UnCommonSense: Informative Negative Knowledge about Everyday Concepts

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

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

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

Jeff Z. Pan
j.z.pan@ed.ac.uk
The University of Edinburgh
Edinburgh, United Kingdom

ABSTRACT

Commonsense knowledge about everyday concepts is an important asset for AI applications, such as question answering and chatbots. Recently, we have seen an increasing interest in the construction of structured commonsense knowledge bases (CSKBs). An important part of human commonsense is about properties that do *not* apply to concepts, yet existing CSKBs only store positive statements. Moreover, since CSKBs operate under the open-world assumption, absent statements are considered to have unknown truth rather than being invalid. This paper presents the UnCommonSense framework for materializing informative negative commonsense statements. Given a target concept, comparable concepts are identified in the CSKB, for which a local closed-world assumption is postulated. This way, positive statements about comparable concepts that are absent for the target concept become seeds for negative statement candidates. The large set of candidates is then scrutinized, pruned and ranked by informativeness. Intrinsic and extrinsic evaluations show that our method significantly outperforms the state-of-the-art. A large dataset of informative negations is released as a resource for future research.

CCS CONCEPTS

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

KEYWORDS

Knowledge Bases, Negation, Commonsense

ACM Reference Format:
Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2022. UnCommonSense: Informative Negative Knowledge about Everyday Concepts. In *Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM '22), Oct. 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557484

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10.
https://doi.org/10.1145/3511808.3557484

1 INTRODUCTION

Motivation. Commonsense knowledge (CSK) is crucial for robust AI applications such as question answering and chatbots. The purpose is to enrich machine knowledge with properties about everyday concepts (e.g., gorilla, pancake, newspaper). Such statements are acquired, organized and stored in structured knowledge bases (KBs) [10, 27, 47]. Large commonsense KBs (CSKBs) include ConceptNet [38], WebChild [43], ATOMIC [36], TransOMCS [50], and Ascent [25]. These projects are almost exclusively focused on positive statements such as *gorillas are mammals, black, and live in forests*, expressed in the form of subject-relation-object triples, e.g., `(gorilla, AtLocation, forest)`. This allows QA systems, for instance, to answer "*Where do gorillas live?*". On the other hand, CSKBs hardly capture any negative statements such as "*gorillas are not territorial*" or "*gorillas are not carnivorous*". By the Open-world Assumption (OWA) underlying most KBs, one cannot assume that an absent statement is invalid [12]; instead, its truth is simply *unknown*. While KB completion [9, 21, 48] is an active research area, creating an ideal KB that fully represents real-world knowledge is elusive, especially for the case of commonsense assertions [47]. Therefore, QA over KBs 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.

State of the art and its limitations. The focus in constructing CSKBs has been on positive statements; only very few projects capture a small fraction of negative statements. In ConceptNet [38], a crowdsourced KB, 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%. Moreover, many statements are all but informative, such as `(envelope, NotHasProperty, alive)`. In the automatically constructed, Web-based CSKB Quasimodo [33], 350k negated statements represent about 10% of all statements, but these are dominated by uninformative knowledge, e.g., `¬(elephant, can, quit smoking)`. A recent method that targets the problem of discovering relevant commonsense negations is NegatER [34, 35]. Given a CSKB and a pre-trained language model (LM), e.g., BERT [8], in order to strengthen the LM's ability to classify true and false statements, the LM is first fine-tuned using the CSKB statements. In a 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 are 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 several limitations: (i) The taxonomic hierarchy between concept phrases is not considered. For instance, from the positive statement `(horse, IsA, expensive pet)`, a semantically sensible corruption of the subject is `hamster`, but `horse riding` or `horserider` are not. Even though they are closer in embedding space, they describe concepts of completely different types (activity, artifact) that cannot be pets. (ii) The method relies on the input CSKB having well-defined relations (e.g., `CapableOf`). This causes issues when triples are merely short phrases with no canonicalized relations (e.g., as in the Quasimodo CSKB); (iii) The ranking based on the LM's negativeness prediction is not interpretable, and follows no clear trend.

Approach and contributions. This paper presents the UNCOMMONSENSE method for identifying *informative negations* about concepts in CSKBs. For a target concept like `gorilla`, we first compute a set of comparable concepts (e.g., `lion`, `zebra`), by employing both structured taxonomies and latent similarity. Among these concepts, we postulate a Local Closed-world Assumption (LCWA) [13], and consider their positive statements that do not hold for the target concept as candidate negations (e.g, has tail, is territorial). To eliminate false positives, candidates are scrutinized against related statements in the input KB using sentence embeddings, and against a pre-trained LM acting as an external source of latent knowledge. In a final step, we quantify the informativeness of negative statements by statistical scores, and generate top-ranked negations with provenances showing why certain negations are interesting. For instance, `¬(gorilla, has, tail)`, unlike other *land mammals*, e.g., `lion` and `zebra`.

The salient contributions of this work are:

(1) We present a method for identifying *informative* negations about everyday concepts in large-scale CSKBs grounded in taxonomic hierarchies between concepts.
(2) We showcase the ability of our method to produce *interpretable* negations via human-readable phrases.
(3) In intrinsic evaluations, our method achieves up to 18% improvement in informativeness and 17% in recall, compared to the prior state-of-the-art.
(4) In three extrinsic evaluations, (i) trivia summaries, (ii) KB completion, and (iii) multiple-choice QA, our method shows substantial improvements in informativeness.
(5) We release the first large dataset of informative commonsense negations, containing over 6 Million negative statements about 8,000 concepts.[1]

2 PROBLEM AND DESIGN SPACE

A commonsense KB consists of a finite set of statements in the form `(s, r, t)`, where `s` is a subject (or concept), `r` is a relation, and `t` is a tail phrase. Following previous work [7], we do not distinguish between `r` and `t`, because for textual CSK expressions, these distinctions are often ad-hoc and a crisp definition of relations is difficult. Hence, for the remainder of the paper, we generalize the above form to `(s, f)`, where `s` is the subject and `f` is a short phrase combining `r` and `t`.

Definition 2.1. A commonsense negative statement `¬(s, f)`, is a statement `(s, f)` 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 the CWA (closed-world assumption) over the KB and consider all non-existing statements as negatives. On top of not being materializable, this approach faces the following challenges. **C1: Avoid false negatives.** In order to assert a negation, it is not sufficient to check if a candidate negation is not positive. KBs in general operate under the OWA (open-world assumption), which means that absent information is merely *unknown*, and not necessarily false. For example, in Ascent, the absence of statement `(elephant, has eye)` is clearly due to missing information. **C2: Generate judgeable negations.** Whether constructed using human crowdsourcing [38] or information extraction techniques [24, 33], KBs mainly reflect the "wisdom" of the crowd about everyday concepts. This causes the augmentation of many subjective or otherwise uninformative statements, such as `(cat, is important)` and `(football, is boring)`. A generated negation 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. **C3: Generate informative negations**. Finally, the explicit materialization of all possible negations is not necessary for most standard AI applications (e.g., user might confuse `tabbouleh` as something that requires an oven but not a `printer`). In other words, it is better to avoid nonsensical negative statements such as `¬(printer, is baked in oven)`.

Research problem. Given a target concept `s` in a CSKB, generate a ranked list of *truly negative* and *informative* statements.

3 THE UNCOMMONSENSE FRAMEWORK

We present UNCOMMONSENSE, a method for automatically identifying informative 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) [13] is made. These relevant positives are then considered as potential informative negations for `s`. Consequently, these candidates might contain many false negatives and nonfactual statements. This is followed by an inspection step, where we use KB-based and LM-based checks to measure the plausibility of candidates. Finally, to measure informativeness, the remaining candidates are scored using relative frequency. An overview is shown in Figure 1.

3.1 Identifying Comparable Concepts

To increase the thematic relevance of candidate negations, we define the parts of the KB where the CWA is helpful to assume [13], i.e., the LCWA. For instance, if the target concept is an animal, negations should mostly reflect animal-related statements such as "*not*

[1] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/commonsense/uncommonsense

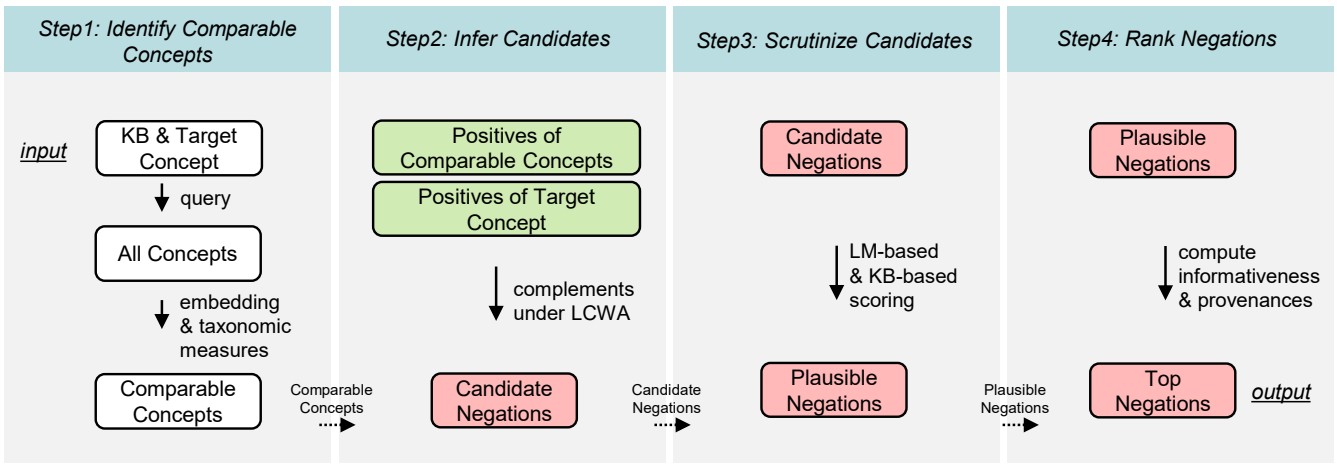

Figure 1: Architecture of UnCommonSense.

carnivorous" or "*not nocturnal*", instead of "*not beverage*" or "*cannot store data*". Therefore, we need to collect *comparable* concepts [1]. One way for collecting related concepts is using pre-computed embeddings. For instance, `elephant` is related to both `tiger` and `lion`, due to their proximity in the vector space [46]. 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 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 sibling, 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 [15], `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 [49], we compute the cosine similarity score between embeddings of `s` and every other concept in the KB, and rank them by descending order of similarity.
(2) Using hypernymy relations [45], we retain siblings that are co-hyponyms of `s`. In particular, for every concept, we collect the top-5 hypernyms (ranked by confidence score[2]). 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 KB 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 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 hyperparamater γ, which we tune in Section 5. For the remainder of the paper, we use the terms *comparable concepts* and *siblings* interchangeably.

Example 3.1. Given `s` = `elephant` from KB = *Ascent*, and $\gamma = 3$, the concepts with the highest cosine similarity are computed using Wikipedia2Vec [49]. The ranked candidate concepts include `tiger`, `lion`, `trunk`, `horse`, ... Here, `trunk` is an obvious intruder as it is not share a hypernym with `s`. This is determined using WebIsALOD [15, 16], 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`. This initial step is meant to address challenge **C3**, which we further demonstrate in Section 5 (Table 8).

3.2 Candidate Negation Inference

To produce a set of candidate negations, we query from the KB 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:

$$N = B \setminus A \tag{1}$$

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 3.2. `elephant`'s statements (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 negation set N is then all the phrases in the siblings' set except for `has tongue`, which is a straightforward contradiction with positives about `elephant`.

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

3.3 Scrutinizing Candidates

Plausibility checks. To remove candidates that might be inaccurate due the KB's incompleteness, and address **C1**, we measure the plausibility of our candidate negations in two steps:

(1) *KB-based scoring*: Unlike encyclopedic KB (e.g., Wikidata [44]), statements in CSKB 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 [31] to capture semantically-close statements in the KB, namely semantically-close information between the concept's and siblings' positives. We filter out candidates that are highly similar to information we already know about the target concept.
(2) *LM-based scoring*: In open-world KBs, it is not sufficient to perform a plausibility check against the knowledge in the KB, 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 [30], 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 3.3. Using Sentence-BERT (or SBERT [31]), 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 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 [8], we construct a probe with 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 negations such as ¬(classroom, is bigger) or ¬(basketball, is important), and address **C2**, we identify frequent statements that are highly uninformative. Inspired by the notion of term-weighting in IR [22] (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 follows: A statement is *generic* if it holds for $\geq \beta$ of the concepts in the KB.

Example 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.

3.4 Quantifying Informativeness

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 informativeness 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 f about target concept s and its siblings $\{x_1, x_2, .., x_\gamma\}$, we measure f's informativeness using *strict sibling frequency*.

$$strict(\mathsf{s}, \mathsf{f}, \{x_1, x_2, .., x_\gamma\}) = \frac{|\{x_i | (x_i, \mathsf{f}) \in \text{KB}\}|}{\gamma} \tag{2}$$

Example 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 *all* their siblings.

Relaxed scoring. The *strict* informativeness scoring only handles the cases where candidate negations 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 [31] in order to capture this similarity and boost the scores of candidates. We measure f's informativeness using *relaxed sibling frequency* as follows.

$$\begin{aligned} &relaxed(\mathsf{s}, \mathsf{f}, \{(x_1, [\mathsf{f}'^1_1, \dots]), \dots, (x_\gamma, [\mathsf{f}'^1_\gamma, \dots])\}) = \\ &\frac{|\{x_i | (x_i, \mathsf{f}'^j_i) \in \text{KB} \wedge (\mathsf{f} = \mathsf{f}'^j_i \vee (sim(\mathsf{f}, \mathsf{f}'^j_i) \geq \lambda))\}|}{\gamma} \end{aligned} \tag{3}$$

where f'^j_i is a phrase that holds for sibling x_i and $sim(\mathsf{f}, \mathsf{f}'^j_i)$ is the semantic similarity between candidate f and candidate-rephrase f'^j_i.

Example 3.6. Candidates (can jump) and (can leap) are semantically similar, 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 [35], negations 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 phrase f.

$$score(h, \mathtt{s}, \mathtt{f}, \{x_1, x_2, .., x_n\}) = \frac{|\{x_i|(x_i, \text{isA}, h) \in \text{TX}\}|}{n} \quad (4)$$

where TX is the taxonomic relations database, e.g., WebIsALOD [15], $(x_i, \text{isA}, h) \in$ TX indicates that hypernym h holds for sibling x_i in TX, and n is the total number of siblings candidate-phrase `f` holds for.

Example 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.

4 EXPERIMENTS

The evaluation of UnCommonSense is centered around answering the 3 challenges introduced in Section 2. Hence, we conduct:

(1) An intrinsic evaluation to demonstrate the ability of our method to produce plausible (**C1, C2**) and informative (**C3**) commonsense negative knowledge against baseline and state-of-the-art methods. We demonstrate the ability of our method to extend negations with valuable information (**C3**) by an evaluation of provenances.
(2) Three extrinsic evaluations:
   (a) A *negative-trivia* use-case where we evaluate the quality of summaries about concepts (**C2, C3**).
   (b) A *KB completion* use-case where we provide challenging negative examples for LM-based triple classifier (**C1, C3**).
   (c) A *multiple-choice QA* use-case, where we utilize our model as an eliminator to exclude improbable options (**C1**).

4.1 Setup

Data Source: Ascent CSKB. We use Ascent++ [24] as our input CSKB (in the following just called Ascent). This choice is motivated by the fact that computing negations benefits from richer input sets (i.e, high statement-recall per concept). In comparison, in ConceptNet, the most prominent CSKB, has 23 statements per concept on average. Ascent, on the other hand, has 256. Moreover, Ascent contains 2m assertions for 23k subjects. We restrict our evaluation to the 8k *primary* subjects and disregard *aspects* and *subgroups*.

Baselines and implementation. We compare our method to the following baselines.

- **CWA**: In this baseline, the KB is simply assumed to be complete. For a given concept, any phrase not asserted gives an immediate negation.
- **Quasimodo$^{\text{neg}}$**: We download the latest version of Quasimodo [33] and retrieve all the statements with *negative polarity* (a total of 350k negations, e.g., ¬(`baby, has hair`)).
- **GPT-3$^{\text{neg}}$**: We prompt GPT-3 [29] 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 relations, 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 concepts.
- **NegatER-θ_r** [35]: This work presents an unsupervised method that ranks out-of-KB potential negatives using a fine-tuned LM. We use the released code[3] 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.
- **NegatER-∇** [35]: 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`).

All the extracted/generated negations of the three external methods are released for future comparisons[4].

UnCommonSense Variants. We consider three variants.

- **UnCommonSense$^{\text{B}}$**: This baseline variant computes comparable concepts as described in Section 3.1, but suspends the scrutinizing and ranking steps.
- **UnCommonSense$^{\text{S}}$**: The complete method, with informativeness computed using strict ranking (i.e., Equation 2).
- **UnCommonSense$^{\text{R}}$**: The complete method, with informativeness computed using relaxed ranking (i.e., Equation 3).

For all variants, 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. Moreover, we collect taxonomic siblings from WebIsALOD [15] and order them using Wikipedia2Vec [49]. We use SBERT [31] for sentence similarity checks and use BERT [8] for LM-based checks.

[3] https://github.com/tsafavi/NegatER
[4] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/commonsense/uncommonsense

4.2 Intrinsic Evaluation

Human plausibility and informativeness evaluation. We conduct a human evaluation[5] to determine the quality of each method in generating plausible and salient negations. We randomly sample 200 concepts and produce for each top-2 negations. We then acquire 3 annotations for each negation via crowdsourcing. The total number of annotated negations is 200 concepts × 2 negations × 8 methods × 3 annotations = 9.6k rows. We ask every annotator to answer two questions, given a statement about a concept: 1) Is this statement truly negative?, 2) Is this 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 [11] inter-annotator agreement is 0.46, i.e. moderate agreement. We interpret this slightly underwhelming agreement on this relatively easy task by the 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 informativeness and plausibility are shown in Table 1 and qualitative examples in Table 2. The *false negatives* column reflects the ratio of result-negations that are in fact positive (i.e., not plausible). Obviously, the CWA baseline dominates with only 0.07% false inferences, as the majority of the produced negations are accurate but *nonsensical* (e.g., in Table 2, "*rabbits are not related to bribery*"). On the notion of *informativeness*, the leading method is UnCommonSense in its both ranking variants, outperforming the second best external method, Quasimodo^{neg}, by 18%, with a slight advantage of the strict ranking variant over the relaxed. We note that in our computation of informativeness, we only consider the negations that have been marked by the majority as truly negative. In this case, only 39% of the negations proposed by Quasimodo^{neg} are plausible, and even less for GPT-3^{neg} with 37%, as opposed to 75% for UnCommonSense and 74% for NegatER. We observe for the baseline variant UnCommonSense^{B} that negations are mostly thematic (due to inferences based on *comparable* concepts), however not frequent enough (due to absence 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 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 informativeness of each method per topic. The results are shown in Table 3. UnCommonSense performs best on topics like animal and food with informativeness 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 judge social negations, e.g., ¬(niece, is pregnant) and ¬(alcoholic, has friend).

Automated recall evaluation. To measure recall, we collect top-200 negations, per target concept, produced by each method. Moreover, we need a ground-truth dataset with negative statements about KB concepts. We create **ConceptNet-neg** by retrieving all the statements from ConceptNet [39] v5.5 that have a negative relation. This KB allows 6 negative relations such as NotCapableOf and NotDesires. The dataset contains 14.1k negations. 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 negation in the ground-truth. In the *relaxed* mode, we use embedding similarity [31] to assess whether a generated-negation and a ground-truth-negation are of similar meaning. The recall results are shown in Figure 2. UnCommonSense outperforms all methods. The strict mode is tougher since the slightest difference between the ground-truth and method-generated negations is considered a mismatch, e.g., ¬(air conditioner, quiet) and ¬(air conditioner, quieter). Relaxing the matching rule to sentence similarity [31] allows for more forgiving comparisons. Our method reaches 26.1% in relaxed@10 (relaxed recall at top-10 negations), followed by NegatER-∇ with 9.6%, Quasimodo^{neg} with 6.3%, and finally GPT-3^{neg} 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).

Table 1: Plausibility and informativeness evaluation.

Method	False Negatives	Informativeness
CWA	0.07	0.07
Quasimodo^{neg}	0.61	0.32
GPT-3^{neg}	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

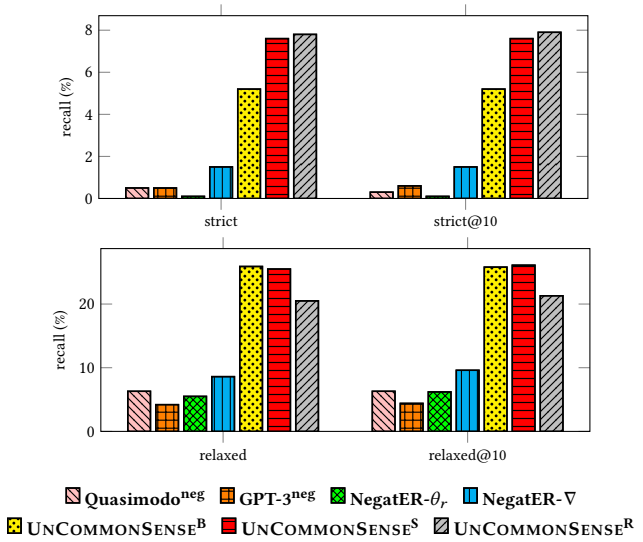

Figure 2: Recall evaluation.

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

Table 2: Intrinsic evaluation, sample results.

Method	Top negations	Truly Negative?
CWA	¬(acne, can give an understanding of truth)	✓
	¬(elephant, can provide clinician)	✓
	¬(yawning, has fluid)	✓
	¬(vinegar, can comprise about 55% nickel)	✓
	¬(rabbit, related to bribery)	✓
Quasimodo^{neg}	¬(acne, is natural)	✗
	¬(elephant, quit smoking)	✓
	¬(yawning, can end)	✗
	¬(vinegar, is vegan)	✗
	¬(rabbit, is rodent)	✓
GPT-3^{neg}	¬(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 3: Informativeness per domain.

Method	Animal	Food	Activity	Human	Object	Other
CWA	0.06	0.09	0.21	0.18	0.10	0.15
Quasimodo^{neg}	0.41	0.44	0.24	0.39	0.20	0.24
GPT-3^{neg}	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*	*niece*	*tripod*	*propaganda*

Table 4: Examples of provenance-extended negations (UnCommonSense^{V}).

Target Concept	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, fenugreek* and *acidic food*, e.g., *tomato*
ear	¬(is muscular) unlike other *body part*, e.g., *shoulder, loin, neck*

Table 5: Example of MCQA through elimination process (eliminated choice and correct choice).

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 (-)

Evaluation of provenance generation. To show the effect of extending negation 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 negations. The results are then judged by 3 annotators. We ask about the general informativeness of the negations and allow "interesting", "slightly interesting", and "not interesting". UnCommonSense^{V} outperforms UnCommonSense by 32% in informativeness, with 81% and 49% respectively. Examples are shown in Table 4. The Fleiss' kappa inter-annotator agreement of this task is 0.44, i.e., moderate.

4.3 Extrinsic Evaluation I: 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 negations for each, and show them to annotators at once. We compare the best version of our model (UnCommonSense^{S}) as the default, and the best of NegatER (NegatER-∇), as well as Quasimodo^{neg} and GPT-3^{neg}. This results in a total of 2.4k annotations (200 concepts $\times$ 4 methods $\times$ 3 annotations). For every list of negations for a given concept, we ask the annotators whether the list 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% informativeness, followed by GPT-3^{neg} (40%), NegatER (30%), and finally Quasimodo^{neg} (23%). An example is top negations about the concept `pancake`: While Quasimodo^{neg} and GPT-3^{neg} are low on plausibility, `¬(pancake, is vegan)` and `¬(pancake, is eaten)`, respectively, UnCommonSense offers the most plausible and informative negations e.g., `¬(pancake, is crumbly)`.

4.4 Extrinsic Evaluation II: KB Completion

KB completion refers to the task of identifying novel *positive* statements not yet in a KB. Recent works approach this as an LM-based true/false classification task on candidate statements [35]. A crucial ingredient for this approach are 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 KB completion accuracy. Following the setup of [35], we compare the impact of negations generated by UnCommonSense with that of COMET [6] and NegatER[6]. We use the code by [35] to train a BERT-based KB 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 6. UnCommonSense shows a statistically significant improvement over all methods with $\alpha < 0.01$.

4.5 Extrinsic Evaluation III: 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 [42]. Examples are shown in Table 5. 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 KB (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 KB-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 negation produced for the target concept (similarity is again measured using SBERT with threshold=0.7). The numerical results are shown in Table 7 and examples in Table 5. A helpful elimination is an deletion of a *wrong answer* and an unhelpful one is a deletion of a *correct answer*. The CWA baseline eliminates most of the options since the absence of the statement is enough to merit a deletion. Besides CWA, the model with the highest number of helpful eliminations is UnCommonSense with 108, followed by NegatER with 35.

Table 6: KB completion evaluation.

Negation Generator	Accuracy (%)
CWA	75.89
COMET	79.06
NegatER	78.61
UnCommonSense	79.56

Table 7: Eliminations for MCQA task.

Eliminator	Helpful	Unhelpful
CWA	290 (72.5%)	72 (72.0%)
Quasimodo^{neg}	17 (4.3%)	1 (1.0%)
NegatER	35 (8.8%)	11 (11.0%)
UnCommonSense	108 (27%)	22 (22.0%)

5 ANALYSIS

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 negations?* and *does the ranking improve the informativeness?* We run our method on the 200 concepts from Section 4.2 and follow the same crowdsourcing setup for 4 different configurations of our method (4 configurations $\times$ 200 concepts $\times$ 2 negations $\times$ 3 annotators). The Fleiss' kappa inter-annotator agreement of this task is fair on both tasks, namely 0.33 on plausibility and 0.26 on informativeness. The results are shown in Table 8. One can see that without comparable concepts (instead random) to derive good thematic candidates from, the informativeness drops to almost half of the complete-configuration (i.e., UnCommonSense^{S}). This is different from the CWA baseline in Section 4 in that we still scrutinize and rank the candidate

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

Table 8: Ablation study results.

Configuration	False Negatives	Informativeness
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*

set. The informativeness 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.

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 (rank threshold for LM), and β (KB threshold for too-generic statements) . We experimented with different values for these parameters, and set them to their ideal values in Section 4 as shown in Figure 3, namely γ to 30, λ to 0.7, τ to 50, and β to 0.05.

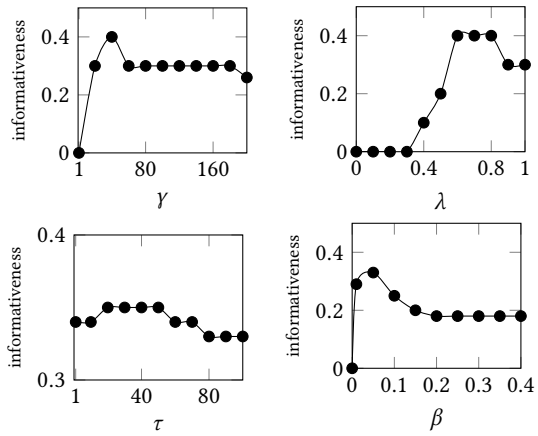

Figure 3: Hyperparameters Tuning.

6 RELATED WORK

Commonsense knowledge bases. Commonsense knowledge acquisition includes several large-scale projects. ConceptNet [38, 39], the most prominent of these projects, was mainly constructed using human crowdsourcing. Similarly, ATOMIC [36] was also constructed using crowdsourcing, with its main focus on collecting *social* commonsense statements. WebChild [43] uses handcrafted extraction patterns. TupleKB [23] and Quasimodo [32, 33] rely on open information extraction [26] followed by cleanup. Ascent [24, 25] builds on these approaches by extending them to large-scale web text extraction, for better corroboration and higher recall.

Negation in knowledge bases. ConceptNet [38] allows the expression of negative statements using 6 pre-defined negative relations. We use these statements in our recall evaluation. The text-extracted Quasimodo [33] contains 350k negated statements (i.e., with negative *polarity*), yet many have quality issues due to problems with the data source or extraction pipeline. We filter these negated statements from the full KB and use them as a baseline in our experiments (i.e., Quasimodo$^{\text{neg}}$). On actively collecting interesting negations, recently, an inference model has been proposed to build a knowledge graph [27] with if-then commonsense contradictions [17]. Unlike our work, [17] 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 [34, 35]. It proposes using LMs to discover meaningful negations. 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 is not enough to obtain informative negations. Other approaches that target *salient* negations in encyclopedic knowledge bases, such as Wikidata [44] and Yago [40], include statistical inferences [2–5] and text extractions [2, 18]. 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.

Language Models. In recent years, Language Models (LMs) showed their ability to store factual knowledge, learned from pre-training data [28, 37]. 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 [14, 31], which helps with the problem of detecting semantic similarity. However, LMs have also been repeatedly shown to struggle with explicit negation [19, 41]. We make use of these models in order to scrutinize our candidate negations and make our rankings stronger via the relaxed sibling frequency.

7 RESOURCES

We release a large dataset as a resource for further research: Up to[7] top-1k negations for all primary concepts from Ascent [24], containing 6.2m negations[8].

8 CONCLUSION

In this work, we presented the UnCommonSense framework for compiling informative negative statements about everyday concepts, by exploiting comparable concepts in commonsense knowledge bases. Our method outperforms baselines and state-of-the-art methods, on both informativeness and recall. Potential future directions include considering further types of negation [20], e.g., conditioned and enriched with semantic facets [25], like "*female* lions do not have manes", and exploring better sources for negative social knowledge [36], which comes with novel challenges due to a lack of previous work on taxonomic organization of activities.

ACKNOWLEDGMENTS

Funded by the German Research Foundation (DFG: Deutsche Forschungsgemeinschaft) - Project 453095897 - "Negative Knowledge at Web Scale".

[7] For some concepts, the final candidate set contains less than 1k candidates due to lack of enough siblings' positives and/or deletion during scrutiny steps
[8] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/commonsense/uncommonsense

REFERENCES

[1] Albin Ahmeti, Simon Razniewski, and Axel Polleres. 2017. Assessing the completeness of entities in knowledge bases. In *ESWC*.

[2] Hiba Arnaout, Simon Razniewski, and Gerhard Weikum. 2020. Enriching Knowledge Bases with Interesting Negative Statements. In *AKBC*.

[3] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2021. Negative Knowledge for Open-World Wikidata. In *WWW Companion*.

[4] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2021. Negative statements considered useful. *JWS* (2021).

[5] Hiba Arnaout, Simon Razniewski, Gerhard Weikum, and Jeff Z. Pan. 2021. Wikinegata: A Knowledge Base with Interesting Negative Statements. In *VLDB Endowment*.

[6] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction. In *ACL*.

[7] Yohan Chalier, Simon Razniewski, and Gerhard Weikum. 2020. Joint Reasoning for Multi-Faceted Commonsense Knowledge. In *AKBC*.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

[9] Jianfeng Du, Jeff Z. Pan, Sylvia Wang, Kunxun Qi, Yuming Shen, and Yu Deng. 2019. Validation of Growing Knowledge Graphs by Abductive Text Evidences. In *AAAI*.

[10] Aidan Hogan et al. 2021. *Knowledge Graphs*. Morgan & Claypool Publishers.

[11] J.L. Fleiss. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin* (1971).

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

[13] Luis Galárraga, Simon Razniewski, Antoine Amarilli, and Fabian M Suchanek. 2017. Predicting completeness in knowledge bases. In *WSDM*.

[14] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

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

[16] Sven Hertling and Heiko Paulheim. 2018. Provision and usage of provenance data in the WebIsALOD Knowledge Graph. In *CKGSemStats@ISWC*.

[17] L. Jiang, A. Bosselut, C. Bhagavatula, and Y. Choi. 2021. "I'm Not Mad": commonsense Implications of Negation and Contradiction. In *NAACL-HLT*.

[18] G. Karagiannis, I. Trummer, S. Jo, S. Khandelwal, X. Wang, and C. Yu. 2019. Mining an "anti-knowledge base" from Wikipedia updates with applications to fact checking and beyond. In *PVLDB*.

[19] Nora Kassner and Hinrich Schütze. 2020. Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly. In *ACL*.

[20] Peter LoBue and Alexander Yates. 2011. Types of Common-Sense Knowledge Needed for Recognizing Textual Entailment. In *ACL*.

[21] Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Commonsense Knowledge Base Completion with Structural and Semantic Context. In *AAAI*.

[22] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[23] Bhavana Dalvi Mishra, Niket Tandon, and Peter Clark. 2017. Domain-Targeted, High Precision Knowledge Extraction. *TACL* (2017).

[24] Tuan-Phong Nguyen, Simon Razniewski, Julien Romero, and Gerhard Weikum. 2021. Refined Commonsense Knowledge from Large-Scale Web Contents. *ArXiv* (2021).

[25] Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2021. Advanced Semantics for Commonsense Knowledge Extraction. In *WWW*.

[26] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A Survey on Open Information Extraction. In *COLING*.

[27] J.Z. Pan, G. Vetere, J.M. Gomez-Perez, and H. Wu. 2016. *Exploiting Linked Data and Knowledge Graphs for Large Organisations*. Springer.

[28] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In *EMNLP*.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI technical report* (2019).

[30] Simon Razniewski, Andrew Yates, Nora Kassner, and Gerhard Weikum. 2021. Language Models As or For Knowledge Bases. *DL4KG* (2021).

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

[32] Julien Romero and Simon Razniewski. 2020. Inside Quasimodo: Exploring Construction and Usage of Commonsense Knowledge. In *CIKM*.

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

[34] T. Safavi and D. Koutra. 2020. Generating Negative Commonsense Knowledge. In *NeurIPS*.

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

[36] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2019. ATOMIC: An Atlas of Machine Commonsense for If-Then Reasoning. In *AAAI*.

[37] Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *EMNLP*.

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

[39] Robyn Speer and Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In *LREC*.

[40] F. Suchanek, G Kasneci, and G. Weikum. 2007. Yago: A Core of Semantic Knowledge. In *WWW*.

[41] Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-On What Language Model Pre-training Captures. *TACL* (2020).

[42] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *NAACL*.

[43] Niket Tandon, Gerard de Melo, Fabian Suchanek, and Gerhard Weikum. 2014. WebChild: Harvesting and Organizing Commonsense Knowledge from the Web. In *WSDM*.

[44] D. Vrandečić and M. Krötzsch. 2014. Wikidata: a Free Collaborative Knowledge base. *CACM* (2014).

[45] Chengyu Wang, Xiaofeng He, and Aoying Zhou. 2017. A Short Survey on Taxonomy Learning from Text Corpora: Issues, Resources and Recent Advances. In *EMNLP*.

[46] Q. Wang, Z. Mao, B. Wang, and L. Guo. 2017. Knowledge Graph Embedding: a Survey of Approaches and Applications. *IEEE TKDE* (2017).

[47] Gerhard Weikum, Xin Luna Dong, Simon Razniewski, and Fabian M. Suchanek. 2021. Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. *Found. Trends Databases* (2021).

[48] Kemas Wiharja, Jeff Z. Pan, Martin J. Kollingbaum, and Yu Deng. 2020. Schema Aware Iterative Knowledge Graph Completion. *Journal of Web Semantics* (2020).

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

[50] Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. 2020. TransOMCS: From Linguistic Graphs to Commonsense Knowledge. In *IJCAI*.