33             max_sc = sc;
34             max_frq = frq;
35             max_vol = vol;
36         end
37     end
38 return max_sc, max_frq, max_vol
```

All statements of the peers are retrieved from the KG (lines 10 and 11). For every candidate statement *st*, the scores of the statement are 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

Statement	Time-based qualifier(s)
`(barack obama, PositionHeld, senator)`	Start: 3 Jan 2005; End: 16 Nov 2008
`(maya angelou, Award, presidential medal of freedom)`	`PointInTime`: 2010
`(donald trump, Spouse, melania trump)`	Start: 22 Jan 2005

Table 3.6: Samples of temporal information in Wikidata.

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 negatives N (line 16).

3.4.3 Evaluation

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

- **Time-based Qualifiers (TQ)**: Temporal qualifiers are time signals associated with statements about entities. In Wikidata, some of those qualifiers are `PointInTime` (P585), `StartTime` (P580), and `EndTime` (P582). A few samples are shown in Table 3.6.
- **Time-based Predicates (TP)**: Temporal predicates are predicates like `Follows` (P155) and `FollowedBy` (P156) indicating a chain of entities, ordered from oldest to newest, or from newest to oldest. For instance, novels of `leo tolstoy`: `[the cossacks, FollowedBy; war and peace, FollowedBy, anna karenina, ..]`

We create TQ groups from aggregating information about people sharing the same timestamp-extended `predicate-object` pairs. For example, `(PositionHeld, president of the u.s.)` is one TQ group, where members will have a `StartTime` for this position, as well as an `EndTime`. In case of the absence of an `EndTime`, this implies that the statement holds to this day (`donald trump`'s statement in Table 3.6). In other words, we aggregate entities sharing the same `predicate-object` pair, which are treated as the peer group's title, and rank them in ascending order of time qualifiers. For `PointInTime`, we simply rank the dates from oldest to newest, and for the `StartDate/EndDate`, we rank the end dates from oldest to newest. If the `EndDate` is missing, the entity is moved to the newest slot.

We collect a total of 19.6K TQ groups (13.6K using the `StartDate/EndDate` qualifiers, and 6K using the `PointInTime` 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 [7].

We create TP groups by first collecting all entities reachable by one of the transitive predicates, `Follows` (P155) and `FollowedBy` (P156). Considering each of the collected entities as a source entity, we compute the longest possible path of entities with only transitive properties. This path consists of an ordered set of peers. To avoid the problem of double-branching (one entity followed by two entities), we consider the two directions separately, but one path is chosen at the end; the one with the maximum length. The total number of TP groups is 19.7K groups. We limit the size of the groups to at least 10 and at most 150 [8].

Setup and Baseline. We collect 100 entities, that belong to at least one ordered set of peers, from Wikidata: 50 people and 50 literature works. We collect top-5 negative statements for each

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

[8] We do not truncate the groups, we simply disregard any group smaller or larger than the thresholds.

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

Table 3.7: Correctness evaluation.

of those entities (for people, we consider TQ groups, and for literature works, TP groups). We made this choice due to the lack of `Person`-type entities with transitive properties. In case an entity belongs to several groups, we merge all the results it is receiving from different groups, rank them, and retrieve the top-5 statements. Similarly, as a baseline, using the peer-based inference method (unordered peer groups), instantiated with cosine similarity on Wikipedia embeddings [YAS$^+$20] as the similarity function, we collect the top-5 negative statements for the same entities. We end up with 1K statements, 500 inferred by each model.

Correctness. We randomly collect 400 negative statements from the 1K statements collected above, 200 from each model (100 about people, and 100 about literature works). We then assess the correctness of each method using crowdsourcing. We show each statement to 3 annotators, asking them to choose whether this statement is correct, incorrect, or ambiguous. Results are shown in Table 3.7. The order-oriented inference method clearly infers *fewer incorrect statements* by 9 percentage points for people and 5 for literary works. It also produces more correct statements for people by 10 percentage points and literature work by 3. The percentage of queries with a 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 sample 1K entities from each dataset and test whether it is a member of at least one ordered set, thus the ability to infer negative statements about it. For TQ and TP groups, we randomly sampled 1K people, resulting in a coverage of 54% for both. Although the order-oriented method produces better negative statements on both notions of correctness and salience (as we will see next), it does not outperform our previous method 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 embeddings.

Salience. To assess the quality of our inferred statements from the order-oriented inference method against the baseline (the peer-based inference method), we present the annotators with two sets of top-5 negative statements about a given entity, and ask them to choose the more interesting set. The total number of opinions collected, given 100 entities, with 3 annotations each, is 300. To avoid biases, we repeatedly switched the position of the sets. Results are shown in Table 3.9. Overall results show that the order-oriented version of the method is preferred for 10% of negatives, 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 the peerness of 2 entities, by automatically producing expressive labels for peer groups (e.g., presidents of the u.s., winners of best actor academy award); and ii) it maximizes the *peerness* within a group. For instance, with Wikipedia embedding [YAS+20], 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, under u.s. presidents.

Statement **Order-oriented peer-based Explanation** **Peer-based Explanation**
¬(emmanuel macron, Member, national assembly) unlike 29 of 36 members of La République En Marche party unlike 70 out of 100 similar people
¬(tim berners-Lee, CitizenOf, u.s.) unlike 101 of the previous 115 winners of the MacArthur Fellowship unlike 53 out of 100 similar computer scientists
¬(michael jordan, Occupation, basketball coach) unlike 27 of the previous 49 winners of the NBA All-Defensive Team 31 out of 100 similar people
¬(theresa may, PositionHeld, opposition leader) unlike 11 of the previous 14 Leaders of the Conservative Party unlike 10 out of 100 similar people
¬(cristiano ronaldo, CitizenOf, brazil) unlike 4 of the previous 7 winners of the Ballon d'Or unlike 20 out of 100 similar football players

Table 3.8: Sample Negative statements and their verbalizations.

Explainability. One main contribution that our order-oriented inference method offers are the *verbalizations* produced with every inferred statement. In other words, it can, unlike the peer-based inference method, produce more concrete explanations of the salience of the inferred negatives. For example, the inferred statement ¬(abraham lincoln, CauseOfDeath, 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*" (since the reason for the similarity is latent and only restricted by entity type), 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 conduct a crowdsourcing task with 100 useful negations that were inferred by both methods from our previous experiment. For every negative statement, the annotator is shown two different verbalizations on "why is this negative statement noteworthy?". We ask the annotator to choose the better explanation, they 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%. More examples are shown in Table 3.8.

	People	**Literature Work**
Chosen method by annotators	%	%
Peer-based inference	42	44
Order-oriented peer-based inference	**52**	**54**
Either/Neither	6	2

Table 3.9: Salience of order-oriented and peer-based methods.

3.5 Conditional Negative Statements

Motivation. 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 `predicate`. 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 `predicate`. 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 3.5.1 (Conditional Negative Statement)**.** 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.

We call the predicate p′ the *aspect* of the conditional negative statement, and o′ the aspect's value.

Example 3.5.1. Consider the statement that `einstein` did not study at any `u.s.` university. It could be written as ¬∃o:(albert einstein, EducatedAt, o), (o, LocatedIn, 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:(albert einstein, EducatedAt, o), (o, IsA, private university), as none of these schools are private.

As before, the challenge is that there is a very large search space 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 `predicate`, 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

Grounded negative statements	Conditional negative statements
¬(EducatedAt, m.i.t.)	¬∃o (EducatedAt, o) (o, LocatedIn, u.s.)
¬(EducatedAt, stanford)	¬∃o (EducatedAt, o) (o, IsA, private university)
¬(EducatedAt, harvard)	

Table 3.10: Negative statements about einstein, before and after lifting.

Predicate	Aspect(s)
EducatedAt	LocatedIn, IsA
Award	SubclassOf
PositionHeld	PartOf

Table 3.11: A few samples of predicates and their aspects.

conditional negative statements. A crucial step is to define this lifting operation, and what the search space for this operation is.

Example 3.5.2. With the einstein example, shown in Table 3.10, we could start from three relevant grounded negative statements that einstein did not study at m.i.t., 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 [ODD06]. For a manual definition, we assume the condition to be in the form of a single triple pattern. A few samples are shown in Table 3.11. For EducatedAt, 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, "e did not win any category of nobel prize"; and for PositionHeld, "e did not hold any position in the house of representatives".

Research Problem. Using a set of salient grounded negative statements about an input entity e, construct a list of salient conditional negative statements.

3.5.1 Lifting Conditional Negative Statements

We propose Algorithm 3. Consider e=albert einstein, and the set of possible aspects *ASP* for lifting consisting of only two aspects about EducatedAt, for readability.

$$ASP = [(\texttt{EducatedAt: LocatedIn, IsA})].$$

The three grounded negative statements about einstein with EducatedAt predicate are:

$$NEG = [\neg(\texttt{EducatedAt: m.i.t., stanford, harvard})].$$

The loop at line 2 considers every predicate (p) in *NEG* (e.g., EducatedAt), and collect its aspects in line 3. For this example, the list of aspects *asp* for this predicate consists of the location and the type of educational institution.

$$asp = [\texttt{LocatedIn, IsA}].$$

Algorithm 3: Lifting grounded negative statements algorithm.

```
Input  : knowledge base $KG$, 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 $e$
1  $cond\_NEG$= $\emptyset$                               ▷ Ranked list of conditional negations about $e$.
2  for $p \in NEG$ do
3      $asp$ = $ASP[p]$                                  ▷ Retrieving aspects of predicate $p$.
4      for $a \in asp$ do
5          for $o \in NEG[p]$ do
6              $cond\_NEG$ += getaspvalues($KG$, $a$, $o$)   ▷ Collecting aspect values about $o$.
7          end
8      end
9  end
10 $cond\_NEG$-=$inKG(e, cond\_NEG)$
11 return $top(cond\_NEG, k)$
```

At line 4, the loop visits every aspect a in asp and look for aspect values (i.e., *the locations and types of Einstein's schools*). $NEG[p]$ are the objects that share the same predicate in the grounded negative statements list.

$$NEG[p] = [\texttt{m.i.t., stanford, harvard}].$$

For every object o, aspect values are collected and their relative frequencies are stored. For readability, line 6 is only a high-level version of this step. As mentioned before, the aspects are manually pre-defined, but their values are automatically retrieved.

$$\begin{aligned}
getaspvalues(\text{Wikidata}, \texttt{LocatedIn, m.i.t.}) &= [\texttt{u.s.}].\\
getaspvalues(\text{Wikidata}, \texttt{LocatedIn, stanford}) &= [\texttt{u.s.}].\\
getaspvalues(\text{Wikidata}, \texttt{LocatedIn, harvard}) &= [\texttt{u.s.}].\\
getaspvalues(\text{Wikidata}, \texttt{IsA, m.i.t.}) &=\\
[\texttt{institute of technology, private university}].&\\
getaspvalues(\text{Wikidata}, \texttt{IsA, stanford}) &=\\
[\texttt{research university, private university}].&\\
getaspvalues(\text{Wikidata}, \texttt{IsA, harvard}) &=\\
[\texttt{research university, private university}].&
\end{aligned}$$

Hence the aspect value for `EducatedAt`, namely `(LocatedIn, 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, EducatedAt, o) (o, LocatedIn, u.s.), 3),(¬∃o(einstein, EducatedAt, o) (o, IsA, private university), 3)]`.

3.5.2 Evaluation

Setup. We evaluate our lifting technique to retrieve salient conditional negative statements, based on three criteria: **(i) compression, (ii) correctness, and (iii) salience**. We collect the top

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

Table 3.12: Salience of conditional negative statements.

Negative statements	Conditional negative statements
`¬(Occupation, film director)`	`¬∃o:(Occupation, o) (o, SubclassOf, director)`
`¬(Occupation, theater director)`	`¬∃o:(MarriedTo, o)`
`¬(Occupation, television director)`	`¬∃o:(Child, o)`
`¬∃o:(MarriedTo, o)`	
`¬∃o:(Child, o)`	

Table 3.13: Top-3 negative statements about `leonardo dicaprio`, before and after lifting.

200 negative statements about 100 entities (people, organizations, and artwork), and then lift grounded negative statements to construct conditional negatives.

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

Correctness. We ask 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[9], we manually convert 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%.

Salience. For every entity, we show 3 annotators 2 sets of top-3 negative statements: a grounded and universally negative statements set, and a conditional negative statement set, and ask them to choose the one with more interesting information.

Results are shown in Table 3.12. 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 outperformance 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 a meaningful manner, at the same time, allowing more diverse statements to be displayed in a top-k set.

An example is shown in Table 3.13, 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 to direct any kind of creative work (films, plays, television shows, etc..).

3.6 Use Cases

We highlight the relevance of negative statements in 3 use cases:

1. Entity summarization on Wikidata.

2. Decision support with hotel data from Booking.com.

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

3. Question answering on various structured search engines.

3.6.1 Entity Summarization

In this experiment, we analyze whether mixed positive-negative statement sets 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.

Setup. We collect 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 infer, we collect salient positive statements about those entities.[10] We then compute for each entity `e`, a sample of 10 positive-only statements, and a mixed set of 7 positive and 3 *correct*[11] negative statements, produced by the peer-based method. We rely on peering using Wikipedia embeddings [YAS+20]. Annotators are 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 collect 3 opinions per task.

Results. Overall results show that mixed sets with negative information were preferred for 72% of the entity summaries, sets with positive-only statements were preferred for 17% of the summaries, and the option "both or neither" was chosen for 11% of the summaries. Table 3.15 shows results per each considered type. The standard deviation is 0.24, and the percentage of queries with full agreement is 22%. Table 3.14 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.

Positives-to-Negatives Best Ratio. We want to find out the ideal portion of negative statements to be added to a positive set of statements while maintaining the same, or better, *nDCG*. Similar to previous crowdsourcing tasks, we ask the annotators whether, given a statement, they would add it to a biographical summary about a given entity. Starting with a positive-only set, the decision to replace one positive statement with a negative must respect the constraint of not decreasing the relevance gain (i.e., *nDCG*) of the top-k results. Once it does, we stop the process of adding negatives and report the best positive-to-negative ratio. We find that the ideal portion of negative statements within top-k statements about `e` for k=3, 5, 10, and 20, is 1 for k=3 and k=5, 2 for k=10, and 5 for k=20.

3.6.2 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,

[10] We defined a number of common/useful `predicates` to each of type, e.g., for people, "`PositionHeld`"is a salient `predicate` for positive statements.

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

daily mirror	
Pos-only	**Pos-and-neg**
(OwnedBy, reach plc) (NewspaperFormat, tabloid) (Country, u.k.) (LanguageOfWork, english) (IsA, newspaper) ...	**¬(NewspaperFormat, broadsheet)** (NewspaperFormat, tabloid) **¬(Country, u.s.)** (LanguageOfWork, english) **¬(OwnedBy, news u.k.)** ...
peter the great	
Pos-only	**Pos-and-neg**
(MilitaryRank, general officer) (OwnerOf, kadriorg palace) (Award, order of the elephant) (Award, order of st. andrew) (Father, alexis of russia) ...	(MilitaryRank, general officer) (OwnerOf, kadriorg palace) **¬(PlaceOfDeath, moscow)** (Award, order of st. andrew) **¬(Award, knight of the order of st. alexander nevsky)** ...
twist And shout	
Pos-only	**Pos-and-neg**
(Composer, phil medley) (Performer, the beatles) (Producer, george martin) (IsA, musical composition) (LyricsBy, phil medley) ...	**¬(Composer, paul mcCartney)** (Performer, the beatles) **¬(Composer, john lennon)** (IsA, musical composition) **¬(LyricsBy, paul mcCartney)** ...

Table 3.14: Results for the entities daily mirror (newspaper), peter the great (person), and twist and shout (song).

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

Table 3.15: Positive-only vs. positive with negative entity summaries.

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.

Setup. 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 infer salient negative features. For peering, we considered all other hotels in *India*, and for ranking, we compute peer frequencies. We then use crowdsourcing over the results of 100 hotels. We ask 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 can choose one of the sets, or either. For every hotel, we request 3 annotators.

Results. Table 3.16 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 3.17 shows three hotels with salient negative features. Although *Hotel Asia The Dawn* lists 64 positive features, negative information such

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

Table 3.16: Salience of hotels' negative features.

Hotel	# 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-hr front desk
Hotel Asia The Dawn	64	¬ Air conditioning; ¬ Free Wifi; ¬ Free parking

Table 3.17: Negative statements for hotels in India.

as that it does not offer air conditioning and free Wifi may give important clues for decision making.

Moreover, we collect 20 pairs of hotels from the same dataset, and show every pair's Booking.com pages to 3 annotators. We ask them to choose the better hotel for them. Then, we show them negative features about the pair, and ask them whether this *new* information would change their mind on their initial decision. A screenshot of the task is shown in Figure 3.2. 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 disregards whether they have changed their decision or retained their initial choice.

3.6.3 Question Answering

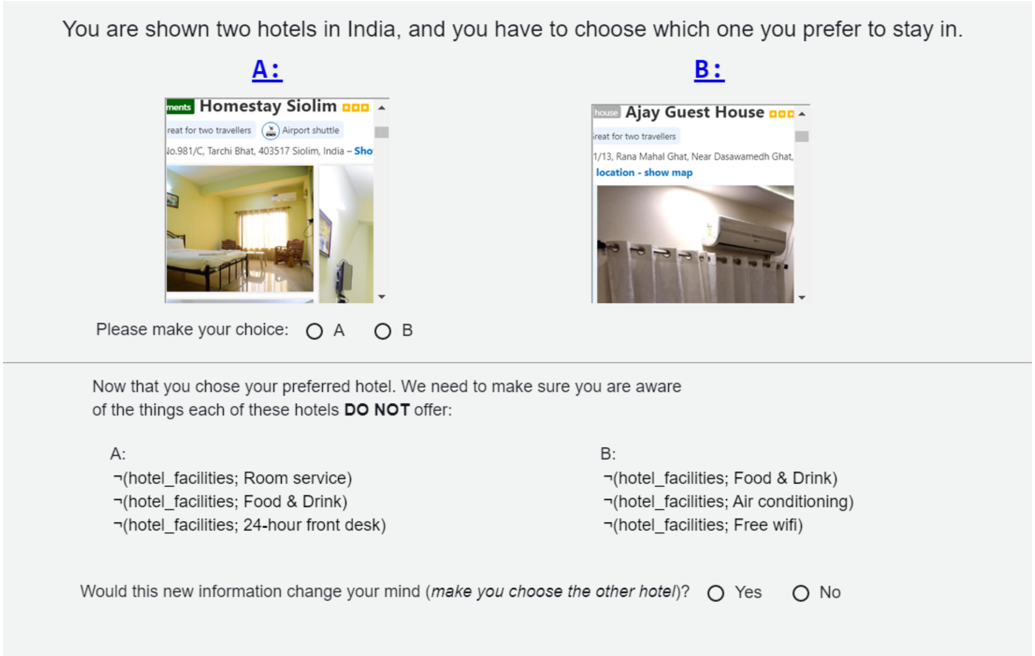
In this experiment, we compare the results to negative questions over a diverse set of sources.
Setup. We manually compile 20 questions that involve negation, such as *"actors with no Oscars", "actors with no spouses", "film actors who are not film directors", "football players with no Ballon d'Or", "politicians who are not lawyers"*. We compare them over four highly diverse sources:

- Google Web Search (increasingly returning structured answers from the Google knowledge graph [Sin12]).
- WDAqua [DSM17] (an academic state-of-the-art KGQA system).
- Wikidata SPARQL endpoint [12] (direct access to structured data).
- Our peer-based negation inference method.

For Google Web Search and WDAqua, we submit the queries in their textual form, and consider answers from Google if they come as structured knowledge panels. For Wikidata and peer-based inference, we transform the queries into SPARQL queries[13], which we either fully execute over the Wikidata SPARQL endpoint, or execute over the subsets of negatives, inferred and scored using our peer-based method. Note that all queries were safe since they

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

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



You are shown two hotels in India, and you have to choose which one you prefer to stay in.

A:

ments Homestay Siolim
reat for two travellers
Airport shuttle
No.981/C, Tarchi Bhat, 403517 Siolim, India – Sho

B:

house Ajay Guest House
reat for two travellers
1/13, Rana Mahal Ghat, Near Dasawamedh Ghat,
location - show map

Please make your choice: ○ A ○ B

Now that you chose your preferred hotel. We need to make sure you are aware of the things each of these hotels **DO NOT** offer:

A:
¬(hotel_facilities; Room service)
¬(hotel_facilities; Food & Drink)
¬(hotel_facilities; 24-hour front desk)

B:
¬(hotel_facilities; Food & Drink)
¬(hotel_facilities; Air conditioning)
¬(hotel_facilities; Free wifi)

Would this new information change your mind (*make you choose the other hotel*)? ○ Yes ○ No

Figure 3.2: Decision support, with negative features statements, on hotel data.

were designed to always ask for a class of entities (e.g., entities of occupation actor) that do not satisfy a certain property (e.g., *having won the Oscar*), which was captured via SPARQL MINUS with a shared variable. For each method, we then self-evaluate the number of results, the correctness, and the relevance of the top-5 results.

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 continue the assessment over a sample of 5 queries. Wikidata SPARQL returns 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 KGs 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.

3.7 Related Work

The problem of compiling informative negative statements about entities is new, so there are no directly comparable methods. Nevertheless, there is prior work on rule mining over KGs [GTHS15] that is conceivably useful in our context.

Most notably, [GRAS17] employed rule mining to predict the completeness of properties for given entities. This corresponds to learning whether partial completeness holds in a local part of the KG, 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 *interesting* negative statements.

[OMP18] devised 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 for predicting negative statements. On the other hand, mining also discovers many convoluted and exotic rules (e.g., people whose body weight is less than their birth year cannot win the *Nobel prize*), often with a large number of atoms in the rule body, and such rules are among the top-ranked ones. Even good rules, such as "people with birth year after 2000 do not win the *Nobel prize*", are not that useful for our task. Such rules predict way too many – correct, but uninformative – negative statements, essentially enumerating a huge set of people who are not Nobel laureates.

[OMP18] also proposed a precision-oriented variant of PCA that assumes negation only if the `subject` and `object` are connected by at least one other relation. This condition is rarely met in interesting cases. For instance, none of the negative statements in Table 3.5 have alternative connections between `subject` and `object` in Wikidata.

Another related line of work is learning which attributes are mandatory in a KG, for only non-mandatory absent predicates are candidates for universal absence. [LS18] exploits density differences along type hierarchies to this end. This could be an initial filter towards discovering negative statements, but does not address our key problem of inferring when a missing statement is truly negative and salient.

3.8 Conclusion

The work covered in this chapter makes the first comprehensive case for explicitly materializing salient negative statements from open-world encyclopedic KGs. We introduced a statistical inference approach, namely the peer-based and order-oriented peer-based negation inference methods, on retrieving, ranking, and verbalizing the salience of candidate negative statements, based on expectations set by highly related peers.

Negation Inference from Query Logs 4

Contents

4.1 Introduction

Motivation. Although the inference-based method, proposed in the previous chapter, produces many *salient* negative statements about almost any given entity, it is restricted by the following two limitations. First, the method is only able to negate *existing predicates*, that were created by KG engineers with positive triples in mind. Salient negative knowledge might require a different set of predicates. For instance, the *Patron Saint of England* (*Saint George*) was not *English*, and in fact, has never been to *England*. Inferring the negative statement ¬ (saint george, Visited, england), in Wikidata, however, is not possible, due to the absence of the predicate Visited. Second, the peer-based inference method uses statistical conclusions to infer whether a negative statement is indeed truly negative, which may be wrong. Stronger signals for the correctness of the negative statements, such as textual evidence, are needed. We propose a new method that has both flexibility (semantic freedom) towards unseen predicates, as well as explicit evidence for the negativity of a statement, for higher correctness.

Preliminary Work. A crucial choice in textual Information Extraction (IE) is the text corpus. Beyond general topical coverage, typical design decisions are whether to opt for larger, typically noisier text collections, or whether to focus efforts on smaller quality corpora with less redundancy. We first consider large text sources like newspapers, blogs, and encyclopedias. Sentences with a negative meaning in newspapers and blogs, from the STICS [HMW14] corpus, were mostly non-salient, including things that people did, or did not, say, e.g., "*Brad Pitt did not threaten Angelina Jolie With Cash Fine*", political opinions, e.g., "*Angela Merkel never made much of an effort to ensure that eastern Germans felt a sense of belonging*". Encyclopedias, on the other hand, focused on positive-only statements. The small set of sentences with negative keywords contained double negation keywords in the same sentence or temporary negatives, such as "*Hawking was not initially successful academically*" and "*His family could not afford the school fees without the financial aid of a scholarship*". Overall, none of these sources contain short trivia sentences with negative keywords. Therefore, we opt for for a small source of particularly high-quality trivia data: search engine query logs, to which limited access can be obtained via auto-completion APIs [RRP$^+$19].

Approach. We present a new method based on textual extractions, i.e., the *pattern-based query log extraction* method. In this approach, we combine open information extraction with a dictionary of negation keywords. We then apply this approach to search engine query logs, for direct access to relevant statements. In a nutshell, we create a few handcrafted *meta-patterns*. Next, we instantiate these patterns with entity mentions, which we submit to prominent search engines with accessible auto-completion APIs, in order to retrieve textual occurrences of salient negative statements.

Research Problem. Given an entity `e` in encyclopedic open KG K^a and a text corpus C, compile a ranked list of salient negative statements about `e`.

4.2 Pattern-based Query Log Extraction

The pattern-based query log extraction method derives salient negative statements about entities, by accessing frequently asked questions, where these questions must contain: (1) the word *why* to indicate that what comes after it is *true* and the user is merely asking for an explanation; (2) a negative keyword to indicate that the information included expresses a negative statement. For instance, for the pattern *why didn't* `e` *..*, with `e` = `stephen hawking`, top returned statements include *why didn't* `stephen hawking` *win a nobel prize in physics*, *receive a knighthood*, *believe in god*, etc.

Negative word	(%)
-n't	55.87
not	23.74
no	13.95
never	3.19

Table 4.1: Negation-bearing words in text corpora [BM11].

4.2.1 Meta Patterns

Inspired by the work on identifying negated findings and diseases in medical discharge summaries [CBCB01], we manually craft 9 meta-patterns to retrieve negative statements in query logs. All our meta-patterns start with the question word *"Why"*, because questions of this kind implicate that the questioner knows, or believes, the statement to be true, but wonders about its cause. We combine this question word with four kinds of negation, *n't, not, no and never*, which according to Blanco [BM11] (using The Penn Treebank) cover 97% of the explicit negation markers (see Table 4.1). Together with two tenses and two verb forms (have and do), these gave rise to a total of 9 frequent meta-patterns, listed in Table 4.2.

4.2.2 Query Log Extraction

An overview of the method is illustrated in Figure 4.1. Given an input entity `e` from a KG, and a list of *meta-patterns*, we proceed as follows:

1. **Instantiate meta-pattern:** We replace the placeholder in each of the meta-patterns with our input entity, e.g., *Why* `Brad Pitt` *never*.

2. **Query auto-completion API:** We submit each of the constructed questions to the search

Meta-pattern	Instantiation	Sample Answer
Why isn't <e>	Why isn't Iceland	in the eu
Why didn't <e>	Why didn't Stephen Hawking	get a nobel prize
Why doesn't <e>	Why doesn't Amazon	accept paypal
Why <e> never	Why Tom Cruise never	won an oscar
Why hasn't <e>	Why hasn't Joe Biden	gone to east palestine
Why hadn't <e>	Why hadn't Russia	granted ukraine its call for independence
Why <e> has no	Why Germany has no	cricket team
Why wasn't <e>	Why wasn't Sylvester Stallone	in creed 3
Why <e> had no	Why Iraq had no	weapons of mass destruction

Table 4.2: Our chosen meta-patterns to retrieve negative statements from query logs.

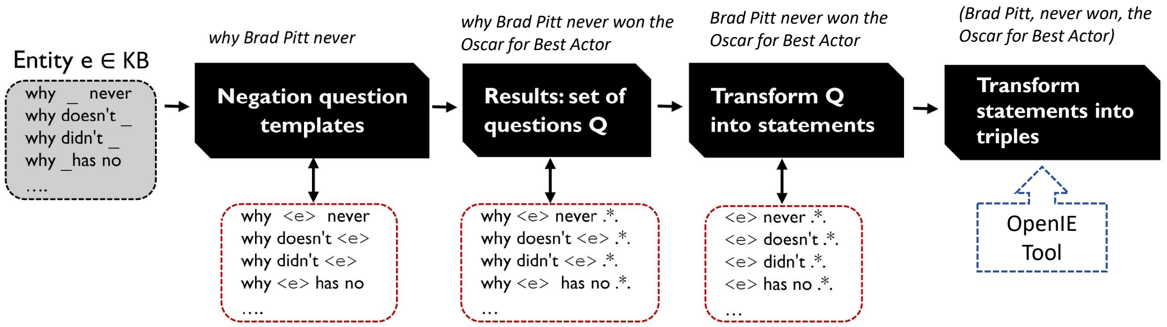


Figure 4.1: Retrieving negative statements about `brad pitt` from a search engine's query log.

engine's auto-completion API, and collect all returned answers, e.g., *Why* `Brad Pitt` *never won the oscar for best actor*.

3. **Convert the question into an assertion**: Using simple heuristics, such as dropping the question word *why* and swapping the positions of the *subject* and *verb*, we convert the returned question into a statement, e.g., `Brad Pitt` *never won the oscar for best actor*.

4. **Convert the statement into a KG triple:** Using OpenIE tools, such as ClausIE [CG13], we convert the statement into a triple with a canonicalized `subject`, i.e., our input entity, and predicate and object phrases.

Note on ranking: the returned result sets are small (54 statements for `Brad Pitt`), especially compared to our previous inference-based method (1500 statements for the same entity). We do, however, use a simple statement frequency across auto-completion APIs as a measure of salience. For instance, each of the meta-patterns is submitted to two prominent search engines, namely Bing and Google, where duplicate or near-duplicate results are merged together in the final set.

4.3 Evaluation

Correctness. For 50 entities, using the peer-based inference and the pattern-based extraction methods, we collect the top negative statement for each, and annotate as "Correct, Incorrect, or Ambiguous (opinion, difficult to find evidence, lacks contexts, ..)". Table 4.3 shows that the pattern-based extraction methods has 2% more correct and 34% less incorrect negatives

Correctness	Peer-based Inference (%)	Pattern-based Extraction (%)
Correct	52	54
Incorrect	48	14
Ambiguous	0	32

Table 4.3: Correctness evaluation of KG-inference-based and query-log-text-based methods.

than the peer-based inference method. Statistical inferences generally only produce statistical conclusions, while textual evidence is, in most cases, a stronger signal that a negative statement is truly negative. On the other hand, the statements produced by the inference-based method are interpretable, i.e., their truthfulness is easily judge-able, because of well-defined `predicates` and `objects`, unlike the extraction-based statements, with 32% marked as ambiguous. A few examples are shown in Table 4.4. For instance, it is not clear which award `salman khan` did not accept, and the last statement is clearly the questioner's opinion on `theresa may`.

Statement	Method	Correctness
(`george washington`, didn't live, in the white house)	extraction	Correct
`¬(tim berners-lee, CitizenOf, u.s.a.)`	inference	Correct
(`bob dylan`, didn't accept, his nobel prize)	extraction	Incorrect
`¬∃o:(Karim Benzema, League, o)`	inference	Incorrect
(`salman khan`, didn't accept, award)	extraction	Ambiguous
(`theresa may`, has no, shame)	extraction	Ambiguous

Table 4.4: Sample correctness annotations.

Salience. We randomly sample 100 entities, and retrieve the top-3 negatives about them, using peer-based inference and pattern-based extraction methods. We submit the negative statements to crowd workers and ask them whether *they found each statement interesting enough to add it to a biographic summary about the entity*. For answers, we allow: Yes, Maybe, and No. We request 3 votes per statement, and interpret the answers as numeric scores, namely 1 for Yes, 0.5 for Maybe, and 0 for No. Our final labels are the numeric averages among the three annotations. We find a standard deviation of 0.2, and a full agreement of the three annotators on 29% of the questions. Results show that the pattern-based extraction method outperforms the peer-based inference by 8% on salience, with 77% for the former, and 69% for the latter. One key reason for this advantage is the flexibility the pattern-based method has, especially in terms of predicates. As shown in Table 4.5, for instance, not meeting someone or not visiting a place cannot be expressed using methods that are exclusively dependent on the list of predicates the KG provides. On the other hand, this introduces the major challenge of linking back the extracted predicate and object phrases, i.e., **canonicalization**. For instance, not only do we need to create a new predicate `Attended` for the last triple in Table 4.5, we also need to identify which daughter the statement about `tom cruise` is referring to, by reasoning over a combination of triples with predicates `Child`, `Gender`, and `Spouse`, or by looking for further evidence in a different text corpus. With the peer-based method, inferences on structured data naturally lead to conclusions that can be expressed within the schema of the data.

Resource-reliability. Text extractions are inherently limited by the coverage of the input text. This holds especially for the query logs, where most frequently queried information is mainly limited to prominent entities and trendy topics. In contrast, statistical inferences can assign scores to almost any statement. To prove this, for 2400 entities, we test the ability of the peer-based inference and pattern-based extraction methods to produce negative statements

Entity	Peer-based Inference	Pattern-based Extraction
lyndon johnson	¬(Occupation, lawyer)	(didn't meet, the queen of england)
vladimir putin	¬(BornIn, moscow)	(never visited, pakistan)
barack obama	¬(MemberOf, republican party)	(didn't meet, kim jong un)
tom cruise	¬(Occupation, singer)	(didn't attend, his daughter's wedding)

Table 4.5: Top negative statements about encyclopedic entities, produced using KG-based inferences and text-based extractions.

about them. The former has a subject coverage of 99%, and the latter with only 26%. Subject coverage for text-based methods can be increased by opting for a larger, though noisier, corpora. Moreover, querying auto-complete APIs for all entities in a web-scale KG can be very expensive. For instance, using Bing's API [1], to query responses for our 9 meta-patterns about all 102906267 Wikidata entities will cost more than 3M dollars. On the other hand, the peer-based inference method does not require any payments.

4.4 Related Work

Negation is an important feature of human language [MS12]. 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 [CHV^{+}13, WMM^{+}14].

Medical data and health records are a type of knowledge where negation is well-studied. In [Día13], a supervised system for detecting negation is proposed, based on the annotated BioScope corpus [SVFC08]. In [GC03], the focus is specifically on negation sentences with the word "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 [BP14b], a rule-based approach based on NegEx [CBCB01], and a vocabulary-based approach for prefix detection, are introduced. PreNex [BP14a] also deals with negation prefixes, e.g. *asymptomatic, nonsurgical, anti-inflammatory* . 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.

Textual information extraction (IE) is a standard paradigm for KG construction, with a set of choices for sources, e.g., Wikipedia vs. richer but less formal corpora, and methodologies, e.g., pattern-based vs. OpenIE vs. neural extractors. Common challenges in textual IE comprise noise and sparsity in observations, and canonicalization of entities and predicates. Our goal is to achieve maximal flexibility w.r.t. open predicates, and to overcome sparsity in negative statements in texts. Our proposed text-based method combines pattern-based and open information extraction techniques, and applies them to a particularly rich data source, search engine query logs.

4.5 Conclusion

In this chapter, we presented the pattern-based extraction method, to complement the previous peer-based negation inference method. We showed that while the extraction-based method outperforms the inference-based method on salience and correctness, the inference-based

[1] At the time of our experiments, a 1000 queries cost 4$.

method has the advantage on the triple-canonicalization and the subject coverage aspects.

5 Negation Inference from Commonsense Knowledge Graphs

Contents

5.1 Introduction

Motivation. Commonsense knowledge (CSK) is crucial for robust AI applications such as question-answering and chatbots. The goal of mining this type of information is to enrich machine knowledge with properties about everyday concepts, e.g., *gorilla*, *pancake*, *newspaper*. These statements are acquired, organized and stored in Commonsense Knowledge Graphs (CSKGs), with prominent projects such as ConceptNet [SCH17], WebChild [TMSW14], Quasimodo [RRP+19], ATOMIC [SBA+19], and Ascent [NRW21]. Similar to encyclopedic KGs, these projects are almost exclusively focused on positive statements such as (`gorilla`, `AtLocation`, *forest*) and (`gorilla`, `HasProperty`, *mammal*). This allows QA systems, for instance, to answer "*Where do gorillas live?*". On the other hand, CSKGs hardly capture any negative statements such as "*gorillas are not territorial*" or "*gorillas are not carnivorous*". Due to the OWA, one cannot assume that an absent statement is invalid [FHP+06]; instead, its truth is simply *unknown*. While KG completion [WPKD20, DPW+19, MBBC20] is an active research area, creating an ideal KG that fully represents real-world knowledge is elusive, especially for the case of commonsense assertions [WDRS21]. Therefore, QA over KGs cannot answer "*Are gorillas territorial?*". However, such *uncommon knowledge* has value for robust AI applications, asserting that *gorillas* are *not* territorial, unlike other *apes* (and *monkeys*) like *chimpanzees* or *gibbons*.

A few CSKGs capture a small fraction of negative statements. In ConceptNet [SCH17], a crowdsourced KG, 6 negative relations are represented, namely `NotIsA`, `NotCapableOf`, `NotDesires`, `NotHasA`, `NotHasProperty`, and `NotMadeOf`. Nonetheless, in its latest version, the portion of negative statements is less than 2%, covering a few salient, but many nonsalient, statements, e.g., (junkfood, NotHasProperty, *good for your health*), (orange, NotIsA, *a president*). In the automatically constructed web-based CSKG Quasimodo [RRP+19], 350K negated statements represent about 10% of all statements, but these are dominated by incorrect knowledge, e.g., ¬(dog, *has body part, tail*). A recent method that targets the problem of discovering relevant commonsense negations is *NegatER* [SZK21, SK20]. Given a CSKG and a pre-trained language model (LM), e.g., BERT [DCLT19], in order to strengthen the LM's ability to classify true and false statements, the LM is first fine-tuned using the CSKG's positive statements. In the second step, plausible negation candidates are generated using dense k-nearest-neighbors retrieval, by either replacing the subject or the object with a neighboring phrase. In a final step, the set of plausible candidates is ranked, using the fine-tuned LM, by descending order of *negativeness* (i.e., higher scores are more likely to be negative). Even though *NegatER* compiles lists of thematically-relevant negations, it suffers from the following limitations. First, the taxonomic hierarchy between concept phrases is not considered, e.g., the positive statement (horse, IsA, *expensive pet*) is used to create the candidate negative statement (horse rider, NotIsA, *expensive pet*). Even though horse and horse rider are close in embedding space, they describe concepts of completely different types. A better alternative for horse rider would be other related animals, such as hamster. Moreover, the ranking based on the LM's negativeness prediction is not interpretable and follows no clear trend.

Negation Inference from Encyclopedic v. Commonsense KGs. Unlike in Chapter 3, inference-based methods on well-structured encyclopedic KGs do not carry over to verbose commonsense data. For instance, while Wikidata has IDs for every item, such as object, Ascent contains many sentences expressing the same exact information. Therefore, comparisons using exact matching are insufficient. Moreover, CSKGs lack expressive predicates. Wikidata has more than 10K predicates covering specific relations such as BirthPlace and EyeColor. On the other hand, Ascent contains only 19 broad predicates such as HasProperty and ReceiveAction. For statements restricted to these predicates, the PCA rule does not have the same effect it had in Chapter 3. Therefore, we need to consider different ways to boost the correctness of inferences.

Approach. We present the *UnCommonsense* method for identifying *salient negative statements* about concepts in CSKGs. For a target concept, we first compute a set of comparable concepts, by employing both external structured taxonomies and latent similarity. Among these concepts, we postulate a Local Closed-world Assumption (LCWA) [GRAS17], and consider their positive statements that do not hold for the target concept as candidate negatives. To eliminate false positives, candidates are scrutinized against related statements in the input CSKG using sentence embeddings, and against a pre-trained LM, acting as an external source of latent knowledge. Finally, we quantify the salience of negative statements by statistical scores, and generate top-ranked negatives with provenances, showing why certain negatives are particularly interesting.

5.2 Problem and Design Space

A CSKG is a set of statements, each being a triple (s, p, *o*), where s stands for subject (an everyday concept), p is a pre-defined predicate, and *o* is an object phrase.

Following previous work [CRW20a], we do not distinguish between p and *o*, because for textual commonsense knowledge expressions, these distinctions are often ad-hoc and a crisp definition of relations is difficult. Hence, for the remainder of this chapter, we generalize the above form to (s, *phrase*), where s is a canonicalized subject and *phrase* is a short phrase combining p and *o*.

Definition 5.2.1 (Commonsense Negative Statement)**.** A commonsense negative statement ¬(s, *phrase*) is a statement that is *not* true.

For example, *"elephants are not carnivorous"* is expressed as ¬(elephant, *is carnivore*). One naive approach to produce such negations is to assume full completeness over the CSKG, and consider all non-existing statements as negatives. On top of not being materializable, this approach faces the following challenges. In order to assert a negation, it is not sufficient to check if a candidate negation is not positive, due to the incompleteness of large-scale CSKGs., e.g., in Ascent, the absence of statement (elephant, *has eye*) is clearly due to missing information. In addition, whether constructed using human crowdsourcing [SCH17] or information extraction techniques [NRRW22, RRP$^+$19], CSKGs mainly reflect the "wisdom" of the crowd about everyday concepts. This causes the augmentation of many *subjective* or otherwise nonsalient statements, such as (football, *is boring*). A generated negative must be easily interpreted by a human annotator as true or false. Therefore it is important to clean the candidate space prior to materializing negations. Finally, the explicit materialization of all possible negatives is not necessary for most standard AI applications (e.g., users might confuse tabbouleh as something that requires an oven, but not a printer). In other words, it is better to avoid (obvious) nonsensical negative statements such as ¬(printer, *is baked in oven*).

Research Problem. Given a target concept s in a CSKG, generate a list of *truly negative* and *salient* statements.

5.3 The UnCommonsense Framework

We present *UnCommonsense*, a method for automatically identifying salient negative knowledge about everyday concepts. *UnCommonsense* first retrieves comparable concepts for a target concept s by exploiting embeddings and taxonomic relations between these concepts. Over the positive knowledge about these comparable concepts, a Local Closed-world Assumption (LCWA) [GRAS17] is made. These relevant positives are then considered as potential salient negatives about s. Consequently, these candidates might contain many false negatives and nonfactual statements. This is followed by an inspection step, where we use KG-based and LM-based checks to measure the correctness of candidates. Finally, to measure salience, the remaining candidates are scored using relative frequency. An overview is shown in Figure 5.1.

5.3.1 Identifying Comparable Concepts

To increase the thematic relevance of candidate negatives, we define the parts of the CSKG where the CWA is helpful to assume [GRAS17], i.e., the LCWA. For instance, if the target concept is an animal, negatives should mostly reflect animal-related statements such as "*not carnivorous*" or "*not nocturnal*", instead of "*not beverage*" or "*cannot store data*". Therefore, we need to collect *comparable* concepts [BRN18]. One way for collecting related concepts is by using pre-computed embeddings. For instance, elephant is related to both tiger and lion, due to their proximity in the vector space [WMWG17]. The problem with relying solely on this similarity function is that it does not take into consideration the taxonomic hierarchy of the concepts. For example, trunk, circus, and jungle are also highly related to elephant. Instead, one can consider using large collections of taxonomic relations and collect comparable concepts

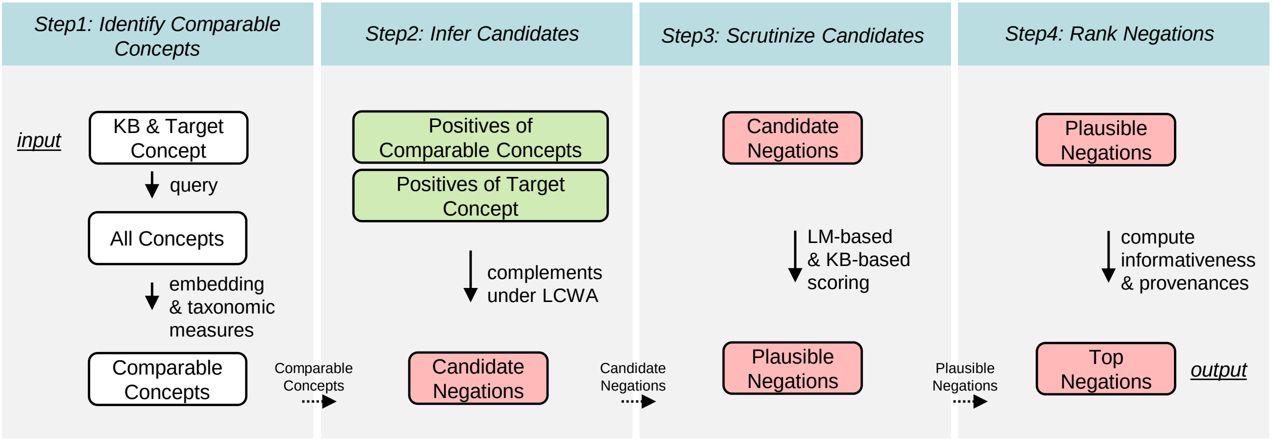


Figure 5.1: Overview of *UnCommonsense*.

only if they are listed as co-hyponyms (e.g., lion and elephant are, trunk and elephant are not). Although this option ensures that related concepts are taxonomic siblings, the group of siblings is unordered. For instance, even though lion and spider are both acceptable taxonomic siblings (under animal), one is clearly more related to elephant than the other. Moreover, large-scale taxonomies are noisy. For instance, using WebIsALOD [HP17], elephant and robot are co-hyponyms under the class toy. We overcome these limitations by combining both techniques and compute *comparable* concepts that are both *semantically and taxonomically highly related*. Given concept s:

1. Using **latent representations** [YAS$^+$20], we compute the cosine similarity score between embeddings of s and every other concept in the KG, then rank them by descending order of similarity.

2. Using **hypernymy relations** [WHZ17], we retain siblings that are co-hyponyms of s. In particular, for every concept, we collect the top-5 hypernyms (ranked by confidence score[1]). For instance, elephant has 843 hypernyms. Top ones include *larger animal*, *land animal*, and *mammal*, and bottom ones include *work of art*, *african*, and *symbol of power*. We retain CSKG concepts as comparable to our target concept if they pass the following taxonomic checks: (i) There exists a common hypernym with the target concept (e.g., both elephant and tiger share *mammal*), and (ii) There does *not* exist an IsA relation in the taxonomy graph with the target concept, e.g., african elephant, IsA, elephant, hence african elephant is not a valid sibling.

The ideal number of comparable concepts to consider for every target concept is a hyperparameter γ, which we tune in our ablation study. For the remainder of the chapter, we use the terms *comparable concepts* and *siblings* interchangeably.

Example 5.3.1. Given s = elephant from CSKG = *Ascent*, and γ = 3, the concepts with the highest cosine similarity are computed using Wikipedia2Vec [YAS$^+$20]. The ranked candidate concepts include tiger, lion, trunk, horse, ... Here, trunk is an obvious intruder as it does not share a hypernym with s. This is determined using WebIsALOD [HP17, HP18], an Is-A database, containing 400m hypernymy relations, mined, using over 50 Hearst-style patterns, from a huge web crawl. We end up with the closest 3 siblings: tiger, lion, and horse.

[1] Using WebIsALOD's SPARQL endpoint: https://webisadb.webdatacommons.org/

5.3.2 Inferring Candidate Negatives

To produce a set of candidate negations, we query from the KG the set of positives about s as well as positives about its siblings. We subtract both sets to produce an initial set of candidates N:

$$N = B \setminus A$$

where B is the set of phrases describing sibling concepts, i.e., each phrase holds for at least one sibling, and A is the set of phrases that hold for the target concept. So, N contains phrases that are $\in B$ but $\notin A$.

Example 5.3.2. Positive statements (phrases) about `elephant`, i.e., A, are: *(is largest land animal)* and *(has tongue)*. Positives of the siblings, i.e., B, are *(is amazing)*, *(can jump)*, *(has tongue)*, *(has hoof)*, *(eat grass)*, *(can leap)*, and *(is big animal)*. The set of negatives N is then all the phrases in the siblings' set, except for *has tongue*, which is a straightforward contradiction with positives about `elephant`.

5.3.3 Scrutinizing Candidates

Plausibility checks. To remove candidates that might be inaccurate due to the CSKG's incompleteness, we measure the plausibility (correctness) of our candidate negatives in two steps:

1. **KG-based scoring**: Unlike encyclopedic KG, e.g., Wikidata [VK14], statements in CSKG are semi-structured. Therefore, it is possible that the same piece of information is expressed in various ways. For example, *lay eggs*, *deposit eggs*, and *lie their eggs* are phrases that hold for different insects in Ascent. Our simple set difference will miss such contradictions. To overcome this issue, we exploit sentence-embeddings [RG19] to capture semantically-close phrases in the CSKG, namely semantically-close information between the concept's and siblings' positives. We filter out candidates that are highly similar to the information we already know about the target concept.

2. **LM-based scoring**: In open-world KGs, it is not sufficient to perform a plausibility check against the knowledge in the KG, as valuable statements might be simply *missing*. We propose consulting an external source for further investigation of candidates. In particular, we probe LMs in, a zero-shot manner, for factual knowledge [RYKW21], by masking the target concept and concatenating the candidate phrase. We then look for a match between predicted tokens and the unmasked concept. We only mask the target concept since it is the most decisive part of a statement.

Example 5.3.3. Using Sentence-BERT, or SBERT [RG19], we measure the similarity (sim) of the candidate and positive phrases:

sim("*can jump*", "*is largest land animal*") = 0.05

sim("*can jump*", "*has hoof*") = 0.20

...

sim("*is big animal*", "*is largest land animal*") = **0.78**

The candidates with similarity greater than or equal to a certain threshold λ (in this example 0.7) are considered *false negatives*. In this case, we drop the candidate ¬(`elephant`, *is big animal*). Next, using BERT [DCLT19], we construct a probe with a masked target concept concatenated with the candidate phrase and look for `s` in the first τ predictions (in this example 100) as follows.

[MASK] has hoof. (*no "elephant" in top-100*)

[MASK] can jump. (*no "elephant" in top-100*)

...

[MASK] eat grass. (***"elephants"* at position 76**)

In this case ¬(`elephant`, *eat grass*) is dropped from the candidate set.

Quality checks. To avoid vague or opinionated negatives such as ¬(`classroom`, *is bigger*) or ¬(`basketball`, *is important*), we identify frequent statements that are highly uninformative. Inspired by the notion of term-weighting in IR [MRS08] (in our case, phrase-weighting), we value phrases of *medium-frequency*, namely ones that are neither too generic nor too rare. While we ensure that rare statements are lower ranked via the pipeline's final step, we tackle too generic statements as follows:

A statement is *generic* if it holds for $\geq \beta$ of the concepts in the KG.

Example 5.3.4. With β = 0.05, ¬(`elephant`, *is amazing*) is dropped from the candidate set as it holds for 16% ($\geq$ 5%) of *all* the concepts in Ascent.

Hyperparameters λ, τ, and β are tuned in Section 5.6.

5.3.4 Quantifying Salience

The output of the previous step is a potentially large set of *truly negative* statements. In fact, beyond our toy example, starting with 30 siblings for `elephant`, *UnCommonsense* produces 1352 initial candidates. Hence, ranking is crucial. We quantify the *importance* of a certain candidate negation by how *uncommon* it is among its siblings. The notion of salience is expressed through unique behavior, characteristic, and so on, of a certain concept, given what is known about its siblings. More formally, given a candidate *phrase* about a target concept `s` and its siblings $\{x_1, x_2, .., x_\gamma\}$, we measure the *phrase*'s salience using *strict sibling frequency*.

$$strict(\texttt{s}, phrase, \{x_1, x_2, .., x_\gamma\}) = \frac{|\{x_i | (x_i, phrase) \in \text{CSKG}\}|}{\gamma}$$

Example 5.3.5. To score candidates, we compute: *strict*(`elephant`, *has hoof*, {`tiger, lion, horse`}) = |{`horse`}|/3 = 0.33, *strict*(`elephant`, *can jump*, {`tiger, lion, horse`}) = |{`tiger, horse`}|/3 = 0.67, and *strict*(`elephant`, *can leap*, {`tiger, lion, horse`}) = |{`lion`}|/3 = 0.33. Therefore it is more noteworthy that elephants cannot jump, unlike 67% of their siblings.

Relaxed scoring. The *strict* salience scoring only handles the cases where candidate negatives are expressed using the same exact phrasing. It cannot, however, capture cases where highly similar candidates are stated using different wording. For instance, the candidate set might contain both ¬(`elephant`, *can jump*) and ¬(`elephant`, *can leap*). To remedy this, we make use of sentence embeddings [RG19] in order to capture this similarity and boost the scores of candidates. We measure the *phrase*'s salience using *relaxed sibling frequency* as follows.

$$\frac{relaxed(\mathtt{s}, phrase, \{(x_1, [phrase'^{1}_{1}, \ldots]), \ldots, (x_\gamma, [phrase'^{1}_{\gamma}, \ldots])\}) = |\{x_i | (x_i, phrase'^{j}_{i}) \in \text{KG} \wedge (phrase = phrase'^{j}_{i} \vee (sim(phrase, phrase'^{j}_{i}) \geq \lambda))\}|}{\gamma}$$

where $phrase'^{j}_{i}$ is a phrase that holds for sibling x_i and $sim(phrase, phrase'^{j}_{i})$ is the semantic similarity between candidate *phrase* and candidate-rephrase $phrase'^{j}_{i}$.

Example 5.3.6. Candidates *(can jump)* and *(can leap)* are semantically similar (0.87 according to [RG19]), hence they are combined under the relaxed scoring. In particular, we compute: *relaxed*(`elephant`, *can jump*, {(`horse`, [*can jump*]), (`lion`, [*can leap*]), (`tiger`, [*can jump*])}) = |{`tiger, lion, horse`}|/3 = 1.0.

Provenance generation. Unlike in previous work on commonsense negation [SZK21], negatives generated by *UnCommonsense* come naturally with an explanation via the relationship between the siblings and the target concept. We call these explanations ***negation provenances***. We generate these human-readable phrases by measuring the in-group frequency of shared hypernyms. In particular, we compute a score for each hypernym *h* that holds for `s` within the set of siblings sharing the same *phrase*.

$$score(h, \mathtt{s}, phrase, \{x_1, x_2, .., x_n\}) = \frac{|\{x_i | (x_i, \text{isA}, h) \in \text{TX}\}|}{n}$$

where TX is the taxonomic relations database, e.g., WebIsALOD [HP17], (x_i, isA, *h*) ∈ TX indicates that hypernym *h* holds for sibling x_i in TX, and *n* is the total number of siblings the candidate-phrase *phrase* holds for.

Example 5.3.7. Assume `elephant` has the hypernyms *wild mammal* and *herbivorous animal*. To build the provenance for top negation ¬(`elephant`, *can jump*) which holds for all siblings (by relaxed scoring), we compute: score(*wild mammal*, `elephant`, *can jump*, {`tiger, horse, lion`}) = |{`tiger, lion`}|/3 = 0.67, and score(*herbivorous animal*, `elephant`, *can jump*, {`tiger, horse, lion`}) = |{`horse`}|/3 = 0.33. The provenance-extended negation then reads: ¬(`elephant`, *can jump*) unlike other *wild mammals*, e.g., `tiger, lion`, and unlike other *herbivorous animals*, e.g., `horse`. To avoid potential multiple appearances of siblings in one provenance, i.e., one sibling belonging to several subgroups, we compute *h* with the highest score iteratively, such that at every iteration we drop already seen siblings.

5.4 Evaluation

Setup. We use Ascent++ [NRRW22] as our input CSKG (in the following just called Ascent). This choice is motivated by the fact that computing negative statements benefits from richer input sets, i.e., high statement-recall per concept. In comparison, in ConceptNet, the most prominent CSKG, has 23 statements per concept on average. Ascent, on the other hand, has 256. Moreover, Ascent contains 2M statements for 23K subjects. We restrict our evaluation to the 8K *primary* subjects, and disregard *aspects* and *subgroups*.

Baselines and Model Variants. In this evaluation, we compare the following baselines, related methods, and variants of our method:

1. **CW-baseline**: In this baseline, the CSKG is simply assumed to be complete, i.e., the closed-world assumption. For a given subject, any phrase not asserted gives an immediate negative statement, e.g., ¬(`duck, CapableOf`, *defend client*).

2. **Text-based Extractions**: We download the latest version of Quasimodo [RRP+19], and retrieve all the statements with *negative polarity* (a total of 350K negatives), e.g., ¬(baby, *has hair*).

3. **LLM-based Generations**: We prompt GPT-3 [RWC+19] (daVinci model), using predefined prompts with negative keywords. Based on Ascent's relations, we define 10 most frequent relations and map to 8 manually-crafted meta patterns "<s> <Negated_NL_relation> ...". <Negated_NL_relation> stands for negated natural language relations we created by rephrasing Ascent's canonicalized predicates, namely "MadeOf" to "is not made of", "CapableOf" to "cannot", "IsA, HasProperty, ReceivesAction" to "is not", "HasA" to "does not have", "AtLocation" to "is not found in", "Causes" to "does not cause", "HasSubevent" to "does not lead to", and "HasPrerequisite" to "does not need". A sample prompt is "butterfly is not a bird". We restrict predictions to a maximum of 6 tokens. We produce 24.4K negations about 200 subjects.

4. **NegatER-$\boldsymbol{\theta}_r$** [SZK21]: This work presents an unsupervised method that ranks out-of-KG potential negatives using a fine-tuned LM. We use the released code[2] to fine-tune BERT on the full Ascent. Similar to the original implementation on ConceptNet, we divide the Ascent dataset into 1.6M/41K/41K rows for training/validation/test, with a total of 715K entity phrases. The evaluation sets are constructed in the same manner, i.e., in terms of balance and negative sampling. We use the given best configuration file and run the fine-tuning step for 3 epochs (6 hours each), using an NVIDIA Quadro RTX 8000 GPU with 48GB of RAM. On the test set, we obtain precision=0.96, and recall=accuracy=0.97. We run the negation generator first in the ranking version *NegatER-θ_r*, which relies on decision thresholds.

5. **NegatER**-∇ [SZK21]: We also run the above negation generator in the ∇ setting, which relies on quantifying "surprisal" using LM's gradients.
   Using both variants of the method, we produce more than 16M scored negations. Note that while we use *canonicalized* Ascent to run *NegatER*, e.g., (elephant, CapableOf, *jump*), for consistency of examples across the methods, we show the *open* version of the triple, e.g., (elephant, *can jump*).

6. **UnCommonsense$^{\mathbf{B}}$**: This baseline variant computes comparable concepts as described in the method, but suspends the scrutinizing and ranking steps.

7. **UnCommonsense$^{\mathbf{S}}$**: The complete method, with salience computed using strict ranking.

8. **UnCommonsense$^{\mathbf{R}}$**: The complete method, with salience computed using relaxed ranking.
   For all variants of our method, γ is set to 30, τ to 50, λ to 0.7, and β to 0.05. These hyperparameters are chosen based on a tuning task in Section 5.6. Moreover, we collect taxonomic siblings from WebIsALOD [HP17] and order them using Wikipedia2Vec [YAS+20]. We use SBERT [RG19] for sentence similarity checks and use BERT [DCLT19] for LM-based checks.

Correctness and Salience. We conduct a crowdsourcing evaluation[3] to determine the quality of each method in generating correct and salient negations. We randomly sample 200 concepts (subjects) and produce for each top-2 negative statements. We then acquire 3 annotations for

[2] https://github.com/tsafavi/NegatER

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

Method	False Negatives	Salience
CWA	0.07	0.07
Text-based Extractions	0.61	0.32
LLM-based Generations	0.63	0.30
NegatER-θ_r	0.27	0.28
NegatER-∇	0.26	0.29
UnCommonsense^{B}	0.29	0.30
UnCommonsense^{S}	0.25	0.50
UnCommonsense^{R}	0.27	0.47

Table 5.1: Correctness and salience evaluation of commonsense negatives.

each negation via crowdsourcing. The total number of annotated negatives is 200 concepts × 2 negatives × 8 methods × 3 annotations = 9.6K rows. We ask every annotator to answer two questions, given a statement about a concept: 1) *Is the following statement truly negative?*, 2) *Is the following statement interesting and/or useful in your opinion?* Since question 1) is a factual question, we only allow "yes" and "no", which we map to 1 and 0 respectively. The Fleiss' kappa [Fle71] inter-annotator agreement is 0.46, i.e. moderate agreement. We interpret this slightly underwhelming agreement on this relatively easy task by a large number of *opinionated statements* produced, especially using the baseline methods, e.g., ¬(football, *is boring*), ¬(muffin, *is delicious*). For question 2), an annotator chooses between "interesting", "slightly interesting", and "not interesting", which we map to 1, 0.5, and 0 respectively. The agreement on this arguably vague task is fair, with Fleiss' kappa inter-annotator agreement 0.30.

Numerical results on correctness and salience are shown in Table 5.1 and qualitative examples in Table 5.2. The *false negatives* column reflects the ratio of results (negative statement) that are in fact positive (i.e., incorrect). Obviously, the *CW-baseline* dominates with only 0.07% false inferences, as the majority of the produced negations are accurate but *nonsensical* (e.g., in Table 5.2, "*rabbits are not related to bribery*"). On the notion of *salience*, the leading method is *UnCommonsense* in its both ranking variants, outperforming the second best external method, *Text-based Extractions*, by 18%, with a slight advantage of the strict ranking variant over the relaxed. We note that in our computation of salience, we only consider the negatives that have been marked by the majority as truly negative. In this case, only 39% of the negatives proposed by *Text-based Extractions* are correct, and even less for *LLM-based Generations*, with 37%, as opposed to 75% for *UnCommonsense*, and 74% for *NegatER*. We observe, for the baseline variant *UnCommonsense*^{B}, that negatives are mostly thematic (due to inferences based on *comparable* concepts), however not frequent enough (due to lack of ranking), e.g., ¬(gorilla, *caught in net*) and in some cases false (due to absence of candidate-scrutiny), e.g., ¬(rabbit, *can feed on seed*). In Table 5.2, *UnCommonsense* shows the most interesting results. For example, it is worth noting that unlike many other small mammals, "*rabbits do not eat insects*".

To give more insights into different kinds of concepts, we show the salience of each method per topic. The results are shown in Table 5.3. *UnCommonsense* performs best on topics like animal and food with salience scores of 67% and 55% respectively. This is expected as both themes contain the most factual statements, and are fairly easy to judge e.g., ¬(banana, *is bitter*) and ¬(horse, *eat fruit*). On the other hand, it is more challenging to generate salient society-related negatives, e.g., ¬(niece, *is pregnant*) and ¬(alcoholic, *has friend*).

Recall. To measure recall, we collect the top 200 negatives, per target concept, produced by each method. Moreover, we need a ground-truth dataset with negative statements about CSKG concepts. We create the **ConceptNet-neg** benchmark, by retrieving all the statements

Method	Top negative statements	Truly negative?
CW-baseline	¬(`acne`, *can give an understanding of truth*)	✓
	¬(`elephant`, *can provide clinician*)	✓
	¬(`yawning`, *has fluid*)	✓
	¬(`vinegar`, *can comprise about 55% nickel*)	✓
	¬(`rabbit`, *related to bribery*)	✓
Text-based Extractions	¬(`acne`, *is natural*)	✗
	¬(`elephant`, *quit smoking*)	✓
	¬(`yawning`, *can end*)	✗
	¬(`vinegar`, *is vegan*)	✗
	¬(`rabbit`, *is rodent*)	✓
LLM-based Generations	¬(`acne`, *can be cured*)	✓
	¬(`elephant`, *found in the dictionary*)	✗
	¬(`yawning`, *can be controlled*)	✗
	¬(`vinegar`, *need to be refrigerated*)	✓
	¬(`rabbit`, *found in the wild*)	✗
NegatER	¬(`acne`, *become unresponsive*)	?
	¬(`elephant`, *interested*)	?
	¬(`yawning`, *attenuated by atropine*)	✓
	¬(`vinegar`, *stocked with herb*)	✓
	¬(`rabbit`, *is the most important animal*)	?
UnCommonsense	¬(`acne`, *is fatal*)	✓
	¬(`elephant`, *is carnivore*)	✓
	¬(`yawning`, *can relax muscles*)	✓
	¬(`vinegar`, *has iron*)	✓
	¬(`rabbit`, *eat insect*)	✓

Table 5.2: Sample negatives about everyday concepts.

Method	Animal	Food	Activity	Social	Object	Other
CW-baseline	0.06	0.09	0.21	0.18	0.10	0.15
Text-based Extractions	0.41	0.44	0.24	0.39	0.20	0.24
LLM-based Generations	0.14	0.46	0.44	0.17	0.22	0.23
NegatER-θ_r	0.10	0.11	0.16	0.26	0.15	0.17
NegatER-∇	0.13	0.14	0.17	0.23	0.12	0.18
UnCommonsense^B	0.29	0.37	0.32	0.24	0.24	0.27
UnCommonsense^S	0.67	0.52	0.49	0.39	0.42	0.45
UnCommonsense^R	0.61	0.55	0.42	0.35	0.41	0.42
Sample concept	*lynx*	*waffle*	*basketball*	*wedding*	*tripod*	*propaganda*

Table 5.3: Salience per domain (topics) of commonsense concepts.

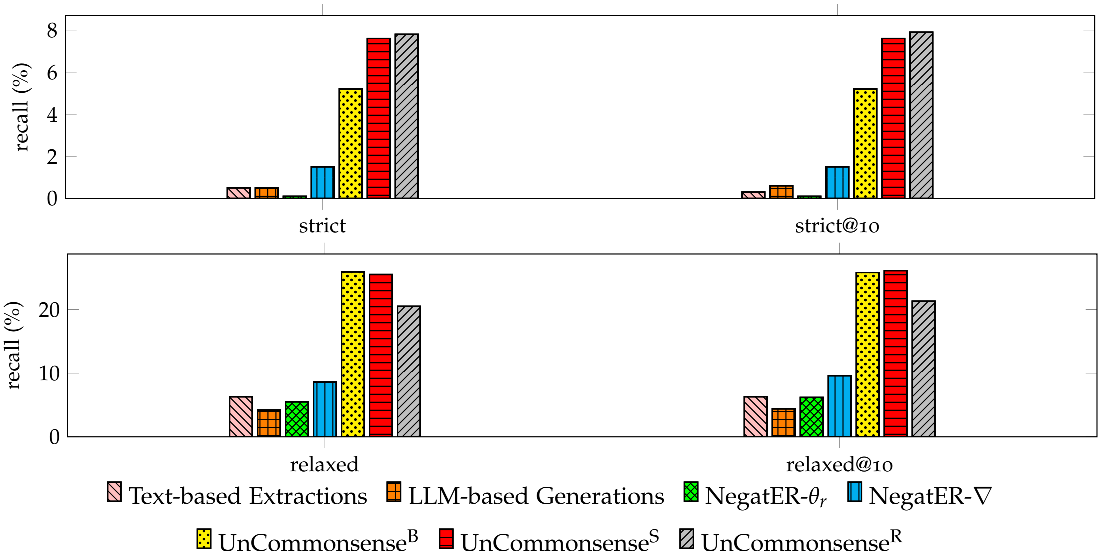


Figure 5.2: Recall evaluation using the ConceptNet-neg [SH12] benchmark.

from ConceptNet [SH12] v5.5 that have a negative relation. This KG allows 6 negative relations such as `NotCapableOf` and `NotDesires`. The dataset contains 14.1K negatives. Samples include (`butterfly`, `NotDesires`, *to sting like a bee*) and (`tortoise`, `NotIsA`, *a turtle*). We remove the negative keywords from relations (i.e., the prefix `Not`). We then compute two modes of recall: In the *strict* mode, we consider a generated negation by a given method to be valid if it matches the *exact phrasing* of a negative statement in the ground truth. In the *relaxed* mode, we use embedding similarity [RG19] to assess whether a generated-negative and a ground-truth-negative are of similar meaning. The recall results are shown in Figure 5.2. *UnCommonsense* outperforms all methods. The strict mode is tougher since the slightest difference between the ground-truth and method-generated negatives is considered a mismatch, e.g., ¬(`air conditioner`, *quiet*) and ¬(`air conditioner`, *quieter*). Relaxing the matching rule to sentence similarity [RG19] allows for more forgiving comparisons. Our method reaches 26.1% in relaxed@10 (relaxed recall at top-10 negatives), followed by *NegatER*-∇ with 9.6%, *Text-based Extractions* with 6.3%, and finally *LLM-based Generations* with 4.4%. An example of a relaxed match is the pair of statements ¬(`bicycle`, *has motor*) (in ground-truth) and ¬(`bicycle`, *has engine*) (generated by *UnCommonsense*).

Provenance Evaluation. To show the effect of extending negative statements with provenances, we conduct a crowdsourcing experiment to compare *UnCommonsense* against *provenance-extended UnCommonsense*. We call the latter *UnCommonsense*^{V}, as in *verbose*. For 200 concepts, for each variant, we produce top-5 negatives. The results are then judged by 3 annotators. We ask about the general salience of the negatives and allow "interesting", "slightly interesting", and "not interesting". *UnCommonsense*^{V} outperforms *UnCommonsense* by 32% in salience, with 81% and 49% respectively. Examples are shown in Table 5.4. The Fleiss' kappa inter-annotator agreement of this task is 0.44, i.e., moderate.

On top of helping to understand the results of our model, generating such explanations help in error analysis, namely in studying failing cases, e.g., whether the nonsalient negative statement made it to the final set due to the noisiness of the taxonomy, low-quality positives from the input CSKG, or other reasons.

Subject	Negation
`muffin`	¬(*is runny*) unlike other *breakfast item*, e.g., *syrup, yogurt*
`gorilla`	¬(*is territorial*) unlike other *wild animal*, e.g., *tiger, lion, monkey, chimpanzee*
`vinegar`	¬(*has iron*) unlike other *ingredient*, e.g., *fennel, celery* and *acidic food*, e.g., *tomato*
`ear`	¬(*is muscular*) unlike other *body part*, e.g., *shoulder, loin, neck*

Table 5.4: Examples of provenance-extended negations (UnCommonsense$^{\text{V}}$).

.

5.5 Use Cases

5.5.1 Negative Trivia

Trivia is an umbrella term for interesting knowledge without a specific purpose. We compare methods for negation generation in their ability to generate *sets* of negative trivia about a concept. We re-use the 200 concepts from before, but now produce top-5 negatives for each, and show them to annotators at once. We compare the best version of our model (*UnCommonsense*$^{\text{S}}$) as the default, and the best of *NegatER* (*NegatER*-∇), as well as *Text-based Extractions* and *LLM-based Generations*. This results in a total of 2.4K annotations (200 concepts × 4 methods × 3 annotations).

For every list of negatives for a given concept, we ask the annotators whether it is interesting, and allow again the same 3 options "interesting", "slightly interesting", and "not interesting". The Fleiss' kappa inter-annotator agreement is 0.24, i.e., fair. *UnCommonsense* leads with 49% salience, followed by *LLM-based Generations* (40%), *NegatER* (30%), and finally *Text-based Extractions* (23%). An example is top negatives about the concept pancake: While *Text-based Extractions* and *LLM-based Generations* are low on plausibility, ¬`(pancake,` *is vegan*`)` and ¬`(pancake,` *is eaten*`)`, respectively, *UnCommonsense* offers the most correct and salient negations e.g., ¬`(pancake,` *is crumbly*`)`.

5.5.2 KG Completion

KG completion refers to the task of identifying novel *positive* statements not yet in a KG. Recent works approach this as an LM-based true/false classification task on candidate statements [SZK21]. A crucial ingredient for this approach is negative examples for training the classifier, and this is where negation generation comes into play. Strong negative examples, i.e., nontrivial ones, can significantly benefit the classifier learning, and in turn, the KG completion accuracy. Following the setup of [SZK21], we compare the impact of negations generated by *UnCommonsense* with that of COMET [BRS+19] and *NegatER*[4].

We use the code by [SZK21] to train a BERT-based KG completion based on each of the three training datasets (100 randomized runs), and report the mean accuracy on the unseen test set. The results are shown in Table 5.5. *UnCommonsense* shows a statistically significant improvement over all methods with $\alpha < 0.01$.

[4] Based on data released at https://github.com/tsafavi/NegatER/tree/master/configs/conceptnet/true-neg/.

Negation Generator	Accuracy (%)
CW-baseline	75.89
COMET	79.06
NegatER	78.61
UnCommonsense	79.56

Table 5.5: KG completion evaluation: LM as a classifier, trained using different negative sampling methods.

Concept = hand, **Query** = What is a hand?
Eliminator = NegatER **A.** foot (-) **B.** feet (-) **C.** digestive organ (-) **D.** body part (-) **E.** help (-)
Eliminator = UnCommonsense **A.** foot (¬ *foot*) **B.** feet (¬ *foot*) **C.** digestive organ (¬ *digestive system*) **D.** body part (-) **E.** help (-)

Table 5.6: Example of MCQA through elimination process (eliminated choice and correct choice).

5.5.3 Multiple-choice Question Answering

Multiple-choice question answering (MCQA) is a common educational and entertainment evaluation setup. Humans approach MCQA often in two ways: (1) Via positive cues on what is the right answer, and (2) Via negative cues that eliminate incorrect answer options, thus narrowing down the set of possible answers. We next investigate to which degree negation generators can help in the second approach. We use the data from the CommonsenseQA task [THLB19]. Examples are shown in Table 5.6. Every question comes with a question concept (i.e., target concept) specifying the topic of the question. For example, the target concept of *"Where can you store a pie?"* is pie. The dataset contains 12K questions, each with only one correct answer. We manually sample 100 questions that: (1) Match concepts in the input CSKG (i.e., Ascent) and (2) Do *not* require any additional condition or information (e.g., *"Where do people read newspapers while riding to work?"*). We translate the questions to a KG-like triple pattern. For instance, *"Where can you store a pie?"* is mapped to `(pie, AtLocation, ?)`. For each question, the eliminator (e.g., *UnCommonsense*) crosses out the answers that *match* a similar negative statement produced for the target concept (similarity is again measured using SBERT with threshold=0.7).

The numerical results are shown in Table 5.7 and examples in Table 5.6. A helpful elimination is a deletion of a *wrong answer* and an unhelpful one is a deletion of a *correct answer*. The *CW-baseline* eliminates most of the options since the absence of the statement is enough to merit a deletion. Besides the *CW-baseline*, the model with the highest number of helpful eliminations is *UnCommonsense* with 108, followed by *NegatER* with 35.

Eliminator	Helpful	Unhelpful
CW-baseline	290 (72.5%)	72 (72.0%)
Text-based Extractions	17 (4.3%)	1 (1.0%)
NegatER	35 (8.8%)	11 (11.0%)
UnCommonsense	108 (27%)	22 (22.0%)

Table 5.7: Answer-eliminations for MCQA task.

Configuration	**False Negatives**	**Salience**
w/o comparable concepts	0.19	0.26
w/o quality checks	0.28	0.22
w/o plausibility checks	0.49	0.38
w/o ranking	0.39	0.29
complete configuration	*0.25*	*0.50*

Table 5.8: Ablation study results.

5.6 Analysis

5.6.1 Ablation Study

In this study, our goal is to show the impact of every component in *UnCommonsense*. For instance, *do the plausibility checks improve the correctness of the inferred negatives?* and *does the ranking improve the salience?* We run our method on 200 subjects and follow the same crowdsourcing setup for 4 different configurations of our method (4 configurations × 200 concepts × 2 negatives × 3 annotators). The Fleiss' kappa inter-annotator agreement of this task is fair on both tasks, namely 0.33 on correctness and 0.26 on salience. The results are shown in Table 5.8. One can see that without comparable concepts (instead random) to derive good thematic candidates from, the salience drops to almost half of the complete configuration (i.e., *UnCommonsense*^{S}). This is different from the *CW-baseline* in Section 5.4 in that we still scrutinize and rank the candidate set. The salience is also highly affected by the suspension of the ranking step (a decrease of 21%). Moreover, holding off the plausibility checks shows an increase of 24% in false negatives.

5.6.2 Hyperparameters Tuning

Our methodology includes four main hyperparameters, namely γ (number of comparable concepts), λ (textual similarity threshold used in scrutinizing candidates and relaxed ranking), τ (the rank threshold for LM), and β (KG threshold for too-generic statements).

We experiment with different ranges of values for these parameters, and set them to their ideal values in Section 5.4 as shown in Figure 5.3, namely γ to 30 (input CSKG=Ascent), λ to 0.7 (similarity measured using SBERT), τ to 50 (LM=BERT), and β to 0.05 (input CSKG=Ascent). We recommend re-tuning these parameters if different CSKGs, LMs, or taxonomies have been used.

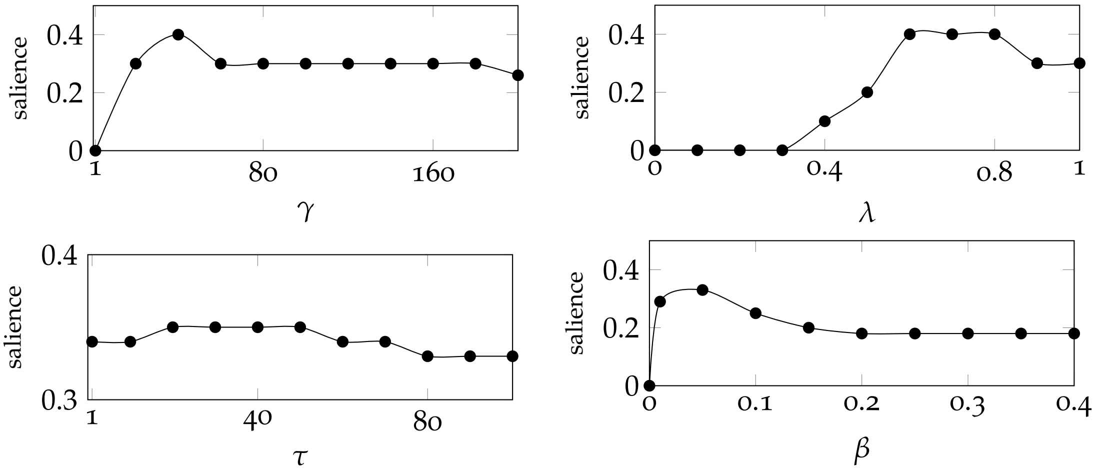


Figure 5.3: Hyperparameters Tuning.

5.7 Related Work

ConceptNet [SCH17] allows the expression of negative statements using 6 pre-defined negative relations. We use these statements in our recall evaluation. The text-extracted Quasimodo [RRP+19] contains 350K negative statements (i.e., with negative *polarity*), yet many have quality issues due to problems with the data source or extraction pipeline. We filter these negative statements from the full KG and use them as a baseline in our experiments (i.e., *Text-based Extractions*). On actively collecting interesting negations, recently, an inference model has been proposed to build a knowledge graph [PVGPW16] with if-then commonsense contradictions [JBBC21]. Unlike our work, [JBBC21] focus on action-based statements and contradictions. For example, *"Wearing a mask is seen as responsible"* and *"Not wearing a mask is seen as carefree"*.

In terms of research problem and goal, the closest work to ours is *NegatER* [SZK21, SK20]. It proposes using LMs to discover meaningful negatives. It fine-tunes the LM for statement truth classification and then uses similarity-based statement corruption to generate candidate negations. In the last step, these are ranked based on proximity to the LM's decision threshold, or a measure of model surprise. As our experiments show, although the methodology is interesting, the taxonomy-unaware corruptions of positive statements are not enough to obtain salient negatives.

Other approaches that target *salient* negations in encyclopedic knowledge graphs, such as Wikidata [VK14] and Yago [SKW07], include statistical inferences (Chapter 3 of this dissertation) and text extractions (Chapter 4 of this dissertation and [KTJ+19]). Yet text extraction is an inherently noisy process, and statistical inference over well-structured encyclopedic data does not carry over to verbose and non-canonicalized textual statements, like in commonsense.

In recent years, Language Models (LMs) have been used to store factual knowledge, learned from pre-training data [PRR+19, SRI+20]. Via LM-probing, one can predict missing tokens in a given claim, e.g. *dogs can [MASK] → walk, run, eat*. In addition, LMs can be trained to derive semantically meaningful sentence embeddings [RG19, GYC21], which helps with the problem of detecting semantic similarity. However, LMs have also been repeatedly shown to struggle with explicit negation [KS20, TEGB20]. We make use of these models in order to scrutinize our candidate negatives and make our rankings stronger via the relaxed sibling frequency. Moreover, we prompt the biggest LM [5], i.e., LLM, GPT-3 [RWC^{+}19], and compare its results with *UnCommonsense*. Our method outperformed the *LLM-based Generations* in all aspects, namely salience, correctness, and recall.

5.8 Conclusion

In this chapter, we presented the *UnCommonsense* framework for compiling salient negative statements about everyday concepts, by exploiting comparable concepts in commonsense knowledge graphs. Our method significantly outperforms baselines and state-of-the-art methods, on both salience and recall.

[5] At the time

Systems and Resources 6

Contents

6.1 System: Wikinegata

Wikinegata (**NEG**ative statements about **Wiki**d**ata** entities) is a platform where users can choose different peering functions to explore the peer-based negation inference methodology, as well as inspect useful negations about Wikidata entities of their choice. The method behind the system is applicable to any other general-purpose KG. The demo is accessible at http://d5demos.mpi-inf.mpg.de/negation, including a demonstrative video on how to use it[1].

6.1.1 Approach Description

This system is built to showcase the *peer-based negation inference methodology* (see Chapter 3). The peer-based method uses the information present on related entities to identify negative statements of interest, for which a Local-Closed World Assumption (LCWA) is assumed. For instance, most persons in Wikidata have no *academic degree* recorded, yet this is often just due to the degree not being important, e.g., for many sports people, artists, or politicians of medium to low fame, and hence, the Open-World Assumption (OWA) applies. We can only make the stronger deduction of negation in more specific cases: Looking at `stephen hawking`, we find that many entities similar to him (e.g., `richard feynman` or `robert oppenheimer`) were `u.s.` citizens, but this information is not mentioned for `hawking`. A conclusion can be made that the LCWA is reasonable to draw for this situation, and hence, that he was truly not a `u.s.` citizen. However, his peers could also share other information, such as that many of them have siblings or many authored literature. To avoid that negative statements of such incidental information come first, the peer-based inference includes, on top of collecting peers and inferring candidate

[1] Video: https://d5demos.mpi-inf.mpg.de/negation/videos/demo.mp4

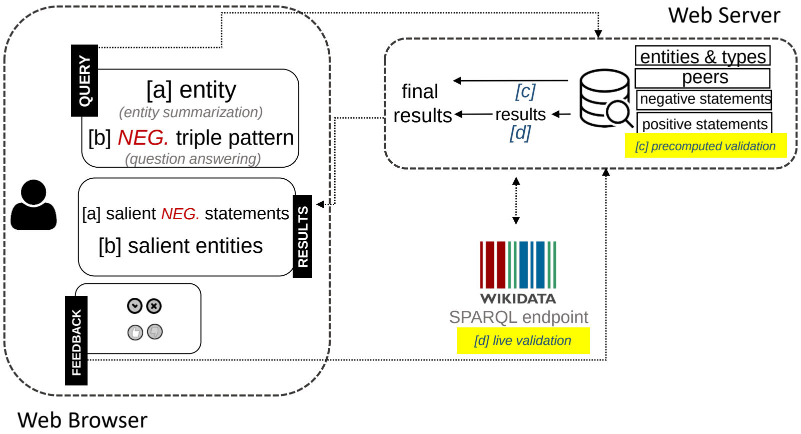


Figure 6.1: Architecture of Wikinegata.

negatives, additional ranking features, such as frequency, unexpectedness, etc. Further details are in Chapter 3.

6.1.2 System Description

Figure 6.1 illustrates the client-server architecture of **Wikinegata**. On the client side, users enter queries that are sent to the server side, where results are retrieved from the database, then displayed for users. The web interface runs on Apache Tomcat. We use HTML, CSS, and Javascript, to build the interface, JSP as the programming language on the server side, and PostgreSQL to create and manage our database. Positive statements are retrieved from Wikidata.

Classes of Negative Statements. Our system is able to produce three classes of negations: (i) grounded negative statements ¬`(s, p, o)`, such as ¬`(stephen hawking, Won, nobel prize in physics)`; (ii) universally negative statements ¬∃`x: (s, p, x)`, such as ¬∃`x: (alan turing, HasChild, x)`; and (iii) conditional negative statements ¬∃`o: (s, p, o), (o, p′, o′)`, such as ¬∃`o: (albert einstein, StudiedAt, o), (o, LocatedIn, u.s.)` (*"Albert Einstein never studied at any U.S. university"*.).

Precomputed Peer-based Negation Inference. As peer-based inference is computationally heavy, yet the validity of inferences is easy to verify live, this step lends itself to an offline precomputation. For this purpose, we implement three orthogonal functions for identifying peers:

(i) structured facets of the subject [BRN18]
(ii) a graph-based similarity measures (e.g., connectivity [PFC17])
(iii) embedding-based similarity (e.g., Wikipedia embeddings [YAS$^+$20])

For 600K popular entities belonging to 11 classes, namely *people* (`Q5`), *books* (`Q571`), *primary schools* (`Q9842`), *films* (`Q11424`), *buildings* (`Q41176`), *organizations* (`Q43229`), *musical groups* (`Q215380`), *businesses* (`Q4830453`), *scientific journals* (`Q5633421`), *literary work* (`Q7725634`), and *countries* (`Q3624078`), we retrieve 100 most similar peer entities, and use these to identify negative statements. The total size of our database, indexed using B-tree indexes, is 64GB, including 681[2] million negative and 100 million positive statements.

Live Validation. Negative statements precomputed offline may turn out incorrect, due to KG completion issues, or real-world changes:

[2] 600,000 entities × (189 negations on average) × 3 similarity functions × 2 negation modes

1. **SPARQL Endpoint:** Until 2016, `leonardo dicaprio` had not won any Oscar, however with his win in that year, in 2023 this assertion is no more true. To address real-world changes, we perform a real-time validation using the Wikidata SPARQL endpoint to check that a *precomputed* statement is not contained in Wikidata at *interaction time*.
2. **User Feedback:** The feedback feature of the platform is storing up and downvotes on the correctness of the displayed negations. If a negation has at least 3 times more downvotes than upvotes (and has at least 10 downvotes), it is then dropped from the result set, i.e., not displayed for future queries.

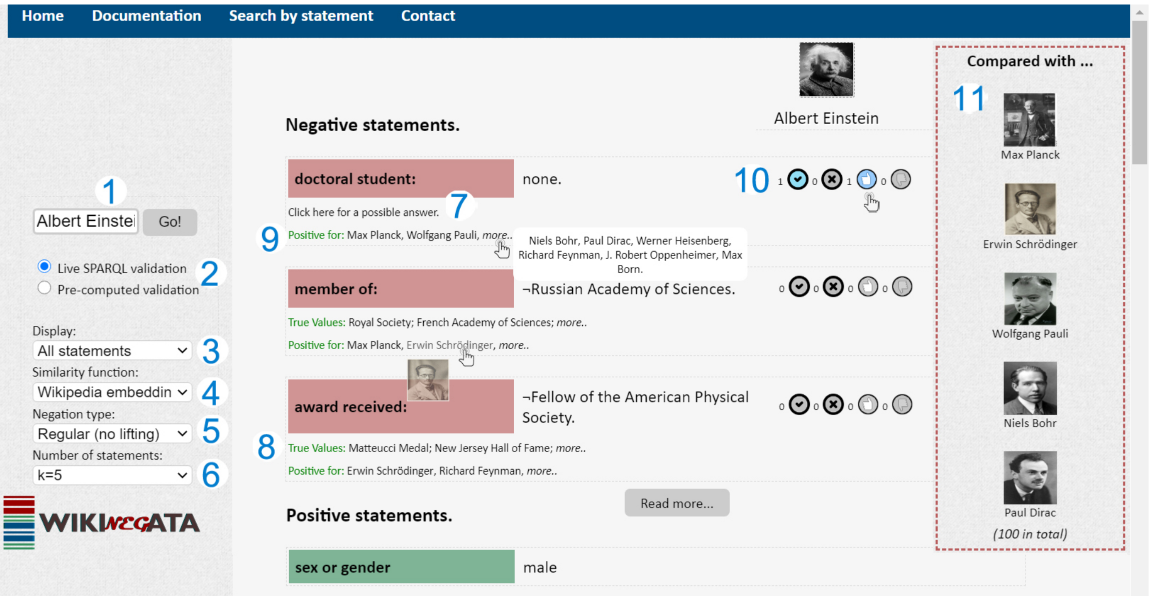
Web Interface. Figure 6.2 shows the platform with results for `albert einstein`. Despite his status as a famous researcher, he never formally supervised any PhD students. And unlike many of his peers, including `max planck`, he was not a member of the `russian academy of sciences`. The platform offers two ways to search negative knowledge:

- **Per-entity Statements:** The platform's main function allows users to discover salient negatives about entities of their choice (see Figure 6.2). The interface has an input entity field **(1)**. One can choose to validate using Wikidata's live SPARQL endpoint or the pre-stored positive information **(2)**. This checks real-world changes at interaction time. Moreover, one can choose whether to display positive and negative or only negative statements **(3)**. The similarity function **(4)** is a choice on *how to collect peers for the input entity*. The negation type **(5)** is a decision on which classes of negation to show (*regular* refers to the grounded and universally negative statements, and *conditional* refers to the conditional negative statements). **(6)** is the number of results to display. **(7)** and **(8)** serve as a glimpse into equivalent positive answers for every negated predicate, by creating a Google query for a possible answer, in the case of universally absent negations **(7)**, and querying Wikidata to show objects that hold for the same predicate, in the case of grounded negations **(8)**. For every result, **(9)** shows the peer entities that the statement *holds* for. Feedback is important to us. One can give signals on correctness and informativeness of results **(10)**. Finally, Under "compared with" **(11)**, the closest peers for the input entity are displayed. By clicking on a peer, a query for that entity is fired. In the unfortunate case where no results are found, a number of alternative queries and features are suggested.
- **Search by Statement:** An additional function allows users to search for entities that share a certain negative statement, such as "*people that did NOT win the Nobel Prize in Physics*" (see Figure 6.5). The interface has an input query field **(1)**. One can choose the similarity function to collect peers **(2)**. Every query can be augmented by further constraints, e.g., the type of entity or other conditions such as "*american citizens who did NOT win the Nobel Prize in Physics*" **(3)**. Top results are displayed in the form of an entity description, image, and the peers for which this statement *holds* for **(4)**. Finally, feedback is allowed to judge the correctness and salience of a given result **(5)**.

The average retrieval time ranges from 4 to 14 seconds. Most of the expensive queries are ones that include many calls to the SPARQL API, especially for the retrieval of conditional statements.

6.1.3 Demonstration Experience

We showcase the **Wikinegata** platform in three scenarios.



Home Documentation Search by statement Contact

Albert Einste Go!

● Live SPARQL validation
○ Pre-computed validation

Display: All statements
Similarity function: Wikipedia embeddin
Negation type: Regular (no lifting)
Number of statements: k=5

Albert Einstein

Negative statements.

doctoral student: none. 1 ✓ 0 ✗ 1 👍 0 👎

Click here for a possible answer.

Positive for: Max Planck, Wolfgang Pauli, *more..*

Niels Bohr, Paul Dirac, Werner Heisenberg, Richard Feynman, J. Robert Oppenheimer, Max Born.

member of: ¬Russian Academy of Sciences. 0 ✓ 0 ✗ 0 👍 0 👎

True Values: Royal Society; French Academy of Sciences; *more..*

Positive for: Max Planck, Erwin Schrödinger, *more..*

award received: ¬Fellow of the American Physical Society. 0 ✓ 0 ✗ 0 👍 0 👎

True Values: Matteucci Medal; New Jersey Hall of Fame; *more..*

Positive for: Erwin Schrödinger, Richard Feynman, *more..*

Read more...

Positive statements.

sex or gender male

Compared with ...

Max Planck
Erwin Schrödinger
Wolfgang Pauli
Niels Bohr
Paul Dirac
(100 in total)

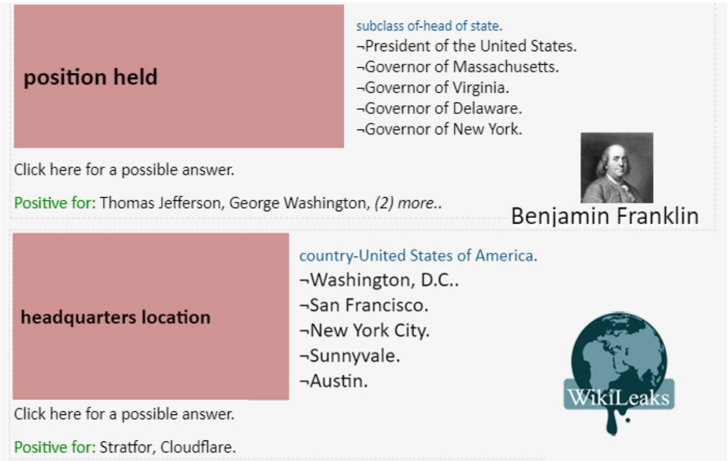
Figure 6.2: The interface for per-entity statements, showing information for `einstein`.

Scenario 1 - Understanding the Peer-based Negation Inference Method. To understand the peer-based inference method, **Wikinegata** offers various levels of introspection. For each entity, peers are shown on the right side of the screen. Moreover, for each inferred negative statement, the set of peers for which it is positive is shown below the statement. For instance, suppose the user enters `steve carell`, the star of the successful comedy show `the office`, and learns that he has *not* won an `emmy award`. She can explore the reason this negation has been inferred and highly ranked by looking at the peers for which this statement holds, i.e., other comedians such as `garry shandling`, as well as other positive values for `carell` for that predicate, i.e., awards such as the `golden globe`, that enabled the partial completeness assumption. Users can actively influence the produced results, too. Suppose a user enters `jeff bezos` as an input entity. She notices that `elon musk` is among his peers when identifying similar entities via Wikipedia embeddings [YAS+20], but not via graph-based measures. This indicates that `bezos` and `musk` share latent information, but have few exact predicate-object combinations in common. Different peer groups then also lead to different deductions, embeddings ranking highest that `musk` is not a writer, graph-based measures ranking highest that he is not a university teacher. More examples are in Table 6.1.

Using the conditional negative statements, one can explore the lifting technique. With one of the founding fathers of the `united states` as the user's input entity, with *conditional* for negation type, she receives the lifted statement that he *never* held a head of state position. Figure 6.3 shows that this technique aggregated 5 grounded negative statements, using one shared relevant aspect.

Scenario 2 - Knowledge Exploration. Interested in negative information about `iceland`, a user enters this country as an input entity and leaves the other fields set to their default values, namely Wikipedia embeddings for peering and regular for negation type. She then starts inspecting the results and was surprised to learn that `iceland` is not a member of the `european union`. She marks this negative statement as informative. Next, she enters `angela merkel` (Figure 6.4). She learns some diverse negative information about her, including that she has no children, unlike many world leaders, is not on `twitter`, and has not studied law.

Scenario 3 - Querying KG with Negated Predicates. The user wants to find prominent people



position held

subclass of-head of state.
¬President of the United States.
¬Governor of Massachusetts.
¬Governor of Virginia.
¬Governor of Delaware.
¬Governor of New York.

Click here for a possible answer.

Positive for: Thomas Jefferson, George Washington, *(2) more..*

Benjamin Franklin

headquarters location

country-United States of America.
¬Washington, D.C..
¬San Francisco.
¬New York City.
¬Sunnyvale.
¬Austin.

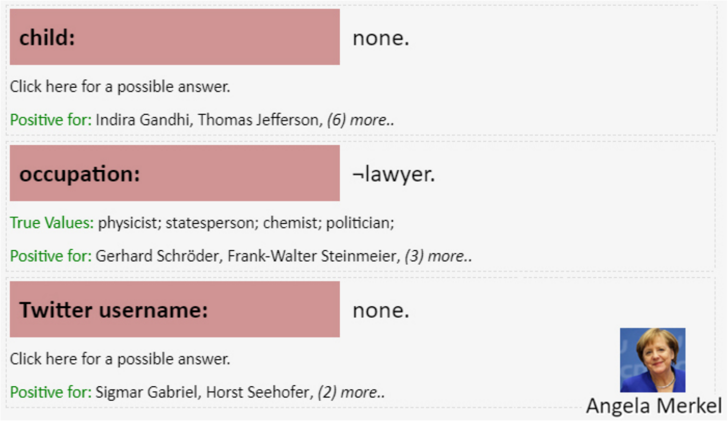
Click here for a possible answer.

Positive for: Stratfor, Cloudflare.

Figure 6.3: Conditional statements for `benjamin franklin` and `wikileaks`.

Entity	Peers	Similarity Function
`winfrey`	`stedman graham`, `barbara walters`, `steve harvey`	Entity embbedings
`winfrey`	`maya angelou`, `ellen deGeneres`, `halle berry`	Graph-based measures
`bezos`	`mark zuckerberg`, `larry page`, `bill gates`	Graph-based measures
`bezos`	`elon musk`, `eric schmidt`, `ginni rometty`	Entity embbedings
`amazon`	`intel`, `adobe`, `microsoft`	Graph-based measures
`amazon`	`best buy`, `walmart`, `ebay`	Entity embbedings

Table 6.1: Peers of `oprah winfrey`, `jeff bezos`, and `amazon`, using different peering functions.



child: none.

Click here for a possible answer.

Positive for: Indira Gandhi, Thomas Jefferson, *(6) more..*

occupation: ¬lawyer.

True Values: physicist; statesperson; chemist; politician;

Positive for: Gerhard Schröder, Frank-Walter Steinmeier, *(3) more..*

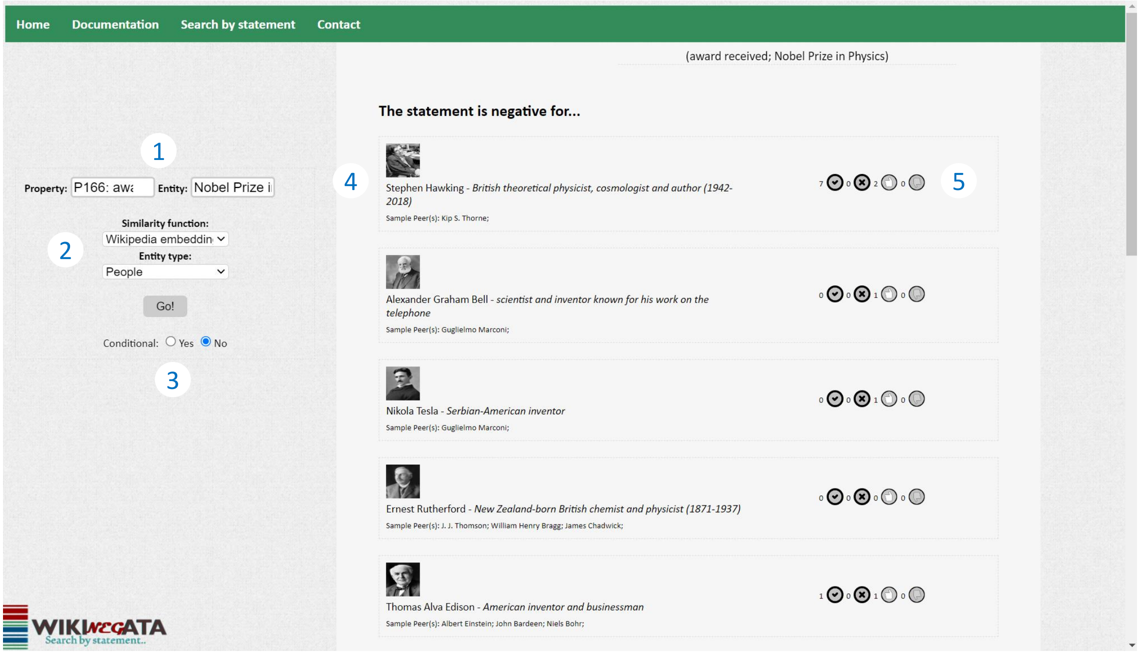
Twitter username: none.

Click here for a possible answer.

Positive for: Sigmar Gabriel, Horst Seehofer, *(2) more..*

Angela Merkel

Figure 6.4: Top negatives about `angela merkel`.



Property: P166: awa Entity: Nobel Prize i

Similarity function: Wikipedia embeddin

Entity type: People

Conditional: ○ Yes ● No

(award received; Nobel Prize in Physics)

The statement is negative for...

Stephen Hawking - *British theoretical physicist, cosmologist and author (1942-2018)* 7 ✓ 0 ☒ 2 ○ 0 ○
Sample Peer(s): Kip S. Thorne;

Alexander Graham Bell - *scientist and inventor known for his work on the telephone* 0 ✓ 0 ☒ 1 ○ 0 ○
Sample Peer(s): Guglielmo Marconi;

Nikola Tesla - *Serbian-American inventor* 0 ✓ 0 ☒ 1 ○ 0 ○
Sample Peer(s): Guglielmo Marconi;

Ernest Rutherford - *New Zealand-born British chemist and physicist (1871-1937)* 0 ✓ 0 ☒ 0 ○ 0 ○
Sample Peer(s): J. J. Thomson; William Henry Bragg; James Chadwick;

Thomas Alva Edison - *American inventor and businessman* 1 ✓ 0 ☒ 1 ○ 0 ○
Sample Peer(s): Albert Einstein; John Bardeen; Niels Bohr;

Figure 6.5: Results for querying *people who did not win a* `nobel prize in physics`.

who *never* received a *nobel prize in physics*, using our search by statement function, shown in Figure 6.5. The figure shows the most salient results, namely 5 of the most famous physicists not to win the award. For instance, two names that stood out are the prominent physicists `stephen hawking` and `ernst rutherford` (a physicist known as the father of nuclear physics who *did not receive the Nobel in physics, but in chemistry*). This is due to the materialization of these negations for such entities. In the absence of the negation inferences, results for such a query, under the CWA, would be random people, including random actors, politicians, and singers.

6.2 System: UnCommonsense

UnCommonsense is a web portal, where researchers can get a better understanding of the *UnCommonsense* method and where general users can browse top negative trivia about concepts of their choice, and query the KG using explicit negated predicates. The method is applicable to any other KG, e.g., ConceptNet [SCH17], but we pick Ascent due to its higher coverage of statements in the ConceptNet schema. The demo is accessible at https://uncommonsense.mpi-inf.mpg.de/.

6.2.1 Approach Description

This system demonstrates the *UnCommonsense* method (see Chapter 5). Given a target concept, e.g., `elephant`, the method computes comparable concepts by employing structured taxonomies and latent similarity measures, e.g., other *wild animals* like `zebra`, `tiger`, `lion`. Among these comparable concepts, the Local Closed-World Assumption (LCWA) is postulated (where *some* parts of the KG are considered complete). Under this, any positive statement that holds for *at least one* of the comparable concepts and *not* the target concept is a candidate negative statement. Restricting the inferences to information about *comparable*, rather than *random* (e.g., `cake, newspaper`), concepts produces much more relevant candidate statements, in this case, animal-related statements such as ¬`(elephant, is carnivore)`. Nonetheless, due to the incompleteness of large-scale CSKGs, inferred negations might be inaccurate, i.e., missing positives. For instance, ¬`(elephant, has eye)` is a missing statement from Ascent. Moreover, lightly-canonicalized CSKGs might contain multiple phrases indicating the same meaning, e.g., `(elephant, is a big animal)` but `(lion, is a large animal)`. This semantic similarity between information about target concept `elephant` and comparable concept `lion` will be overlooked during the previous inference step. To overcome these issues, we scrutinize the candidates against related statements in the input CSKG using sentence embeddings [RG19] and against a pre-trained language model (LM) as an external source of latent knowledge [LLW$^+$20]. Finally, the potentially large set of candidates is ranked by computing salience using statistical scores, i.e., relative frequency within groups of comparable concepts (or type siblings). For example, while `elephant` *cannot*, 67% of its type siblings can jump.

6.2.2 System Description

The web portal is implemented in Python, using the Django framework[3]. We use nginx[4] as a web server and store our datasets in a PostgreSQL database. The demo is deployed on a Debian virtual machine at the Max Planck Institute for Informatics that has 8GB of RAM and 50GB of storage.

Data and Method Hyperparameters. This demo covers all 8029 *primary* concepts in Ascent. The data follows the established ConceptNet schema i.e., canonicalized concepts and relations (both positive and negative). We initially produce 6.7B negations from assuming CWA, which are reduced to 47.2M negations after LCWA is postulated, and lastly to 6.4M final negatives [5] after the scrutinizing step. We set the hyperparameters to their best-performing values as reported in Chapter 5, namely we set the number of siblings to 30, the nonfactual statement threshold to 0.05, the semantic similarity threshold to 0.7, and the rank threshold of LM to 50.

6.2.3 Demonstration Experience

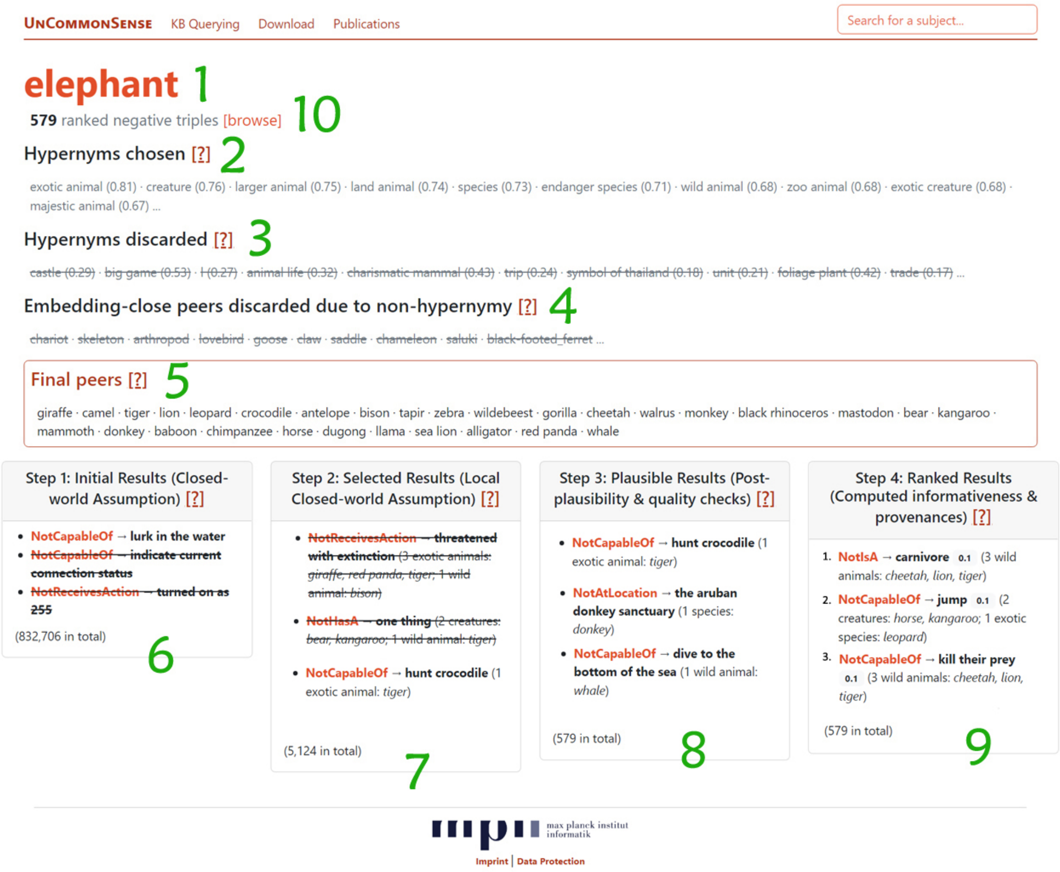
We showcase the **UnCommonsense** platform in three scenarios.

Scenario 1 - Inside UnCommonsense. The main function of **UnCommonsense** allows users to understand the various steps of the methodology (see Figure 6.6). This interface has a target concept field **(1)**, which takes an Ascent primary concept as input (i.e., "search for a subject" auto-completion field at the top-right side). The ranked list of comparable concepts is displayed in **(5)**, e.g., *giraffe*. Users can refer to **(2)**, **(3)**, and **(4)** for a better understanding of their retrieval: high-confidence hypernyms are retained, e.g., *land animal*, while low-confidence or noisier ones are discarded, e.g., *trip*. Moreover, highly related concepts that are not taxonomic siblings and have been discarded are also displayed, e.g. *chariot* is related to *elephant*, but inconsistent by type. To give the user a feel of the full size of the negation sets at every step of the process, we display the total number of results at the bottom of boxes **(6)**, **(7)**, **(8)**, and **(9)**. In the

[3] https://www.djangoproject.com/

[4] https://www.nginx.com/

[5] We release JSON-formatted data dumps at: https://uncommonsense.mpi-inf.mpg.de/download.



UnCommonSense KB Querying Download Publications Search for a subject...

elephant

579 ranked negative triples [browse]

Hypernyms chosen [?]

exotic animal (0.81) · creature (0.76) · larger animal (0.75) · land animal (0.74) · species (0.73) · endanger species (0.71) · wild animal (0.68) · zoo animal (0.68) · exotic creature (0.68) · majestic animal (0.67) ...

Hypernyms discarded [?]

~~castle (0.29)~~ · ~~big game (0.53)~~ · ~~I (0.27)~~ · ~~animal life (0.32)~~ · ~~charismatic mammal (0.43)~~ · ~~trip (0.24)~~ · ~~symbol of thailand (0.18)~~ · ~~unit (0.21)~~ · ~~foliage plant (0.42)~~ · ~~trade (0.17)~~ ...

Embedding-close peers discarded due to non-hypernymy [?]

~~chariot~~ · ~~skeleton~~ · ~~arthropod~~ · ~~lovebird~~ · ~~goose~~ · ~~claw~~ · ~~saddle~~ · ~~chameleon~~ · ~~saluki~~ · ~~black-footed_ferret~~ ...

Final peers [?]

giraffe · camel · tiger · lion · leopard · crocodile · antelope · bison · tapir · zebra · wildebeest · gorilla · cheetah · walrus · monkey · black rhinoceros · mastodon · bear · kangaroo · mammoth · donkey · baboon · chimpanzee · horse · dugong · llama · sea lion · alligator · red panda · whale

Step 1: Initial Results (Closed-world Assumption) [?]

- NotCapableOf → lurk in the water
- ~~NotCapableOf → indicate current connection status~~
- ~~NotReceivesAction → turned on as 255~~

(832,706 in total)

Step 2: Selected Results (Local Closed-world Assumption) [?]

- ~~NotReceivesAction → threatened with extinction (3 exotic animals: *giraffe, red panda, tiger*; 1 wild animal: *bison*)~~
- ~~NotHasA → one thing (2 creatures: *bear, kangaroo*; 1 wild animal: *tiger*)~~
- NotCapableOf → hunt crocodile (1 exotic animal: *tiger*)

(5,124 in total)

Step 3: Plausible Results (Post-plausibility & quality checks) [?]

- NotCapableOf → hunt crocodile (1 exotic animal: *tiger*)
- NotAtLocation → the aruban donkey sanctuary (1 species: *donkey*)
- NotCapableOf → dive to the bottom of the sea (1 wild animal: *whale*)

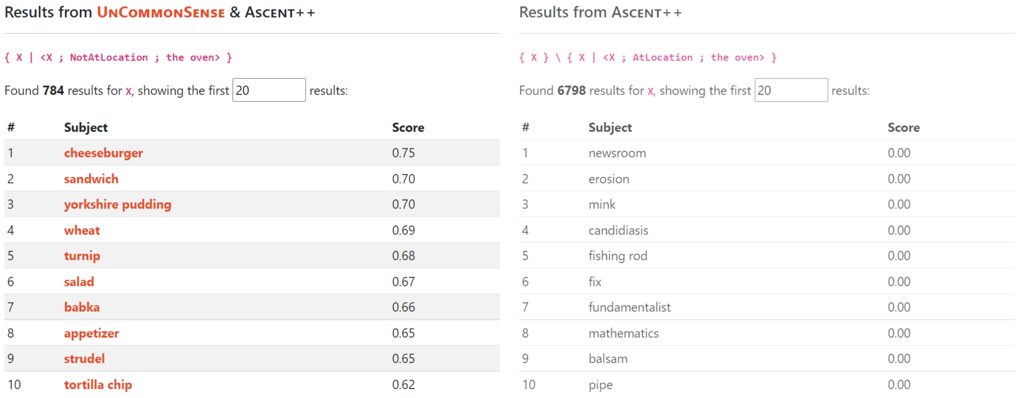
(579 in total)

Step 4: Ranked Results (Computed informativeness & provenances) [?]

1. NotIsA → carnivore 0.1 (3 wild animals: *cheetah, lion, tiger*)
2. NotCapableOf → jump 0.1 (2 creatures: *horse, kangaroo*; 1 exotic species: *leopard*)
3. NotCapableOf → kill their prey 0.1 (3 wild animals: *cheetah, lion, tiger*)

(579 in total)

Figure 6.6: A look into how *UnCommonsense* collects salient negatives about `elephant`.



Results from UnCommonSense & Ascent++

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

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

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

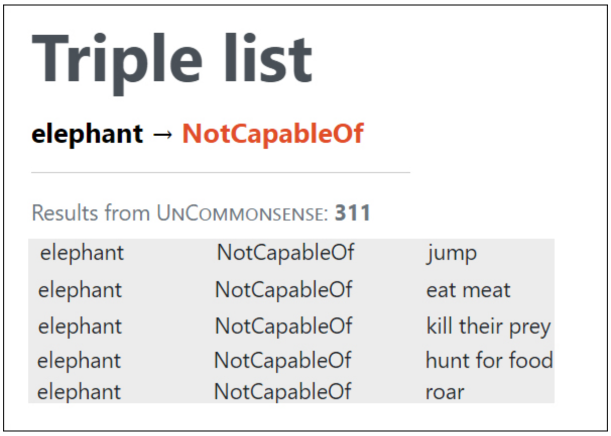
Results from Ascent++

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

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

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

Figure 6.7: Querying for food that *doesn't* require the usage of an oven.



Triple list

elephant → NotCapableOf

Results from UnCommonsense: **311**

elephant	NotCapableOf	jump
elephant	NotCapableOf	eat meat
elephant	NotCapableOf	kill their prey
elephant	NotCapableOf	hunt for food
elephant	NotCapableOf	roar

Figure 6.8: Negative Trivia about `elephant` with predicate `capableOf`.

initial step, see **(6)**, where the naive CWA is postulated, *elephant* received more than 832K candidate statements. This includes an overwhelming number of nonsensical negations, e.g., ¬(`CapableOf`, *indicate current connection status*). The crossed-out negations do not proceed to the next step, see **(7)**, where the LCWA is postulated using the comparable concepts. The number of candidates decreases by 162 times. These statements are thematic, but not yet scrutinized for plausibility and quality. The crossed-out negations here do not make it to the next step, see **(8)**, e.g., ¬(`elephant`, `ReceivesAction`, *threatened with extinction*) contradicts the positive statement in Ascent (`elephant`, `ReceivesAction`, *endangered*)[6] with semantic similarity of 0.72 between the two phrases. Finally, 579 plausible negations are ranked by informativeness, see **(9)**, e.g., ¬(`elephant`, `CapableOf`, *jump*). For more on the ranking metrics, refer to Chapter 5. For users interested in browsing the final negations and not particular steps, they can click on browse, see **(10)**, and will be directed to our next interface.

Scenario 2 - Knowledge Exploration. The user is an elementary school student who is fascinated by the animal kingdom. She has explored many positive statements about them in Ascent [7], namely about their properties and what they are capable of doing. Next, she would like to explore more on things she might *not* be aware of. By querying `elephant` in **UnCommonsense** (see Figure 6.8), she learns that, unlike other *exotic animals*[8] such as `leopard`, *elephants* cannot jump. She also learns that they do not attack prey. This made perfect sense since they also do not eat meat or hunt.

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

A second user is interested in sports that are *not* part of the Olympic games, they translate their information need by entering `<?x NotIsA olympic sport>`. **UnCommonsense** returns

[6] https://ascentpp.mpi-inf.mpg.de/primary-subjects/elephant
[7] https://ascentpp.mpi-inf.mpg.de
[8] Explanations are omitted for readability

96 sports, top ones including `croquet`, `rodeo`, and `kayaking`. On the other hand, positive-only Ascent returns 7320 unranked concepts, including `dialect`, `bread`, and `accountant`. **UnCommonsense** shows that even a simple negative triple-pattern with no positive conjunction or restriction of result-concept type, e.g., `<?x IsA sport.  ?x NotIsA olympic sport>` returns highly relevant concepts, unlike the baseline with mostly off-topic answers. This is especially helpful for users who are not familiar with the wording of object phrases a certain CSKG accepts, e.g., should they augment the query with `<?x IsA sport>`, `<?x IsA game>`, `<?x IsA activity>`, ..

6.3 Resource: Demographics and Outliers in Communities of Interest

We release the first large-scale dataset of demographic information and outliers of communities of interest. Identified from Wikidata, the data covers 7.5K communities, e.g., *members of the White House Coronavirus Task Force*, and 345K subjects, e.g., `deborah birx`. We use components from the peer-based negation inference methodology (Chapter 3) to mine such data.

We release subject-centric and group-centric datasets (Accessible at: https://doi.org/10.5281/zenodo.7410436) in JSON format, as well as a browsing interface (Accessible at https://wikiknowledge.onrender.com/demographics/).

6.3.1 Motivation

A community consists of a group of people who share a commonality such as geography (*Texans*), religion (*Christian*), ethnicity (*Arab*), or a combination (*Arab Texans*). One commonality that is less often discussed is communities of passion or purpose, the so-called *communities of interest* [Fis01]. This refers to groups of people who share a profession, a practice, or an interest. For instance, *members of the White House Coronavirus Task Force* is a community of practitioners in the medical field. Not to be confused with the much broader community of *all* medical practitioners, we focus on contextualized groups of people. In this case, people who were appointed by the *White House* for a specific task.

One standard task for understanding communities is identifying their demographic factors. Demographics are statistical information about a community that includes such factors as gender, occupation, linguistic background, nationality, and location [Ash20]. In geo-based communities, for example, identifying demographics can contribute to local policy-making or to understanding consumer behavior for national businesses. In communities of interest, it could contribute to identifying under-represented groups or to studying cultural differences between similar communities across countries or continents. For instance, compiling top demographics facilitates the task of finding outliers, i.e., members that have different characteristics than the majority, e.g., `deborah birx` is a female while 86% of the `white house coronavirus task force` members are male. These can contribute to studies of under-represented groups in different settings.

6.3.2 Dataset Creation

Identifying Communities. We choose Wikidata as the source of these communities. On top of offering expressive `predicate-object` pairs to construct salient groups of interest, e.g., MemberOf-acm fellows, Wikidata contains additional information about every entity, which allows for mining demographic factors. We pick 6 predicates indicating interest or profession (PositionHeld - P39, Award - P166, ParticipatedIn - P1344, CandidateInElection - P3602, NominatedFor - P4353, MemberOf - P463). We instantiate a SPARQL query [9] with a *predicate of interest* and one of its objects (community title) to collect its members, e.g., select distinct ?subject where {?subject wdt:P166 wd:Q18748039} is used to collect acm fellows. A list of subjects is returned, including thomas henzinger, susan nycum, calvin gotlieb, etc.

The outcome of this step is 7.5K communities of interest covering 16 topics and 345K subjects. Given a community of interest, the Wiki-topic tool [10] is queried for top-3 topics. Note that a community can belong to more than 1 topic, e.g., *Presidents of the Senate of Nigeria* is both related to *Politics* under *History & Society* and to *Africa* under *Geography*.

Defining Demographic Factors. Now that we have the communities of interest with their members and topics, we want to identify their most frequent values, given a set of standard demographic factors [Ash20]. We map each of those to equivalent Wikidata predicates (see Table 6.2). For instance, we identify the nationality of a certain member using predicate P27.

Inferring Demographics and Outliers. At this point, we have all the ingredients to start collecting community demographics and outliers. For every community, e.g., Recipient-acm fellowship:

1. From Wikidata, query values for the predefined demography-predicates, e.g., Gender(thomas henzinger, calvin gotlieb, ..) = male.
2. Compute relative incidence of each factor-value pair, e.g., # male recipients of acm fellowship/# recipients of acm fellowship= 673/839 = 0.80
3. Sort by descending order of relative incidence, e.g., Occupation-computer scientist (0.93), Gender-male (0.80), Nationality-u.s. (0.58), etc.
4. Collect outliers as members with demographic values not matching that of the top-k of the community, e.g., NOT(Gender-male) applies to susan nycum and NOT(Nationality-u.s.) applies to calvin gotlieb.

Accuracy of Inferred Information. When inferring non-asserted factors for certain members, one unavoidable challenge is the *correctness* of these inferences (due to the open-world assumption). Present statements in some cases can also be undetectable using exact-match querying due to potential modeling issues. We remedy these using three heuristics:

(i) **The partial completeness assumption PCA** [GTHS13, DGH$^+$14], which asserts that if a subject has *at least one* object for a given predicate, then there are no other objects beyond those that are in the KG, e.g., if we have at least 1 award for subject X then we assume that their list of awards is complete. (ii) **Hierarchical checks**, where we exploit the type system, i.e., class taxonomy, in search of a contradiction of a certain negated factor. For instance, Occupation-catholic priest does *not* hold for subject X, but Occupation-latin catholic priest does, and (latin catholic priest, SubclassOf, catholic priest) is a statement in Wikidata. Hence, if X is a *latin catholic priest*, they are also a *catholic priest*. (iii) **Semantic similarity checks** to avoid possible synonymous or near-synonymous contradictions, we compute the sentence similarity between a candidate statement and an existing statement for subject X. We do so using SBert [RG19] with 0.6 as a similarity threshold, e.g., similarity (*"teacher"*, *"professor"*) = 0.62, avoiding the inference that someone is a *professor* but not a *teacher* and vice versa.

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

[10] https://wiki-topic.toolforge.org/topic

Factor	Wikidata Predicate
Gender	sex or gender (P21)
Sexual orientation	sexual orientation (P91)
Occupation	occupation (P106)
Political leaning	member of political party (P102)
Religion	religion or worldview (P140)
Linguistic background	native language (P103)
Ethnicity & race	ethnic group (P172)
Nationality	country of citizenship (P27)
Location	residence (P551)

Table 6.2: Standard demographic factors and their KG predicates.

95% of the eliminated candidates are due to PCA, 2% due to hierarchical checks, and due to 3% semantic similarity checks.

6.3.3 Dataset Description

We release two datasets [11] on Zenodo [12] and a browsing interface [13].
Group-centric Dataset. This dataset consists of 7530 rows in the English language, with a total size of 64MB in JSON format.

The fields of a JSON record are:

- Title ID: title of the community using Wikidata IDs.
- Title label: title of the community using equivalent Wikidata labels.
- Number of recorded members: number of subjects in Wikidata that belong to the community.
- Topics: a list of topics describing the community, e.g., `Culture.Media.Music`.
- Demographic factors: a list of top demographics, each consisting of an ID, a label, and a score. The ID describes the factor using Wikidata identifiers, and the label describes it using natural language. The score is the relative incidence within the community.
- Outliers:
  - Reason: a statement on why the following members are considered outliers.
  - Score: a numerical value indicating the frequency of this factor in the community.
  - Members: a list of members for which this factor does not hold.

A sample record [14] from the group-centric dataset:

[11] The data was collected during December 2022.

[12] https://doi.org/10.5281/zenodo.7410436

[13] https://wikiknowledge.onrender.com/demographics/

[14] For readability we omit some of the listed fields.

```
{
        "title": "holders of position Lord Mayor of Dublin",
        "recorded_members": 91,
        "topics": ["Geography.Northern_Europe"],
        "demographics": [
            "gender-male",
            "occupation-politician"
        ],
        "outliers": [
                {
                        "reason": "NOT(male) unlike 81 out of 91 recorded members",
                        "members": [
                            "Catherine Byrne (female)",
                            "Emer Costello (female)",
                            "Alison Gilliland (female)"
                            ]
                },
                {
                        "reason": "NOT(politician) unlike 47 out of 91 recorded
                            members",
                        "members": [
                            "John D'Arcy (businessperson)",
                            "Dermot Lacey (environmentalist)"
                ]
                }
        ]
}
```

Listing 6.1: JSON record from the Group-centric dataset.

Subject-centric Dataset. For a given subject, we merge all outlier statements across different communities and rank them by descending order of incidence.

Moreover, for richer lists, we extend the list of demography-predicates to *all* possible Wikidata predicates.

This subject-centric dataset consists of 345435 rows in the English language, with a total size of 172MB in JSON format.

The fields of a JSON record are:

- Subject ID: the Wikidata ID of the subject.
- Subject label: its equivalent label.
- Statements: a list of salient statements across all communities of interest this individual is a member of:
  - Statement ID: a statement using Wikidata ids.
  - Statement label: a statement using Wikidata labels.
  - Score: relative incidence.

A sample record [15] from the subject-centric dataset:

[15] For readability we omit some of the described fields.

```
{
        "subject": "Serena Williams",
        "statements": [
        {
                        "statement": "NOT(gender-male) but (female) unlike 56 out of 68 recorded winners of L'Equipe Champion of Champions.",
                        "score": 0.82
                },
                {
                        "statement": "NOT(sport-basketball) but (tennis) unlike 4 out of 8 recorded winners of Best Female Athlete ESPY Award.",
                        "score": 0.50
                }
        ]
}
```

Listing 6.2: JSON record from the Subject-centric dataset.

Technical Details. We run our method on a CPU cluster with a total of 5376 CPU cores; Hardware: 42x Dell PowerEdge R6525 server; RAM: 16 GB per core; Pre-processing steps [16] include: identifying communities, querying their topics from the Wiki-topic tool, collecting Wikidata statements about members; Running time of demographics and outliers inference process: 4hr 8min.

6.3.4 Applications

Demographic Data Analysis for Social Sciences. One use case for our data is in academic research in humanities. One standard social problem is *identifying under-represented groups* [ALC+19, Bur21, Cla91]. These groups can be defined using one or more demographic factors, e.g., *ethnicity*, *gender*. Our data can be considered a resource to answer questions such as: is group X under-represented in community/domain Y? or what is the difference in the representation of group X between different domains? A more specific example is shown in Figure 6.9. We show the fraction of *female* award recipients (in Wikidata assigned `gender-female`) in STEM [17], and in political offices holders in different continents. These numbers can support needed initiatives for more representation of certain groups, e.g., programs such as *Women in Tech*. In this example, we used Wiki-topics to specify spatial and topical dimensions, e.g., *STEM.Physics* for Physics awards and an intersection of *Regions.Europe and History_and_Society.Politics_and_government* for political offices in certain geographical regions. Beyond the topics tool, our dataset is not isolated in terms of what we know about communities of interest and their members. Each subject or group can be linked to its Wikidata profile or Wikipedia article, allowing for more customized analyses. For instance, in award-winning or holding public offices, statements are normally associated with temporal data, allowing the user of this dataset to explore progress across time. For instance, the charts in Figure 6.9 can be re-plotted to include time windows, i.e., female award recipients in Physics [1960-1990], Physics [1991-present], and so on.

Our data can be also used in political science research such as *understanding governing*

[16] We did not record the running time of pre-processing steps.
[17] https://en.wikipedia.org/wiki/Wikipedia:AfC_sorting/STEM

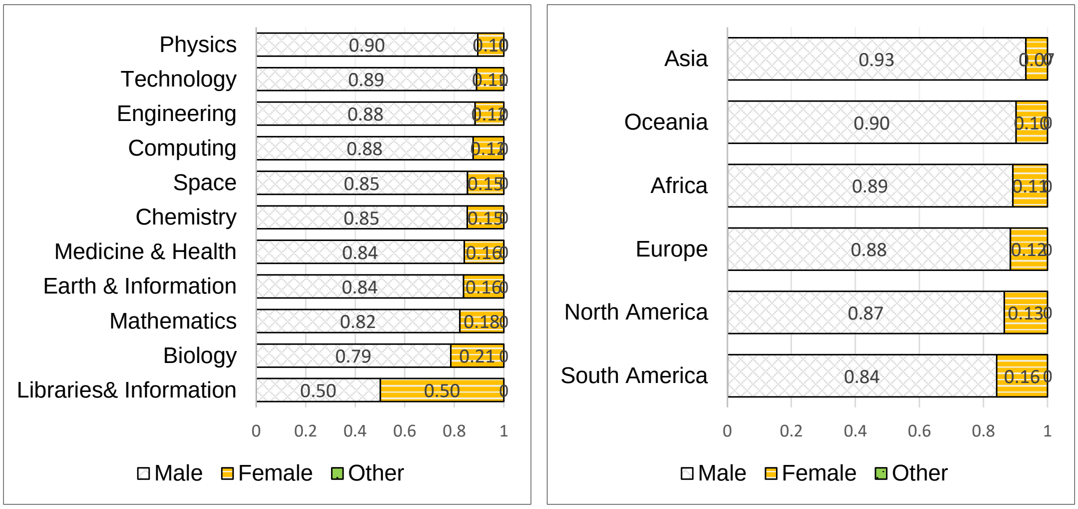


Figure 6.9: Female award recipients in STEM (left) and female political office holders in different continents (right).

in different cultures/parts of the world. One angle is to understand what kind of professions dominate public offices, e.g., *presidents, mayors, governors, ministers*, etc. We compute these as communities of interest created using the predicate `PositionHeld` [18]. We retain communities of interest under *Politics_and_Government*, and assign each to its equivalent geographical area, e.g., *Central America* (see Table 6.3).

Note that we drop the profession at rank 1 since it is *politician* for *all* the geographical areas, due to the fact that holding a certain public office automatically turns one into a politician. This data can give a better understanding of certain cultures or in the case of democracies, how people vote. *Lawyer* is a recurring profession in the Americas, especially in *North America* with quarter of public office holders with `Occupation-lawyer`.

Edit Recommendations for Collaborative Encyclopedias. The Web-scale collaborative KG Wikidata contains more than 100 million items (or subjects) that have received almost 2 billion edits since its inception. Editors often need to prioritize their efforts, so useful tools to guide them can improve data quality and completeness, e.g., the Recoin plugin [BRN18] helps focus the editing on missing predicates of subjects. Our approach and dataset can be used to improve this service by, not only proposing relevant missing predicates, but also proposing a full statement about that predicate. As mentioned, we consider the PCA prior to inferring the negativity of a certain demographic factor. In that step, one cannot assert absent information but can offer a *calculated guess* of what that might be, leaving it for human curators to confirm or deny. For example, `maja vuković` is a member of winners of `ibm fellowship`. For the predicate `Occupation` in Wikidata, she has zero values [19] and Recoin lists `Occupation` as the top missing predicate. Given the demographic data we have about professions of this community she is a member of, we propose `computer scientist`, `mathematician`, and `engineer` as the top 3 candidates. This can especially contribute to the completeness of information about long-tail entities. Moreover, for subjects who are members of multiple communities, one can merge similar demographic values across communities, then average their confidence scores.

[18] https://www.wikidata.org/wiki/Property:P39
[19] https://www.wikidata.org/wiki/Q111536437

Area	Top professions
Central Africa	diplomat (0.27), economist (0.04), civil servant (0.01), philosopher (0.01), minister (0.01)
Eastern Africa	diplomat (0.09), judge (0.03), lawyer (0.03), military personnel (0.03), economist (0.02)
Northern Africa	diplomat (0.12), ruler (0.12), lawyer (0.03), military personnel (0.01), imam (0.01)
Southern Africa	judge (0.28), lawyer (0.11), civil servant (0.01), businessperson (0.01)
Western Africa	diplomat (0.17), lawyer (0.03), military personnel (0.03), economist (0.01), judge (0.01)
Central America	lawyer (0.07), diplomat (0.07), writer (0.02), economist (0.01), military personnel (0.01)
North America	lawyer (0.25), diplomat (0.06), judge (0.03), military personnel (0.01), businessperson (0.01)
South America	lawyer (0.17), diplomat (0.05), military personnel (0.02), journalist (0.01), historian (0.01)
East Asia	monarch (0.09), diplomat (0.07), lawyer (0.06), judge (0.06), prosecutor (0.01)
South Asia	diplomat (0.05), lawyer (0.03), economist (0.02), civil servant (0.02), judge (0.01)
Southeast Asia	sovereign (0.09), judge (0.08), lawyer (0.07), military personnel (0.03), diplomat (0.02)
West Asia	diplomat (0.12), sovereign (0.08), military personnel (0.05), physician (0.02), poet (0.01)
Eastern Europe	diplomat (0.12), economist (0.04), lawyer (0.02), monarch (0.02), university teacher (0.01)
Northern Europe	judge (0.08), diplomat (0.04), monarch (0.02), lawyer (0.02), journalist (0.01)
Southern Europe	diplomat (0.07), lawyer (0.04), military personnel (0.02), jurist (0.01), monarch (0.01)
Western Europe	lawyer (0.13), judge (0.06), diplomat (0.03), military personnel (0.02), suffragist (0.02), teacher (0.01)
Oceania	lawyer (0.08), diplomat (0.04), judge (0.01), pastoralist (0.01), solicitor (0.01), farmer (0.01)

Table 6.3: Top professions in political offices in different parts of the world.

6.4 Conclusion

In this chapter, we presented the **Wikinegata** and **UnCommonsense** web portals to demonstrate the methods proposed in Chapters 3 and 5, to compile lists of negative statements about encyclopedic and commonsense subjects, respectively. Moreover, we release the first large-scale dataset about outliers (negations) in communities of interest, which uses the peer-based inference methods from Chapter 3.

Conclusion 7

Contents

7.1 Summary

In this dissertation, we studied the problem of enriching open-world KGs with salient negative statements, and proposed three methodologies to infer correct and salient lists of negatives about real-world entities, from different sources:

In Chapter 3, we presented the *peer-based negation inference method* to compile lists of interesting negatives from encyclopedic KGs, after formally defining three types of negative statements, namely grounded, universally absent, and conditional. The method assumes the LCWA over relevant subgraphs in the KG, i.e., triples about peer entities. It then infers negatives using positives about the peers, which are later scrutinized using the PCA rule. Finally, the remaining candidates are ranked by salience, using metrics such as peer frequency.

In Chapter 4, we proposed the *pattern-based query log extraction method* to extract salient negative statements about entities from rich textual sources, namely query logs of famous search engines. In this method, we create meta-patterns with negation keywords and retrieve textual occurrences of salient negatives. We then use open information extraction to transform natural language sentences into KG-like triples.

In Chapter 5, we revisited the peer-based inference method in the context of commonsense KGs and propose an adjusted version of the method to address new challenges. The *UnCommonsense* method handles the verbose nature of commonsense data by allowing comparisons using sentence embeddings, instead of the previous exact string matching. Moreover, it replaces the PCA rule for scrutiny, which underperforms due to the lack of expressive predicates, with external sources of knowledge, namely LMs.

In Chapter 6, we finally released resources for future research on the topic, including demo systems for exploring salient negatives and understanding the proposed methods, as well as large-scale datasets produced during these projects.

7.2 Discussion and Future Opportunities

We reflect on the proposed methods, their limitations, and possible future research directions.

7.2.1 Quality Considerations in Encyclopedic KGs

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 this dissertation, we showed that enriching KGs with useful negative statements is beneficial in use cases such as entity summarization. In order to compile a set of likely correct negative statements about an entity, we assumed the closed-world assumption in parts of the KGs, i.e., the local closed-world assumption within peer groups. We strengthen this assumption with the requirement that a negative statement can be inferred only in the presence of another positive statement with the same `subject-predicate` pair (Chapter 3).

Although this approach outperforms other techniques, like embedding-based KG completion, inferences may still be incorrect. In Table 4.3, we saw that text-based methods generate only 14% incorrect negatives, compared to inference-based methods, with 48%. It is therefore advised to show candidate statements from automatic inference to KG curators for final assessment [BRN18]. For instance, for our Wikinegata system (Chapter 6), we chose to show the inferred negative statements about Wikidata entities in a separate interface. We also allow users to give feedback on their correctness and salience, which affects the visibility of down-voted statements. The correctness limitation in the inference-based method was later (Chapter 5) (partially) remedied using an external source of latent knowledge, namely LMs. In Table 5.8, we see that this improves the correctness of inferred statements by 24%.

Real-world Changes and KG Maintenance. Due to real-world changes or new information added to the KG, which happens more frequently in the encyclopedic setting than in commonsense, some of the inferred negatives might become incorrect, i.e., positive. For instance, `leo dicaprio` has won his first `oscar` in 2016. After this year, the negative statement ¬`(leo 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 negative statements can be detected with the help of subsumption checks (`rdfs:subClassOf`). For example, the peer-based negation inference 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, in encyclopedic KGs, 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 KGs at the scale of Wikidata, one could precompute prominent subsumptions, and build these checks into the methodology (e.g., triggering a check for the presence of "`Occupation-writer`" whenever "¬ `Occupation-author`" is inferred).

Modelling and Constraint Enforcement. Some of the inferred negative statements are false negatives due to modeling issues. 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. This redundancy might be remedied using semantic similarity [RG19] measures, which we included in our construction of the communities of interest dataset (Chapter 6).

Some other incorrect negatives could be due to a lack of constraints. For instance, for most businesses, the `HeadquartersLocation` predicate is completed using `cities`, but for `siemens`, in Wikidata, the `building` is added instead (`Palais Ludwig Ferdinand`), making our inferred statement ¬`(siemens, HeadquartersLocation, munich)` a false negative. Although Wikidata encourages editors to use cities for this predicate and advises them to use another predicate for specific buildings, it has not been automatically enforced yet[1].

7.2.2 Long Tail Entities

Compared to the pattern-based extraction method, both versions of the inference-based method have very high subject coverage (26% for the former, 99% for the latter). Nevertheless, our inference-based methods build on the assumption that peer entities are available, for which we have sufficient data. For long-tail entities, both assumptions may be challenged, which can reflect on the quality of inferred negatives. 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 using the class-based similarity function, and hence, our method is not applicable.

On the other hand, low amounts of positive information on peers have little effect on salience. 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 appears only once). This, however, can affect the correctness, especially in the case where no external sources of knowledge are consulted.

7.2.3 Social and Cultural Negatives in Commonsense KGs

In an assessment of the performance of the *UnCommonsense* method over different domains of commonsense concepts (Table 5.3), we notice that the method performs the worst on social concepts, such as `lawyer` or `wedding`.

Upon closer inspection, we interpret this underwhelming performance for two main reasons: (i) the cultural dependency of the quality of inferred social negations. For instance, the statement ¬`(wedding, include wine drinking)` is a true negative in *Arab* cultures but a false negative in *Western* cultures; (ii) the lack of social positive statements in general commonsense KGs. For instance, two positive statements about the concept `wedding` in Ascent are `(wedding, is canceled)` and `(wedding, is held)`.

A recent effort to generate specifically culture-aware commonsense structured knowledge is a better new source to address these limitations [NRVW23]. One future direction is using this cultural commonsense KG with additional textual sources, to mine cultural negations or *no-no* statements. Social behavior/taboos differ between cultures, e.g., countries, continents, and regions. For instance, while it is not legally mandatory to tip at a restaurant in the *U.S.*, there is an unspoken rule that you should. If you do not, you might be chased by a service provider demanding a tip. On the other hand, tipping in *Japan* can be considered insulting. In *Egypt*, haggling is part of the shopping culture. In *Germany*, unless in an open market, it is not. In future work, we aim at mining negated statements about `culture-concept` pairs, e.g., `haggling-germany`, `tipping-japan`.

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

7.2.4 LLMs for Salient Negation Generation.

In a preliminary study [AR23], we examine the ability of advanced LLMs, such as ChatGPT [Ope22], to generate salient negative statements. In particular, we consider 50 entities (25 encyclopedic and 25 commonsense). We infer negations about them using the inference-based methods (Chapter 3 and 5) and text-based methods (Chapter 4). Moreover, we build zero- and few-shot probes, which we submit to ChatGPT [Ope22] and Alpaca-13B (a model fine-tuned from LLaMA on instruction-following demonstrations by Stanford [TGZ$^+$23]).

A sample of the zero-shot probe (LLM=ChatGPT).

> *Write a list of 3 salient factual negated statement about Michael Jordan.*
>
> **Answer:**
>
> 1. *didn't invent Basketball.*
> 2. *didn't only play Basketball.*
> 3. *didn't play in the NBA 2020-2021 season.*

A sample of the few-shot probe (LLM=ChatGPT).

> *A salient factual negated statement about an entity means that the statement doesn't hold in reality. Moreover, the negated statement is either surprising, unexpected, or useful to the reader. For example:*
> *penguins can't fly.*
> *tom cruise never won an oscar.*
>
> On the other hand, the following examples are factual negated statements that are not salient:
> *penguins can't run for presidency.*
> *tom cruise never won the nba best player award.*
>
> *Given this definition and examples, write a list of 3 salient factual negated statement about Michael Jordan.*
>
> **Answer:**
>
> 1. *never played for a team outside the united states.*
> 2. *didn't only play for the Chicago Bulls.*
> 3. *didn't play for the Boston Celtics.*

We then annotate all the negative statements resulting from different methods, for salience and correctness. Results are shown in Table 7.1.

Our evaluation shows that guided probes do in fact improve the quality of generated negatives, compared to the zero-shot variant. We also find that ChatGPT significantly outperforms

Model	**correctness**	**salience**
	overall	
Text Extractions	0.38	0.63
KG Inferences	**0.94**	0.88
ChatGPT *0-shot*	0.71	0.73
ChatGPT *k-shot*	0.76	**0.89**
Alpaca *0-shot*	0.34	0.62
Alpaca *k-shot*	0.50	0.66
	encyclopedic subjects	
Text Extractions	0.32	0.86
KG Inferences	**0.88**	**0.91**
ChatGPT *0-shot*	0.71	0.65
ChatGPT *k-shot*	0.76	0.89
Alpaca *0-shot*	0.32	0.63
Alpaca *k-shot*	0.52	0.69
	commonsense subjects	
Text Extractions	0.47	0.44
KG Inferences	**1.0**	0.83
ChatGPT *0-shot*	0.72	0.81
ChatGPT *k-shot*	0.75	**0.89**
Alpaca *0-shot*	0.36	0.61
Alpaca *k-shot*	0.48	0.63

Table 7.1: Results@1 on correctness and salience of top negative statements.

Alpaca-13B on this task. Nevertheless, using both prompts, both LLMs still struggle with the notion of the factuality of negatives, frequently generating many ambiguous/opinion statements, e.g., *avocados are not bad* or statements with negative keywords but a positive meaning, e.g., *Lebanon is not devoid of historical sites*.

We observe that, on this and other tasks, *designing intuitive prompts* is the most important part of the process. For instance, using the expressions *negative statements*, *negated statements*, or *negation statements* returns completely different responses. For instance, the probe with the word *negated* (alone without *salient factual*) returns obviously true statements with negative keywords added to them, e.g., "*stephen hawking was not a physicist*". The probe with the word *negative* does not return any results, but an apology from the AI about not being able to give *bad statements* about individuals. On this and other tasks, designing intuitive prompts is the most important part of the process. Moreover, real-world changes and maintenance are more of an issue here than in collaborative KGs, which are updated on a daily basis. For LLMs, the process of re-training is much more expensive. e.g., in May 2023, ChatGPT still generates the statement *brendan fraser has never won an oscar*, which is no longer true, due to his win in 2023 (the training of the model has been completed in September 2021).

List of Algorithms

List of Figures

List of Tables

Bibliography

[AADP13] Anastasia Analyti, Grigoris Antoniou, Carlos Viegas Damásio, and Ioannis Pachoulakis. A framework for modular ERDF ontologies. *Annals of Mathematics and Artificial Intelligence*, 2013. Cited on page 11.

[AADW04] Anastasia Analyti, Grigoris Antoniou, Carlos Viegas Damásio, and Gerd Wagner. Negation and negative information in the W3C Resource Description Framework. *Annals of Mathematics, Computing and Teleinformatics*, 2004. Cited on page 11.

[ABK$^+$07] Sören Auer, Chris Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *International Semantic Web Conference*, 2007. Cited on pages 2, 7, and 15.

[AE18] Hiba Arnaout and Shady Elbassuoni. Effective searching of RDF knowledge graphs. *The Journal of Web Semantics*, 2018. Cited on page 8.

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. Cited on page 11.

[ALC$^+$19] Rachel Atkinson, Pamela Lu, Nancy L. Cho, Nelya Melnitchouk, and Lindsay E. Kuo. Gender disparities in award recipients from surgical specialty societies. *Surgery*, 2019. Cited on page 76.

[ANRW23] Hiba Arnaout, Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. Uncommonsense in action! informative negations for commonsense knowledge bases. In *Web Search and Data Mining*, 2023. Cited on pages 4 and 5.

[AR23] Hiba Arnaout and Simon Razniewski. Can large language models generate salient negative statements? *arXiv*, 2023. Cited on pages 5 and 82.

[ARP23] Hiba Arnaout, Simon Razniewski, and Jeff Z. Pan. Wiki-based Communities of Interest: Demographics and Outliers. In *International AAAI Conference on Web and Social Media*, 2023. Cited on page 5.

[ARW20] Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. Enriching knowledge bases with interesting negative statements. In *Conference on Automated Knowledge Base Construction*, 2020. Cited on pages 3 and 4.

[ARWP21a] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Negative knowledge for open-world wikidata. In *The Web Conference*, 2021. Cited on pages 3 and 4.

[ARWP21b] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Negative statements considered useful. *The Journal of Web Semantics*, 2021. Cited on pages 3 and 4.

[ARWP21c] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Wikinegata: A knowledge base with interesting negative statements. *International Conference on Very Large Data Bases*, 2021. Cited on pages 3 and 4.

[ARWP22] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. Uncommonsense: Informative negative knowledge about everyday concepts. In *Conference on Information and Knowledge Management*, 2022. Cited on page 4.

[Ash20] Mohammad Ali Ashraf. Demographic factors, compensation, job satisfaction and organizational commitment in private university: an analysis using sem. *Journal of Global Responsibility*, 2020. Cited on pages 72 and 73.

[BCM$^+$07] Franz Baader, Diego Calvanese, Deborah Mcguinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic handbook*. Cambridge University Press, 2007. Cited on pages 11 and 16.

[BM11] Eduardo Blanco and Dan I. Moldovan. Some issues on detecting negation from text. 2011. Cited on pages 42 and 89.

[BP14a] Ioana Bărbăntan and Rodica Potolea. Exploiting word meaning for negation identification in electronic health records. In *IEEE International Conference on Automation, Quality and Testing*, 2014. Cited on pages 12 and 45.

[BP14b] Ioana Bărbăntan and Rodica Potolea. Towards knowledge extraction from electronic health records - automatic negation identification. In *International Conference on Advancements of Medicine and Health Care through Technology*, 2014. Cited on pages 12 and 45.

[BRN18] Vevake Balaraman, Simon Razniewski, and Werner Nutt. Recoin: Relative completeness in Wikidata. *Wiki Workshop at The Web Conference*, 2018. Cited on pages 17, 19, 21, 49, 64, 77, and 80.

[BRS$^+$19] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In *Annual Meeting of the Association for Computational Linguistics*, 2019. Cited on pages 8 and 58.

[BUGD$^+$13] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Conference on Neural Information Processing Systems*. 2013. Cited on pages 22 and 24.

[Bur21] Cynthia V. Burek. Female medal and fund recipients of the geological society of london: a historical perspective. *Geological Society, London, Special Publications*, 2021. Cited on page 76.

[Car98] Robyn Carston. Informativeness, relevance and scalar implicature. *Pragmatics And Beyond New Series*, 1998. Cited on page 12.

[CBCB01] Wendy W. Chapman, Will Bridewell, Gregory F. Cooper, and Bruce G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of Biomedical Informatics*, 2001. Cited on pages 12, 42, and 45.

[CG13] Luciano Del Corro and Rainer Gemulla. ClausIE: Clause-based open information extraction. In *The Web Conference*, 2013. Cited on page 43.

[CGL$^+$07] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati. Tractable reasoning and efficient query answering in Description Logics: The DL-Lite family. *Journal of Automated Reasoning*, 2007. Cited on page 11.

[CHV$^+$13] Wendy W. Chapman, Dieter Hilert, Sumithra Velupillai, Maria Kvist, Maria Skeppstedt, Brian E. Chapman, Michael Conway, Melissa Tharp, Danielle L. Mowery, and Louise Delegerd. Extending the NegEx lexicon for multiple languages. *Studies in Health technology and Informatics*, 2013. Cited on pages 12 and 45.

[Cla91] Janet Clark. Getting there: Women in political office. *The Annals of the American Academy of Political and Social Science*, 1991. Cited on page 76.

[CRW19] Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Tifi: Taxonomy induction for fictional domains. In *The Web Conference*, 2019. Cited on page 8.

[CRW20a] Yohan Chalier, Simon Razniewski, and Gerhard Weikum. Joint reasoning for multi-faceted commonsense knowledge. In *Conference on Automated Knowledge Base Construction*, 2020. Cited on page 48.

[CRW20b] Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Entyfi: Entity typing in fictional texts. In *Web Search and Data Mining*, 2020. Cited on page 8.

[CRW21] Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. Knowfi: Knowledge extraction from long fictional texts. In *Conference on Automated Knowledge Base Construction*, 2021. Cited on page 8.

[DCCBA08] Marc Denecker, Álvaro Cortés-Calabuig, Maurice Bruynooghes, and Ofer Arieli. Towards a logical reconstruction of a theory for locally closed databases. *Transactions on Database Systems*, 2008. Cited on pages 2 and 9.

[DCLT19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Annual Meeting of the Association for Computational Linguistics*, 2019. Cited on pages 2, 13, 48, 52, and 54.

[DGH$^+$14] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, and Wei Zhang. From data fusion to knowledge fusion. International Conference on Very Large Data Bases, 2014. Cited on page 73.

[DHK$^+$20] Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, and Gabriel Blanco Saldana. Autoknow: Self-driving knowledge collection for products of thousands of types. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020. Cited on pages 1 and 8.

[Día13] Noa P. Cruz Díaz. Detecting negated and uncertain information in biomedical and review texts. In *Recent Advances in Natural Language Processing*, 2013. Cited on pages 12 and 45.

[DPN15] Fariz Darari, Radityo Eko Prasojo, and Werner Nutt. Expressing no-value information in RDF. In *International Semantic Web Conference*, 2015. Cited on page 11.

[DPW$^+$19] Jianfeng Du, Jeff Z. Pan, Sylvia Wang, Kunxun Qi, Yuming Shen, and Yu Deng. Validation of Growing Knowledge Graphs by Abductive Text Evidences. In *AAAI Conference on Artificial Intelligence*, 2019. Cited on page 47.

[DSM17] Dennis Diefenbach, Kamal Singh, and Pierre Maret. WDAqua-coreo: A question answering component for the research community. In *Extended Semantic Web Conference*, 2017. Cited on page 37.

[EGK$^+$14] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, , and Denny Vrandečić. Introducing Wikidata to the linked data web. In *International Semantic Web Conference*, 2014. Cited on page 16.

[EMSW14] Patrick Ernst, Cynthia Meng, Amy Siu, and Gerhard Weikum. Knowlife: a knowledge graph for health and life sciences. In *IEEE International Conference on Data Engineering*, 2014. Cited on page 8.

[ESW15] Patrick Ernst, Amy Siu, and Gerhard Weikum. Knowlife: a versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC Bioinformatics*, 2015. Cited on pages 2, 8, and 10.

[FHP$^+$06] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, and Holger Wache. Inconsistencies, negations and changes in ontologies. In *AAAI Conference on Artificial Intelligence*, 2006. Cited on pages 1, 15, and 47.

[Fis01] Gerhard Fischer. Communities of interest: Learning through the interaction of multiple knowledge systems. In *Information Systems Research Seminar in Scandinavia*, 2001. Cited on page 72.

[Fle71] Joseph L. Fleiss. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, 1971. Cited on page 55.

[GC03] Ilya Goldin and Wendy Chapman. Learning to detect negation with "Not" in medical texts. In *Conference on Research and Development in Information Retrieval*, 2003. Cited on pages 12 and 45.

[GLH$^+$21] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI Open*, 2021. Cited on page 9.

[GRAS17] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek. Predicting completeness in knowledge bases. In *Web Search and Data Mining*, 2017. Cited on pages 12, 38, 48, and 49.

[Gri75] Herbert P. Grice. Logic and conversation. Speech acts, 1975. Cited on page 12.

[GRW20] Shrestha Ghosh, Simon Razniewski, and Gerhard Weikum. Uncovering hidden semantics of set information in knowledge bases. *The Journal of Web Semantics*, 2020. Cited on page 10.

[GTHS13] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In *The Web Conference*, 2013. Cited on pages 3, 16, 18, and 73.

[GTHS15] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. Fast rule mining in ontological knowledge bases with AMIE+. *International Conference on Very Large Data Bases*, 2015. Cited on pages 3, 24, and 38.

[GYC21] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing*, 2021. Cited on page 61.

[HMW14] Johannes Hoffart, Dragan Milchevski, and Gerhard Weikum. STICS: Searching with strings, things, and cats. In *Conference on Research and Development in Information Retrieval*, 2014. Cited on page 41.

[HP17] Sven Hertling and Heiko Paulheim. Webisalod: Providing hypernymy relations extracted from the web as linked open data. In *International Semantic Web Conference*, 2017. Cited on pages 50, 53, and 54.

[HP18] Sven Hertling and Heiko Paulheim. Provision and usage of provenance data in the webisalod knowledge graph. In *International Semantic Web Conference*, 2018. Cited on page 50.

[HSBW13] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial intelligence*, 2013. Cited on page 7.

[HSGE$^+$18] Vinh Thinh Ho, Daria Stepanova, Mohamed H. Gad-Elrab, Evgeny Kharlamov, and Gerhard Weikum. Rule learning from knowledge graphs guided by embedding models. In *International Semantic Web Conference*, 2018. Cited on page 22.

[JBBC21] Liwei Jiang, Antoine Bosselut, Chandra Bhagavatula, and Yejin Choi. "i'm not mad": Commonsense implications of negation and contradiction. *North American Annual Meeting of the Association for Computational Linguistics*, 2021. Cited on pages 8 and 61.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 2002. Cited on page 22.

[JMCC21] Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys*, 2021. Cited on page 9.

[KS20] Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Annual Meeting of the Association for Computational Linguistics*, 2020. Cited on pages 2, 13, and 61.

[KTJ+19] Georgios Karagiannis, Immanuel Trummer, Saehan Jo, Shubham Khandelwal, Xuezhi Wang, and Cong Yu. Mining an "anti-knowledge base" from Wikipedia updates with applications to fact checking and beyond. *International Conference on Very Large Data Bases*, 2019. Cited on pages 2, 12, 13, and 61.

[Len95] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 1995. Cited on page 8.

[LLW+20] Nayeon Lee, Belinda Z. Li, Sinong Wang, Wen tau Yih, Hao Ma, and Madian Khabsa. Language models as fact checkers? In *FEVER workshop at Annual Meeting of the Association for Computational Linguistics*, 2020. Cited on page 69.

[LLY+20] Xusheng Luo, Luxin Liu, Yonghua Yang, Le Bo, Yuanpeng Cao, Jinghang Wu, Qiang Li, Keping Yang, and Kenny Q Zhu. Alicoco: Alibaba e-commerce cognitive concept net. In *ACM SIGMOD International Conference on Management of Data*, 2020. Cited on page 8.

[LS18] Jonathan Lajus and Fabian M. Suchanek. Are all people married? determining obligatory attributes in knowledge bases. In *The Web Conference*, 2018. Cited on page 39.

[MBBC20] Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. Commonsense knowledge base completion with structural and semantic context. In *AAAI Conference on Artificial Intelligence*, 2020. Cited on page 47.

[Mei19] Edgar Meij. Understanding news uing the bloomberg knowledge graph. *The Web Conference*, 2019. Cited on page 8.

[Mot89] Amihai Motro. Integrity= validity+completeness. *Transactions on Database Systems*, 1989. Cited on pages 2 and 9.

[MRN16] Paramita Mirza, Simon Razniewski, and Werner Nutt. Expanding wikidata's parenthood information by 178%, or how to mine relation cardinality information. In *International Semantic Web Conference*, 2016. Cited on page 10.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. Cited on page 52.

[MS12] Roser Morante and Caroline Sporleder. Modality and negation: An introduction to the special issue. *Computational Linguistics*, 2012. Cited on pages 12 and 45.

[MSB+] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Cited on page 12.

[MvH04] Deborah L. McGuinness and Frank van Harmelen. OWL web ontology language overview. *W3C recommendation*, 2004. Cited on page 11.

[NRP16] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *AAAI Conference on Artificial Intelligence*, 2016. Cited on pages 22 and 24.

[NRRW22] Tuan-Phong Nguyen, Simon Razniewski, Julien Romero, and Gerhard Weikum. Refined commonsense knowledge from large-scale web contents. *IEEE Transactions on Knowledge and Data Engineering*, 2022. Cited on pages 2, 8, 49, and 53.

[NRVW23] Tuan-Phong Nguyen, Simon Razniewski, Aparna Varde, and Gerhard Weikum. Extracting cultural commonsense knowledge at scale. In *The Web Conference*, 2023. Cited on page 81.

[NRW21] Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. Advanced semantics for commonsense knowledge extraction. In *The Web Conference*, 2021. Cited on pages 2, 8, and 47.

[ODD06] Eyal Oren, Renaud Delbru, and Stefan Decker. Extending faceted navigation for RDF data. In *International Semantic Web Conference*, 2006. Cited on page 32.

[OEAS$^+$16] Sergio Oramas, Luis Espinosa-Anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. Information extraction for knowledge base construction in the music domain. *Data & Knowledge Engineering*, 2016. Cited on page 8.

[OMP18] Stefano Ortona, Venkata Vamsikrishna Meduri, and Paolo Papotti. RuDiK: Rule discovery in knowledge bases. *International Conference on Very Large Data Bases*, 2018. Cited on pages 12, 38, and 39.

[Ope22] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022. Cited on pages 13 and 82.

[PCE$^+$17] Jeff Z. Pan, Diego Calvanese, Thomas Eiter, Ian Horrocks, Michael Kifer, Fangzhen Lin, and Yuting Zhao. *Reasoning Web: Logical Foundation of Knowledge Graph Construction and Query Answering*. Springer, 2017. Cited on page 16.

[PFC17] Marco Ponza, Paolo Ferragina, and Soumen Chakrabarti. A two-stage framework for computing entity relatedness in Wikipedia. In *Conference on Information and Knowledge Management*, 2017. Cited on pages 17 and 64.

[PRR$^+$19] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *Empirical Methods in Natural Language Processing*, 2019. Cited on pages 13 and 61.

[PVGPW16] Jeff Z. Pan, Guido Vetere, Jose Manuel Gomez-Perez, and Honghan Wu. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer, 2016. Cited on page 61.

[RAGS21] Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, and Fabian M. Suchanek. On the limits of machine knowledge: Completeness, recall and negation in web-scale knowledge bases. *International Conference on Very Large Data Bases*, 2021. Cited on pages 5 and 9.

[RBN17] Simon Razniewski, Vevake Balaraman, and Werner Nutt. Doctoral advisor or medical condition: Towards entity-specific rankings of knowledge base properties. In *Advanced Data Mining and Applications: 13th International Conference*, 2017. Cited on page 21.

[RG19] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Empirical Methods in Natural Language Processing*, 2019. Cited on pages 51, 52, 53, 54, 57, 61, 69, 73, and 80.

[RN11] Simon Razniewski and Werner Nutt. Completeness of queries over incomplete databases. International Conference on Very Large Data Bases, 2011. Cited on page 16.

[RR20] Julien Romero and Simon Razniewski. Inside quasimodo: Exploring construction and usage of commonsense knowledge. In *Conference on Information and Knowledge Management*, 2020. Cited on page 2.

[RRP+19] Julien Romero, Simon Razniewski, Koninika Pal, Jeff Z. Pan, Archit Sakhadeo, and Gerhard Weikum. Commonsense properties from query logs and question answering forums. In *Conference on Information and Knowledge Management*, 2019. Cited on pages 2, 8, 11, 41, 47, 48, 49, 54, and 61.

[RSH+16] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. Yago: a multilingual knowledge base from wikipedia, wordnet, and geonames. In *International Semantic Web Conference*, 2016. Cited on page 7.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020. Cited on page 8.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI technical report*, 2019. Cited on pages 2, 13, 54, and 62.

[RYKW21] Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. Language models as or for knowledge bases. *Workshop on Deep Learning for Knowledge Graphs at the International Semantic Web Conference*, 2021. Cited on page 51.

[SBA+19] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: an atlas of machine commonsense for if-then reasoning. In *AAAI Conference on Artificial Intelligence*, 2019. Cited on pages 8 and 47.

[SCH17] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, 2017. Cited on pages 2, 8, 47, 49, 61, and 68.

[SH12] Robyn Speer and Catherine Havasi. Representing general relational knowledge in ConceptNet 5. In *International Conference on Language Resources and Evaluation*, 2012. Cited on pages 57 and 87.

[Sin12] Amit Singhal. Introducing the knowledge graph: Things, not strings. https://www.blog.google/products/search/introducing-knowledge-graph-things-not, 2012. Cited on pages 1, 7, 8, and 37.

[SK20] Tara Safavi and Danai Koutra. Generating negative commonsense knowledge. In *Conference on Neural Information Processing Systems*, 2020. Cited on pages 2, 13, 48, and 61.

[SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *The Web Conference*, 2007. Cited on pages 1, 7, 15, and 61.

[SRI+20] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Empirical Methods in Natural Language Processing*, 2020. Cited on pages 13 and 61.

[SVFC08] György Szarvas, Veronika Vincze, Richárd Farkas, and János Csirik. The BioScope corpus: Annotation for negation, uncertainty and their scope in biomedical texts. Current Trends in Biomedical Natural Language Processing, 2008. Cited on pages 12 and 45.

[SZK21] Tara Safavi, Jing Zhu, and Danai Koutra. NegatER: Unsupervised Discovery of Negatives in Commonsense Knowledge Bases. In *Empirical Methods in Natural Language Processing*, 2021. Cited on pages 2, 13, 48, 53, 54, 58, and 61.

[TEGB20] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 2020. Cited on pages 13 and 61.

[TGES+20] Trung-Kien Tran, Mohamed H. Gad-Elrab, Daria Stepanova, Evgeny Kharlamov, and Jannik Strötgen. Fast computation of explanations for inconsistency in large-scale knowledge graphs. In *The Web Conference*, 2020. Cited on page 16.

[TGZ+23] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and T. B. Hashimoto. Alpaca: A strong, replicable instruction-following model. https://crfm.stanford.edu/2023/03/13/alpaca.html, 2023. Cited on page 82.

[THLB19] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *North American Annual Meeting of the Association for Computational Linguistics*, 2019. Cited on page 59.

[TMSW14] Niket Tandon, Gerard De Melo, Fabian Suchanek, and Gerhard Weikum. Webchild: Harvesting and organizing commonsense knowledge from the web. In *Web Search and Data Mining*, 2014. Cited on pages 8 and 47.

[TS19] Thomas Pellissier Tanon and Fabian Suchanek. Querying the edit history of Wikidata. In *Extended Semantic Web Conference*, 2019. Cited on page 9.

[TWS20] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. Yago 4: a reason-able knowledge base. In *Extended Semantic Web Conference*, 2020. Cited on page 7.

[VK14] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge base. *Communications of the Association for Computing Machinery*, 2014. Cited on pages 1, 2, 7, 15, 51, and 61.

[WDRS21] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. Machine knowledge: Creation and curation of comprehensive knowledge bases. *Foundations Trends Databases*, 2021. Cited on page 47.

[WHZ17] Chengyu Wang, Xiaofeng He, and Aoying Zhou. A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Empirical Methods in Natural Language Processing*, 2017. Cited on page 50.

[WMM+14] Stephen Wu, Timothy Miller, James Masanz, Matt Coarr, Scott Halgrim, David Carrell, and Cheryl Clark. Negation's not solved: Generalizability versus optimizability in clinical natural language processing. *The Public Library of Science (PLOS One)*, 2014. Cited on pages 12 and 45.

[WMWG17] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: a survey of approaches and applications. *IEEE TKDE*, 2017. Cited on page 49.

[WPKD20] Kemas Wiharja, Jeff Z. Pan, Martin J. Kollingbaum, and Yu Deng. Schema Aware Iterative Knowledge Graph Completion. *The Journal of Web Semantics*, 2020. Cited on page 47.

[WWKY18] Tien-Hsuan Wu, Zhiyong Wu, Ben Kao, and Pengcheng Yin. Towards practical open knowledge base canonicalization. In *Conference on Information and Knowledge Management*, 2018. Cited on page 13.

[YAS+20] Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. Wikipedia2Vec: an optimized tool for learning embeddings of words and entities from wikipedia. *Empirical Methods in Natural Language Processing*, 2020. Cited on pages 17, 21, 24, 29, 30, 35, 50, 54, 64, and 66.

[YBB+16] Mohamed Yahya, Denilson Barbosa, Klaus Berberich, Qiuyue Wang, and Gerhard Weikum. Relationship queries on extended knowledge graphs. In *Web Search and Data Mining*, 2016. Cited on page 12.