Advanced Semantics for Commonsense Knowledge Extraction

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

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

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

ABSTRACT

Commonsense knowledge (CSK) about concepts and their properties is useful for AI applications such as robust chatbots. Prior works like ConceptNet, TupleKB and others compiled large CSK collections, but are restricted in their expressiveness to subject-predicate-object (SPO) triples with simple concepts for S and monolithic strings for P and O. Also, these projects have either prioritized precision or recall, but hardly reconcile these complementary goals. This paper presents a methodology, called Ascent, to automatically build a large-scale knowledge base (KB) of CSK assertions, with advanced expressiveness and both better precision and recall than prior works. Ascent goes beyond triples by capturing composite concepts with subgroups and aspects, and by refining assertions with semantic facets. The latter are important to express temporal and spatial validity of assertions and further qualifiers. Ascent combines open information extraction with judicious cleaning using language models. Intrinsic evaluation shows the superior size and quality of the Ascent KB, and an extrinsic evaluation for QA-support tasks underlines the benefits of Ascent. A web interface, data and code can be found at https://www.mpi-inf.mpg.de/ascent.

ACM Reference Format:
Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2021. Advanced Semantics for Commonsense Knowledge Extraction. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449827

1 INTRODUCTION

Motivation. Commonsense knowledge (CSK) is a long-standing goal of AI [14, 26, 33]: equip machines with structured knowledge about everyday concepts and their properties (e.g., elephants are big and eat plants, buses carry passengers and drive on roads) and about typical human behavior and emotions (e.g., children love visiting zoos, children enter buses to go to school). In recent years, research on automatic acquisition of CSK assertions has been greatly advanced and several commonsense knowledge bases (CSKBs) of considerable size have been constructed (see, e.g., [35, 46, 53, 55]). Use cases for CSK include particularly language-centric tasks such as question answering and conversational systems (see, e.g., [27, 28, 59]).

Examples: Question-answering systems often need CSK as background knowledge for robust answers. For example, when a child asks "Which zoos have habitats for T-Rex dinosaurs?", the system should point out that i) dinosaurs are extinct, and ii) can be seen in museums, not in zoos. Dialogue systems should not just generate plausible utterances from a language model, but should be situative, understand metaphors and implicit contexts and avoid blunders. For example, when a user says "tigers will soon join the dinosaurs", the machine should understand that this refers to an endangered species rather than alive tigers invading museums.

The goal of this paper is to advance the automatic acquisition of CSK assertions from online contents better expressiveness, higher precision and wider coverage.

State of the Art and its Limitations. Large KBs like DBpedia, Wikidata or Yago largely focus on encyclopedic knowledge on individual entities like people, places etc., and and are very sparse on general concepts [24]. Notable projects that focus on CSK include ConceptNet [53], WebChild [55], Mosaic TupleKB [35] and Quasimodo [46]. They are all based on SPO triples as knowledge representation and have major shortcomings:

- *Expressiveness for S:* As subjects, prior CSKBs strongly focus on simple concepts expressed by single nouns (e.g., elephant, car, trunk). This misses semantic refinements (e.g., diesel car vs. electric car) that lead to different properties (e.g., polluting vs. green), and is also prone to word-sense disambiguation problems (e.g., elephant trunk vs. car trunk). Even when CSK acquisition considers multi-word phrases, it still lacks the awareness of semantic relations among concepts. Hypernymy lexicons like WordNet or Wiktionary are also very sparse on multi-word concepts. With these limitations, word-sense disambiguation does not work robustly; prior attempts showed mixed results at best (e.g., [35, 55]).
- *Expressiveness for P and O:* Predicates and objects are treated as monolithic strings, such as
  - A1: buses, [used for], [transporting people];
  - A2: buses, [used for], [bringing children to school];
  - A3: buses, [carry], [passengers];
  - A4: buses, [drop], [visitors at the zoo on the weekend].

  This misses the equivalence of assertions A1 and A3, and is unable to capture the semantic relation between A1 and A2, namely, A2 refining A1. Finally, the spatial facets of A2 and A4 are cluttered into unrelated strings, and the temporal facet in A4 is not explicit either. The alternative of restricting P to a small number of pre-specified predicates (e.g., [53, 55]) and O to very short phrases comes at the cost of much lower coverage.
- *Quality of CSK assertions:* Some of the major CSKBs have prioritized precision (i.e., the validity of the assertions) but have fairly limited coverage (e.g., [35, 53]. Others have wider coverage but include many noisy if not implausible assertions (e.g., [46, 55]). Very few have paid attention to the saliency of assertions, i.e., the degree to which statements are common knowledge, as

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449827

opposed to merely capturing many assertions. Projects along these lines (e.g., [47, 53]) fall short in coverage, though.

Ascent aims to overcome these limitations of prior works, while retaining their positive characteristics. In particular, we aim to reconcile high precision with wide coverage and saliency. Like [35, 46], we aim to acquire open assertions (as opposed to pre-specified predicates only), but strive for more expressive representations by refining subjects and capturing semantic facets of assertions.

Approach. We present the Ascent method for acquiring CSK assertions with advanced semantics, from web contents. Ascent operates in three phases: (i) source discovery, (ii) open information extraction (OIE), (iii) automatic consolidation. In the first phase, Ascent generates search queries for a given target concept such as "star" to retrieve relevant pages. The queries include hypernyms from lexicons such as WordNet, this way covers different meanings of "star" while distinguishing results for "star (celebrity)" (with hypernym "human") vs. "star (celestial body)" (with hypernym "natural object"). Results are further scrutinized by comparing, via embedding similarity, against the respective Wikipedia articles. In the second phase, Ascent collects OIE-style tuples by carefully designed dependency-parse-based rules, taking into account assertions for subgroups and aspects of target subjects, and increasing recall by co-reference resolution. The extractors use cues from prepositional phrases to detect semantic facets, and use supervised classification for eight facet types. Finally, in the consolidation phase, assertions are iteratively grouped and semantically organized by an efficient combination of filtering based on fast word2vec similarity, and classification based on a fine-tuned RoBERTa language model.

We ran Ascent for 10,000 frequently used concepts as target subjects. The resulting CSKB significantly outperforms automatically built state-of-the-art CSK collections in salience and recall. In addition, we performed an extrinsic evaluation in which commonsense knowledge was used to support language models in question answering. Ascent significantly outperformed language models without context, and was consistently among the top-scoring KBs in this evaluation.

Contributions. Salient contributions of this work are:

- introducing an expressive model for commonsense knowledge with advanced semantics, with subgroups of subjects and faceted assertions as first-class citizens;
- developing a fully automated methodology for populating the model with high-quality CSK assertions by extraction from web contents;
- constructing a large CSKB for 10,000 important concepts.

A web interface to the Ascent KB, along with downloadable data and code is available at `https://www.mpi-inf.mpg.de/ascent`.

2 RELATED WORK

Commonsense knowledge bases (CSKBs). CSK acquistion has a long tradition in AI (e.g., [19, 26, 30, 51]). A few projects have constructed large-scale collections that are publicly available. ConceptNet [53] is the most prominent project on CSK acquisition. Relying mostly on human crowdsourcing, it contains highly salient information for a small number of pre-specified predicates (isa/type, part-whole, used for, capable of, location of, plus lexical relations such as synonymy, etymology, derived terms etc.), and this CSKB is most widely used. However, it has limited coverage on many concepts, and its ranking of assertions, based on the number of crowdsourcing inputs, is very spares and unable to discriminate salient properties against atypical or exotic ones (e.g., listing trees, gardens and the bible as locations of snakes, with similar scores). ConceptNet does not properly disambiguate concepts, leading to incorrect assertion chains like ⟨*elephant, hasPart, trunk*⟩; ⟨*trunk, locationOf, spare tire*⟩.

WebChild [55], TupleKB [35] and Quasimodo [46] devised fully automated methods for CSKB construction. They use judiciously selected text corpora (incl. book n-grams, image tags, QA forums) to extract large amounts of SPO triples. WebChild builds on handcrafted extraction patterns, and TupleKB and Quasimodo rely on open information extraction with subsequent cleaning. All three are limited to SPO triples.

Recently, TransOMCS [60] has harnessed statistics about preferential attachment to convert a large linguistic collection of patterns into a CSKB of SPO triples with a pre-specified set of predicates. It uses Transformer-based neural learning for plausibility scoring.

We adopt the idea of using search engines for source discovery and open information extraction (OIE). Our novelty for source discovery lies in generating better focused queries and scrutinizing candidate documents against reference Wikipedia articles. For extraction, we extend OIE to capture expressive facets and also multi-word compounds as subjects. Multi-word compounds enable a higher recall on salient assertions, as well as avoiding common disambiguation errors.

Taxonomy and meronymy induction. The organization of concepts in terms of subclass and part-whole relationships, termed hypernymy and meronymy, has received great attention in NLP and web mining (e.g., [13, 17, 22, 40, 41, 44, 52, 58]). The handcrafted WordNet lexicon [34] organizes over 100k synonym sets with respect to these relationships, although meronymy is sparsely populated.

Recent methods for large-scale taxonomy induction from web sources include WebIsADB [22, 49] building on Hearst patterns and other techniques, and the industrial GIANT ontology [29] based on neural learning from user-action logs and other sources.

Meronymy induction at large scale has been addressed by [1, 2, 56] with pre-specified and automatically learned patterns for refined relations like physical-part-of, member-of and substance-of.

Our approach includes relations of both kinds, by extracting knowledge about salient subgroups and aspects of subjects. In contrast to typical taxonomies and part-whole collections, our subgroups include many multi-word phrases: composite noun phrases (e.g., "circus lion", "lion pride") and adjectival and verbal phrases (e.g., "male lion", "roaring lion"). Aspects cover additional refinements of subjects that do not fall under taxonomy or meronymy (e.g., "lion habitat" or "lion's prey").

Expressive knowledge representation and extraction. Modalities such as `always, often, rarely, never` have a long tradition in AI research (e.g., [16]), based on various kinds of modal logics or semantic frame representations, and semantic web formalisms can

capture context using e.g., RDF* or reification [23]. While such expressive knowledge representations have been around for decades, there has hardly been any work that populated KBs with such refined models, notable exceptions being the Knext project [48] at small scale, and OntoSenticNet [11] with focus on affective valence annotations.

Other projects have pursued different kinds of contextualizations for CSK extraction, notably [61], which scored natural language sentences on an ordinal scale covering the spectrum *very likely, likely, plausible, technically possible* and *impossible*, Chen et al. [6] with probabilistic scores, and the Dice project [5] which ranked assertions along the dimensions of plausibility, typicality and saliency.

Semantic role labelling (SRL) is a representation and methodology where sentences are mapped onto frames (often for certain types of events) and respective slots (e.g., agent, participant, instrument) are filled with values extracted from the input text [8, 39, 54]. Recently, this paradigm has been extended towards facet-based open information extraction, where extracted tuples are qualified with semantic facets like location and mode [4, 45]. Ascent builds on this general approach, but extends it in various ways geared for the case of CSK: focusing on specifically relevant facets, refining subjects by subgroups and aspects, and aiming to reconcile precision and coverage for concepts as target subjects.

Pre-trained language models. Recently, there has been great progress on pre-trained language models (LMs) like BERT and GPT [3, 10]. In Ascent we make use of such language models, utilizing them to cluster semantically similar phrases in order to reduce redundancy and group related assertions. We also use LMs in the extrinsic evaluation for question answering, showing that priming LMs with structured knowledge from CSKBs can greatly improve performance (cf. also [42]).

3 MODEL AND ARCHITECTURE

3.1 Knowledge Model

Existing CSKBs typically follow a triple-based data model, where subjects are linked via predicate phrases to object words or phrases. Typical examples, from ConceptNet, are ⟨*bus, usedFor, travel*⟩ and ⟨*bus, usedFor, not taking the subway*⟩. Few projects [35, 55] have attempted to sharpen such assertions by word sense disambiguation (WSD) [36], distinguishing, for example, buses on the road from computer buses. Likewise, only few projects [5, 20, 46, 61], have tried to identify salient assertions against correct ones that are unspecific, atypical or even misleading (e.g., buses used for avoiding the subway or used for enjoying the scenery). We extend this prevalent paradigm in two major ways.

Expressive subjects. CSK acquisition starts by collecting assertions for target subjects, which are usually single nouns. This has two handicaps: 1) it conflates different meanings for the same word, and 2) it misses out on refinements and variants of word senses. While word sense disambiguation (WSD) has been tried to overcome the first issue [35, 55], it has been inherently limited because the underlying word-sense lexicons, like WordNet and Wiktionary, mostly restrict themselves to single nouns. For example, phrases like "city bus" or "tourist bus" are not present at all.

Our approach to rectify this problem is twofold:

- First, our source discovery method combines the target subject with an informative hypernym (using WordNet, applied to single nouns or head words in phrases). For example, instead of searching with the semantically overloaded word "bus", we generate queries "bus public transport" and "bus network topology" to disentangle the different senses.
- Second, when extracting candidates for assertions from the retrieved web pages, we capture also multi-word phrases as candidates for refined subjects, such as "school bus", "city bus", "tourist bus", "circus elephant", "elephant cow", "domesticated elephant", etc. This way, we can acquire *isa*-like refinements, to create *subgroups* of broader subjects, and also other kinds of *aspects* that are relevant to the general concept. An example for the latter would be "bus driver" or, for the target subject "elephant", phrases such as "elephant tusk", "elephant habitat" or "elephant keeper".

Our notion of *subgroups* can be thought of as an inverse *isa* relation. It goes beyond traditional taxonomies by better coverage of multi-word composites (e.g., "circus elephant"). This allows us to better represent specialized assertions such as ⟨*circus elephants, catch, balls*⟩.

Our notion of *aspects* includes part-whole relations (partOf, memberOf, substanceOf) [2, 17, 50, 56], but also further aspects that do not fall under the themes of hypernymy or meronymy. Examples are "elephant habitat", "bus accident", etc. Note that, unlike single nouns, such compound phrases are rarely ambiguous, so we have crisp concepts without the need for explicit WSD.

Semantic facets. For CSK, assertion validity depends often on specific temporal and spatial circumstances, e.g., elephants scare away lions only in Africa, or bathe in rivers only during daytime. Furthermore, assertions often become crisper by contextualization in terms of causes/effects and instruments (e.g., children ride the bus …to go to school, circus elephants catch balls …with their trunks).

To incorporate such information into an expressive model, we choose to contextualize subject-predicate-object triples with semantic *facets*. To this end, we build on ideas from research on semantic role labeling (SRL) [8, 39, 54]. This line of research has originally been devised to fill hand-crafted frames (e.g., purchase) with values for frame-specific roles (e.g., buyer, goods, price etc.). We start with a set of 35 labels proposed in [45], a combination of those in the Illinois Curator SRL [8] and 22 hand-crafted ones derived from an analysis of semantic roles of prepositions in Wiktionary (`https://en.wiktionary.org/`). As many of these are very special, we condense them into eight widely useful roles that are of relevance for CSK: 4 that qualify the validity of assertions (degree, location, temporal, other-quality), and 4 that capture other dimensions of context (cause, manner, purpose, transitive objects).

These design considerations lead us to the following knowledge model.

Definition [Commonsense Assertion]:
Let C_0 be a set of primary concepts of interest, which could be manually defined or taken from a dictionary.
Subjects for assertions include all $s_0 \in C_0$ as well as judiciously selected multi-word phrases that contain some s_0.
Subjects are interrelated by *subgroup* and *aspect* relations: each s_0

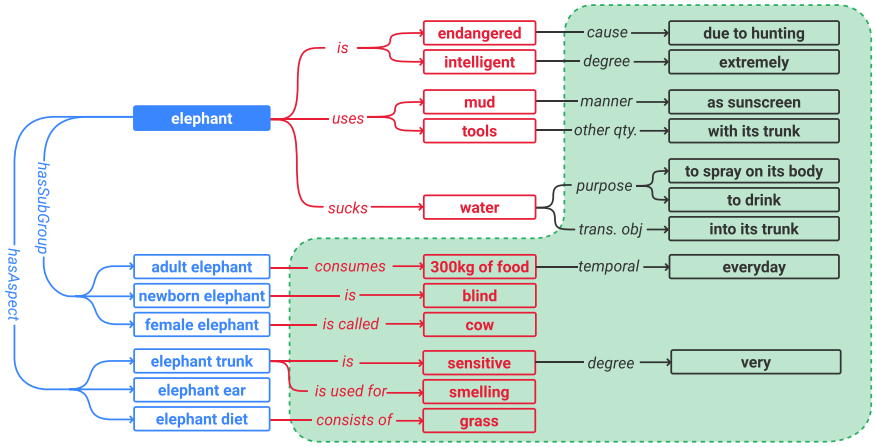


Figure 1: Example of Ascent's knowledge for the concept *elephant*. The data model of traditional CSKBs like ConceptNet is restricted to assertions outside the green box.

can be refined by a set of subgroup subjects denoted $sg(s_0)$, and by a set of aspect subjects denoted $asp(s_0)$. The overall set of subjects is $C := C_0 \cup sg_{C_0} \cup asp_{C_0}$.
A commonsense assertion for $s \in C$ is a quadruple $\langle s, p, o, F \rangle$ with single-noun or noun-phrase subject s, short phrases for predicate p and object o and a set F of semantic facets. Each facet $(k, v) \in F$ is a key-value pair with one of eight possible keys k and a short phrase as v. Note that a single assertion can have multiple key-value pairs with the same key (e.g., different spatial phrases). □

An example of assertions for s_0 = elephant is shown in Fig. 1.

3.2 Extraction Architecture

Design considerations. CSK collection has three major design points: (i) the choice of sources, (ii) the choice of the extraction techniques, and (iii) the choice of cleaning or consoliding the extracting candidate assertions.

As *sources*, most prior works carefully selected high-quality input sources, including book n-grams [55], concept definitions in encyclopedic sources, and school text corpora about science [7]. These are often a limiting factor in the KB coverage. Moreover, even seemingly clean texts like book n-grams come with a surprisingly high level of noise and bias (cf. [18]). Focused queries for retrieving suitable web pages were used by [35], but the query formulations required non-negligible effort. Query auto-completion and question-answering forums were tapped by Quasimodo [46]. While this gave access to highly salient assertions, it was, at the same time, adversely affected by heavily biased and sensational contents (e.g., search-engine auto-completion for "snakes eat" suggesting "…themselves" and "…children"). In Ascent we opt for using search engines for wide coverage, and devise techniques for quality assurance.

For the *extraction techniques*, choices range from co-occurrence- and pattern-based methods (e.g., [12]) and open information extraction (OIE) (e.g., [35, 46]) to supervised learning for classification and sequence tagging. Co-occurrence works well for a few pre-specified, clearly distinguished predicates, using distant seeds. Supervised extractors require training data for each predicate, and thus have the same limitation. Recent approaches, therefore, prefer OIE techniques, and the Ascent extractors follow this trend, too.

For *knowledge consolidation*, early approaches simply kept all assertions from the ingest process (e.g., crowdsourcing [53]), whereas recent projects employed supervised classifiers or rankers for cleaning [5, 35, 46], and also limited forms of clustering [35, 46] for canonicalization (taming semantic redundancy). In Ascent, the careful source selection already eliminates certain kinds of noise, rendering extraction frequency statistics a much better signal than in earlier works. Therefore, we focus on reinforcing these signals for consolidation, based on clustering with contextual language models for informative similarity measures.

Approach. The Ascent method operates in three phases (illustrated in Fig. 2):

1. Source discovery:
   1a. Retrieval of web pages from search engines with specifically generated queries;
   1b. Filtering of result pages based on similarity to Wikipedia reference articles.
2. Extraction of assertions with subgroups, aspects and facets:
   2a. OIE for rule-based extraction using dependency-parsing patterns;
   2b. Labeling of semantic facets by supervised classifier.
3. Clustering of assertions based on contextualized embeddings.

The following section elaborates on these steps.

4 METHODOLOGY

4.1 Relevant Document Retrieval

Web search. We use targeted web search to obtain documents specific to each subject, this way aiming to reduce the noise from out-of-context concept mentions, and the processing of large collections of mostly irrelevant documents, like encountered for instance in general web crawls. This is especially relevant as we later utilize

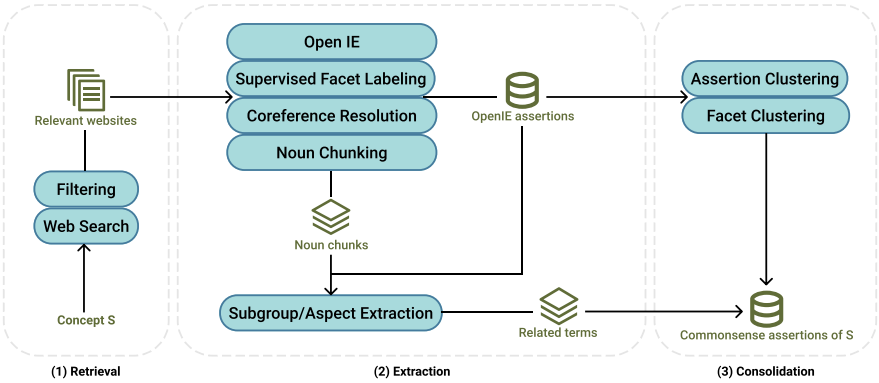


Figure 2: Architecture of our extraction pipeline.

coreference resolution, which is by itself a source of additional noise. Specifically, we utilize the Bing Web Search API.

Given a concept s_0, we first map it to a corresponding WordNet synset by taking the synset with the most lemma names, and use its hypernyms to refine search queries. For example, if s_0 has hypernym *animal.n.01* then its search query is "s_0 animal facts", or if s_0 has hypernym *professional.n.01* then its search query is "s_0 job descriptions", etc. We have manually designed templates for 35 commonly encountered hypernyms. These cover 82.5% of our subjects. When none of the templates can be applied, we default to the direct hypernym of s_0 and form the following search query: "s_0 (hypernym)". Below we provide an example of search query for the animal lynx whose WordNet synset is *lynx.n.02*, and a few top results returned by Bing.

Query: `lynx animal facts`
Top 5 results:
- `Lynx | National Geographic`
- `Interesting facts about lynx | Just Fun Facts`
- `Lynx Facts | Softschools.com`
- `Facts About Bobcats & Other Lynx | Live Science`
- `Lynx | Wikipedia`

Document filtering. Commercial search engines give us the benefit that (near-)duplicates, e.g., copies from Wikipedia, are well detected and ranked lower. At the same time, the search engine goal of diversification may introduce spurious results, despite our efforts with the search query refinement. This is exacerbated by our interest to obtain large sets of articles. We therefore propose a filter to remove irrelevant results. Given a subject s_0, we use the Bing API to retrieve 500 websites. For each website, we use a popular article scraping library[1] to scrape its main content. Next, each retrieved document is compared with a Wikipedia reference article by the cosine similarity of the bag of words of both pages. As Wikipedia reference, we leverage the WordNet-Wikipedia pairings of BabelNet [37] and the resource by [15] as the first fallback. If both resources do not contain the desired WordNet synset, we simply pick the first Wikipedia article appearing in the search result. After this, only documents with similarity higher than 0.55 (chosen based on tuning on withheld data) are retained.

[1]https://github.com/codelucas/newspaper

4.2 Knowledge Extraction

To enable the extraction of diverse pieces of information, our extraction step relies on open information extraction [32, 38]. Similarly, as open assertions typically follow a general grammatical structure, we utilize dependency-path-based rules to identify extractions. We also rely on rules to identify aspects via possessive constructions, and subgroups via compound nouns. For assigning facets to semantic groups, we use supervised models, as the set of facets is small.

Rule-based statement extraction. Our open information extraction (OIE) method builts upon the StuffIE approach [45], a series of hand-crafted dependency-parse-based rules to extract triples and facets. The core ideas are to consider each verb as a candidate predicate of an assertion, and to identify subjects, objects and facets via grammatical relations, so-called dependency paths. The elaboration below uses the Clear style format (`http://www.clearnlp.com`), as used by the spaCy dependency parser:

- Subjects are captured based on dependencies of the type subject (**`nsubj`**, **`nsubjpass`** and **`csubj`**) and adjectival clauses (**`acl`**). If no subject is found, the parent verb of the predicate identified through adverbial clause modifier (**`advcl`**) and open clausal complement (**`xcomp`**) edges is used to identify subjects.
- Dependency edges used to find objects are direct object (**`dobj`**), indirect object (**`iobj`**), nominal modifier (**`nmod`**), clausal complement (**`ccomp`**) and adverbial clause modifier (**`advcl`**).
- Once a triple has been formed, its constituents are completed by expanding their head words with related words via various dependency edges. For compound predicates, these include **`xcomp`**, **`auxpass`**, **`mwe`**, **`advmod`**. For compound subjects and objects, they are **`compound`**, **`nummod`**, **`det`**, **`advmod`**, **`amod`**.
- Finally, facets of a verb are identified through the following complements to the given verb: adverb modifier, prepositional and clausal complement.

We extend StuffIE's algorithm in the following ways:

(1) The original algorithm includes all conjuncts of head words into one assertion, thus producing often overly specific assertions. In our method we break conjunctive objects (Table 1, row 1) and facets (Table 1, row 2) into separate assertions.

No.	Sentence	StuffIE [45]	Ascent OIE extractor
1	They eat ptarmigans, voles, and grouse.	(1) They; eat; ptarmigans, voles, and grouse	(1) They; eat; ptarmigans (2) They; eat; voles (3) They; eat; grouse
2	Lynx are active during evening and early morning.	(1) Lynx; are; active (1.1) TEMPORAL: during evening and early morning	(1) Lynx; are; active (1.1) TEMPORAL: during evening (1.2) TEMPORAL: during early morning
3	Lions live for 20 years in captivity.	(1) Lions; live;_ (1.1) PURPOSE: for 20 years (1.2) LOCATION: in captivity	(1) Lions; live; for 20 years (1.1) LOCATION: in captivity
4	Lions hunt many animals, such as gnus and antelopes.	(1) Lions; hunt; many animals, such as gnus and antelopes.	(1) Lions; hunt; gnus (2) Lions; hunt; antelopes
5	Dogs are extremely smart.	(1) Dogs; are; extremely smart	(1) Dogs; are; smart (1.1) DEGREE: extremely
6	Elephants are extremely good swimmers.	(1) Elephants; are; extremely good swimmers	(1) Elephants; are; good swimmers (1.1) DEGREE: extremely

Table 1: Comparison of outputs returned by our OIE method and StuffIE.

Note that conjuncts should be connected by either *"and"* or *"or"*.

(2) The original algorithm frequently returns assertions with empty objects. To only return complete triples, in such cases, we identify the nearest prepositional facet after its predicate and convert the facet into the assertion's object (Table 1, row 3).

(3) We post-process special cases of sentences used for giving examples with the words: *"like"*, *"such as"* and *"including"* to get finer-grained output (Table 1, row 4).

(4) We convert all adverb modifiers of objects (besides those of predicates as in StuffIE) into facets. There are two types of modifiers we consider: (i) direct adverb modifiers connected to object's head word through the edge **`advmod`** (Table 1, row 5); (ii) the adverb in a noun phrase that follows the pattern "adverb + adjective + object" (Table 1, row 6).

Table 1 gives a qualitative comparison of StuffIE's and our extraction results, while in the experiment section (Table 10) we investigate their quantitative differences.

Subject and predicate postprocessing. After OIE, we perform coreference resolution[2] on paragraph level to resolve nominative pronouns occuring as subjects. For instance, the primarily extracted assertion ⟨*they, have, long trunks*⟩ will be replaced by ⟨*the elephants, have, long trunks*⟩ if "they" is resolved to "the elephants". This step helps improve the number of assertions extracted for each concept. Then, all subjects are normalized by removing determiners and punctuation, and by lemmatizing head nouns. Moreover, predicates are normalized so that main verbs are transformed to their infinite forms (e.g., "has been found in" → "be found in", "is performing" → "perform"). Finally all extracted facet words are removed from predicates and objects.

Facet type labeling. The extraction algorithm so far extracts facet values, but is unaware of their semantic type (e.g., "spatial" or "causal"). For assigning semantic types, we fine-tune a RoBERTa [31] model to classify each facet into one of the aforementioned eight types. The input sequences of RoBERTa take the form: "[CLS] subject [PRED] predicate [OBJ] object [FCT] facet [SEP]", where [PRED], [OBJ] and [FCT] are special tokens used for marking the borders between different elements. The output vector of the [CLS] token is then passed to a fully-connected layer stacked with a softmax layer on top of the transformer architecture to label the facet. Details on classifier training are in Section 5.5.

Extraction of subgroups. Subgroups could be sub-species in case of animals, or refer to the target concept in different states, such as "hunting cheetah" and "retired policeman". For subject s_0, we collect all noun chunks (normalized as for triple subjects described above) ending with s_0 or any of its WordNet lemmas as potential candidates. Semantically similar chunks, such as "Canadian lynx" and "Canada lynx", are then grouped using hierarchical agglomerative clustering (HAC) on average word2vec representations.In addition, we leverage WordNet to distinguish antonyms, with which vector space embeddings typically struggle. Note that the subgroups are restricted to be less-than-5-words, and subgroups that syntactically contain other subgroups are disregarded (e.g., "old male Canadian lynx" is grouped with "Canadian lynx"). In addition, a chunk will be ignored if it is a named entity (e.g., "Will Smith" for the concept "smith"). Finally we use WordNet hyponyms to remove spurious subgroups, e.g., "sea lion" and "ant lion" w.r.t. "lion".

Extraction of related aspects. Given subject s_0 and its WordNet lemmas L_{s_0}, related aspects of the subject are extracted from noun chunks collected from two sources:

(i) *Possessive noun chunks* where the possessives refer to any lemma in L_{s_0}, for example, "elephant's diet" and "their diet" (with resolution to "elephant");

(ii) ⟨*s, p, noun chunk*⟩ triples where $s \in L_{s_0}$ and p is one of the following verb phrases: "have", "contain", "be assembled of" or "be composed of".

In order to prevent too specific aspects (e.g., "large paws" or "short tails"), only compound nouns (if applicable) or nouns in these *noun chunks* are then extracted as aspects of s_0. For example, if we observe ⟨*lynx, have, black ear tuft*⟩, then the adjective "black" is ignored and "ear tuft" will be extracted instead of only extracting the head noun "tuft".

[2] https://huggingface.co/coref

Retained assertions. For each primary subject, a separate set of documents is processed, and the output of this stage are three sets of assertions: Assertions for the primary subject s_0, assertions for its subgroups, and assertions for its aspects. These are selected as follows.

As assertions for the main subject and its subgroups we simply retain all assertions that have a subject that matches a WordNet lemma of the primary subject, or the name of one of its subgroups.

The case of aspect assertions is slightly more complex, we merge three cases:

(1) Assertions that have a subject which is among the previously identified aspects;
(2) Assertions that have a subject among the lemmas of the main subject, and an object which is a noun chunk consisting of an aspect $t \in asp_{s_0}$ as the head noun and an adjectival modifier *adj* of t. For instance, from the assertion ⟨*elephant, have, a long very trunk*⟩ we infer that ⟨*elephant trunk, be, long, DEGREE: very*⟩.
(3) All noun chunks that follow the pattern *"possessive + adj + t"* (e.g., "elephant's long trunks"), where *possessive* refers to any lemma in L_{s_0}, *adj* is an adjectival modifier of t, and $t \in asp_{s_0}$.

Results from the latter two cases are transformed into ⟨*t, be, adj, F*⟩ assertions where the facets F are extracted from adverb modifiers of *adj*.

4.3 Knowledge Consolidation

Natural language is rich in paraphrases, and consequently, the extraction pipeline so far produces frequently assertions that carry the same or nearly the same meaning. Identifying and clustering such assertions is necessary, in order to avoid redundancies, and get better frequency signals for individual assertions.

Triple clustering. Because extraction is done for each concept separately, we only need to cluster predicate-object pairs. First, we train a RoBERTa model to detect if two given triples are semantically similar (for setup details see Sec. 5.5). Confidence scores given by the model are then used to compute distances for the HAC algorithm to group assertions into clusters. Given two assertions $\langle s, p_1, o_1 \rangle$ and $\langle s, p_2, o_2 \rangle$, the input sentence given to RoBERTa is: "[CLS] [SUBJ] p_1 [U-SEP] o_1 [SEP] [SUBJ] p_2 [U-SEP] o_2 [SEP]", where [SUBJ] and [U-SEP] are new special tokens introduced to replace identical subjects and mark the borders between predicates and objects, respectively. The output vector of the [CLS] token is used for the classification purpose in the same way as in the model used for facet labeling described above.

Ideally one would compute the full distance matrix between all assertions (an $n \times n$ matrix for n triples), but given that pretrained language models (LM) are exceedingly resource-intensive, this quadratic computation would be expensive even for moderate assertion sets. We therefore reduce the computational effort by pre-filtering the set of pairs to be compared by the pretrained LM.

(1) The assertions are sorted in decreasing order of frequency.
(2) We compute cosine similarities between vector representations of predicate-object pairs, using word2vec embeddings. This can be done very fast with parallel matrix multiplication.
(3) For each assertion a_i, we then only compute the RoBERTa-based distances with the top-k most similar assertions (ranked by word2vec-based similarities) that succeed a_i in the sorted list (the sorted list helps us focus on salient assertions). All other pairs get the distance of 1.0. This produces a "sparse" distance matrix for n assertions.
(4) For clustering, we use the HAC algorithm with single linkage, because it only looks at the most similar pairs between two clusters. That helps to reduce the chance of missing similar triples whose similarities were not computed by RoBERTa in the third step.

After clustering, the most frequent assertion inside each cluster is used as representative.

Facet value clustering. Facet values may similarly exhibit redundancy, for example, the degree facet may come with values "often", "frequently", "mostly", "regularly", etc. Also, sources may occasionally mention odd values. We combat both by clustering facet values per facet type, and retaining only the one with strongest support.

Considering the small number of facet values per assertion and facet type (usually less than 5), we utilize simple methods for clustering. Specifically, given the list of values, we use the HAC algorithm to cluster values which are adverbs, in which distance between two values is measured by the cosine distance of their word2vec presentations. Other values are grouped if they have the same head word (e.g., "during evening" and "in the evening" go to one same cluster). Similarly, the most frequent value inside a cluster is used as representative of that cluster.

5 EXPERIMENTS

The evaluation of Ascent is centered on three research questions:

- **RQ1:** Is the resulting CSKB of higher quality than existing resources?
- **RQ2:** Does (structured) CSK help in extrinsic use cases?
- **RQ3:** What is the quality and extrinsic value of facets?

We first present the implementation of Ascent, then discuss each of these research questions in its own subsection.

5.1 Implementation

We executed the pipeline for the 10,000 most popular subjects in ConceptNet (ranked by number of assertions). The execution of took a total of 10 days, of which about 5 days were spent on website crawling, 3 days on statement extraction, and 2 days on clustering. For each subject, we used the Bing Search API to retrieve 500 websites. The resulting CSKB contains 3,693,990 assertions for these primary subjects, and 1,768,538 assertions for 280,970 subgroups and 3,349,198 for 92,038 aspects. On average, half of all assertions have a facet (see Table 2).

In Table 3, we show statistics of our CSKB in comparison with popular existing resources. For comparability, we report statistics on a sample of 50 popular animals and 50 popular occupations introduced in [46], in addition to 50 popular concepts in the engineering domain collected using Wiktionary word frequencies (e.g., car, bus, computer, phone, etc.). For statistics, subgroups are collected through *hyponyms* (WordNet) and relation *IsA* (ConceptNet and TupleKB). Aspects are collected via *part meronyms* (WordNet),

Subject type	#s	#spo	#facets
Primary	10,000	3,693,990	2,169,119
Subgroup	280,970	1,768,538	944,124
Aspect	92,038	3,349,198	1,467,159
All	382,555	8,562,593	4,425,628

Table 2: Statistics of Ascent KB.

Resource	#s	#spo	#facets	#subgroups	#aspects
WordNet [34]	150	-	-	1,472	229
WebChild [55]	150	178,073	-	-	47,171
ConceptNet [53]	150	7,313	-	7,239	368
TupleKB [35]	133	23,106	-	231	2,302
Quasimodo [46]	150	137,880	-	-	563
GenericsKB [1]	150	192,075	-	-	-
Ascent	150	132,070	80,717	10,026	5,843
Ascent^{sg}	8,251	110,631	64,449	-	-
Ascent^{asp}	5,618	169,770	74,449	-	-

Table 3: Statistics of different resources on top 50 subjects for three domains: animals, occupations, engineering.

relation *PartOf* (ConceptNet), *hasPart* (TupleKB), *hasPhysicalPart* (WebChild) and *hasBodyPart* (Quasimodo). We divide the statistics of our KB into three categories: general assertions (Ascent), subgroup assertions (Ascent^{sg}) and aspect assertions (Ascent^{asp}). Table 3 shows that Ascent, among all resources, is the only one which conveys qualitative facets besides triples. Ascent also extracts a considerable amount of assertions for the primary subjects. In addition, Ascent has the capability to extend the 150 primary subjects to 13,869 subgroups and related aspects, approximately tripling the number of the extracted assertions. We extract more subgroups than any other KB. Regarding aspects we are only outperformed by WebChild, which includes many uninformative and rather "exotic" part-of triples (e.g., teacher has cell, lion has facial vein).

5.2 Intrinsic Evaluation

To investigate RQ1, we instantiate *quality* with the standard notions of precision and recall, splitting precision further up into the dimensions of *typicality* and *salience*, measuring this way the degree of truth, and the degree of relevance of assertions (cf. [46]). Typicality states that an assertion holds for most instances of a concept. For example, elephants using their trunk is typical, whereas elephants drinking milk holds only for baby elephants. Salience refers to the human perspective of whether an assertion is associated with a concept by most humans more or less on first thought. For example, elephants having trunks is salient, whereas elephants killing their mahouts (trainers) is not.

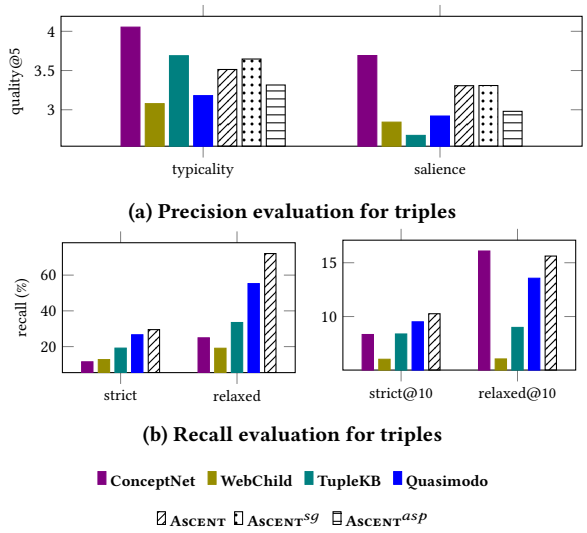
Assertion precision. Unlike for encyclopedic knowledge ("The Lion King" was either produced by Disney, or it wasn't), precision of CSK is generally not a binary concept, calling for more refined evaluation metrics. We follow the Quasimodo project [46] which assessed typicality and salience. Given a CSK triple, annotators on Amazon MTurk are asked to evaluate each of the two aspects on a scale from 1 (lowest) to 5 (highest). We use the same sampling setup as proposed in [46]: for each KB (i.e., Ascent and the prior CSKBs), create a pool that contains the 5 top-ranked triples of each of a selected set of subjects, then randomly sample 50 triples from this pool. In addition, specifically for our KB, we create a pool from top-5 ranked subgroup assertions of each subject, then also draw 50 random triples from the pool for evaluation, which is reported as Ascent^{sg}. The same sampling process is applied for aspect assertions in our KB, which is reported as Ascent^{asp}. Each triple is evaluated by three different crowd-workers. We iteratively evaluate triple quality for three sets of 50 subjects of three domains: animals, occupations and engineering, respectively. We report the aggregated results in Fig. 3a. Among the automatically-constructed KBs (i.e., except for ConceptNet), our KB has the most salient assertions while demonstrating competitive quality when it comes to typicality. These results indicate that our source selection, filtering and extraction scheme allows to pull out important assertions better than other CSKBs.

Assertion recall. Evaluating recall requires a notion of ground truth. For this purpose, we use crowdsourcing-based phrases from humans collected by Quasimodo [46]: 2,400 free association sentences for 50 occupations and 50 animals. We also evaluate using the same metrics, strict and relaxed sentence-assertion match. In the *relaxed* mode, we measure the fraction of tokens, from the human-written phrase, that are contained in some KB triples for the corresponding subject. In the *strict* mode, we only consider statements where P, O or PO is exactly found in the human-written phrase, and measure the fraction of matching characters vs. the total length of the human-written phrase. To match natural language with KB predicates, we use generic translations (e.g., *hasProperty* → *is*, *hasPhysicalPart* → *has*, *is-part-of* → *is part of*, etc.). The evaluation results can be seen in Fig. 3b. We observe that Ascent captures a significantly higher fraction of the ground-truth assertions provided by crowd workers than any of the other CSKBs. When we limit CSKBs to their top-10 ranked triples for each subject, Ascent outperforms all other KBs in the strict mode and is the second-best after ConceptNet, which is the only one that was constructed manually, in the relaxed mode. This result affirms that our top-ranked assertions have high quality compared to other CSKBs.

Subgroups and aspects. We compare Ascent *subgroup* entries to the manually created ConceptNet, and against a comprehensive taxonomy, WebIsALOD [22], automatically built by applying 58 Hearst-style extraction patterns to the Common Crawl corpus. For a random sample of 500 subgroup entries per resource, we manually found an average precision of 5.6% for WebIsALOD, 83.4% for ConceptNet, and 92.0% for Ascent (note that we manually filtered out instances in WebIsALOD's entries). Our precision significantly outperforms WebIsALOD, and is even better than the manually constructed ConceptNet. At the same time, it is worth to point out that our approach misses out on subgroups that do not lexically contain the main subject, e.g., "panda" as subgroup of "bear".

We compare *aspects* against two resources: hasPartKB [2] and predictions made by masked language models (LMs). As neural-embedding LM, we use RoBERTa-Large and follow the idea of [57] to ask the LM to predict the missing word in the sentence "Everyone knows that <subject> has <?>." We use the human-generated CSLB concept property norm dataset [9] as ground truth, retaining only

[8]



(a) Precision evaluation for triples

(b) Recall evaluation for triples

Figure 3: Precision and recall assessment of different CSKBs.

headwords to allow a fair comparison with the masked prediction that produces only a single token. Since Ascent contains a wider range of aspects than just physical parts as in hasPartKB and the CSLB dataset, we use recall@k as the metrics for this evaluation, focusing on the top-5 terms from CSLB. Considering the top-5, top-10 and top-20 assertions per KB/LM, Ascent achieves recall@5 of 0.27, 0.41, 0.53, compared with hasPartKB at 0.13, 0.22, 0.35, and RoBERTa-Large at 0.29, 0.41, 0.51. Thus, Ascent considerably outperforms hasPartKB in this setup, and performs on par with state-of-the-art language models.

5.3 Extrinsic Evaluation

To answer RQ2, we conduct a comprehensive evaluation of the contribution of commonsense knowledge to question answering (QA) via four different setups, all based on the idea of priming pre-trained LMs with context [21, 42]:

(1) In *masked prediction* (MP) [43], we ask language models to predict single tokens in generic sentences.
(2) In *free generation* (FG), we provide only questions, and let LMs generate arbitrary answer sentences.
(3) In *guided generation* (GG), LMs are provided with an answer sentence prefix. This provides a middle ground between the previous two setups, allowing multi-token answers, but avoiding some overly evasive answers.
(4) In *span prediction* (SP), LMs select best answers from provided content [25].

We illustrate all settings in Table 4. In all settings, LMs are provided with context in the form of assertions taken from either ConceptNet, TupleKB, Quasimodo, GenericsKB or Ascent. These setups are motivated by the observation that priming language models with context can significantly influence their predictions [21, 42]. Previous works on language model priming mostly focused on evaluating retrieval strategies. In contrast, our comprehensive test suite

Setting	Input	Sample output
MP	Elephants eat [MASK]. [SEP] Elephants eat roots, grasses, fruit, and bark, and they eat a lot of these things.	everything (15.52%), trees (15.32%), plants (11.26%)
FG	C: Elephants eat roots, grasses, fruit, and bark, and they eat a lot of these things. Q: What do elephants eat? A:	They eat a lot of grasses, fruits, and trees.
GG	C: Elephants eat roots, grasses, fruit, and bark, and they eat a lot of these things. Q: What do elephants eat? A: Elephants eat	Elephants eat a lot of things.
SP	question="What do elephants eat?" context="Elephants eat roots, grasses, fruit, and bark, and they eat a lot of these things."	start=14, end=46, answer="roots, grasses, fruit, and bark"

Table 4: Examples of 4 QA settings (MP - masked prediction, FG - free generation, GG - guided generation, SP - span prediction). Sample output was given by RoBERTa (for MP), GPT-2 (for FG and GG) and ALBERT (for SP).

focuses on the impact of utilizing different CSK resources, while leaving the retrieval component constant.

Masked prediction is perhaps the best researched problem, coming with the advantage of allowing automated evaluation, although automated evaluation may unfairly discount sensible alternative answers. Also, masked prediction is limited to single tokens. Free generations circumvent this restriction, although they necessitate human annotations, and are prone to evasive answers. They are thus well complemented by extractive answering schemes, which limit the language models abstraction abilities, but provide the cleanest way to evaluate the context alone.

Models. Following standard usage, we use RoBERTa-Large for masked prediction, the autoregressive GPT-2 for the two generative setups, and ALBERT-xxlarge [25], fine-tuned on SQuAD 2.0 for span prediction.

Context retrieval method. Given a query, we use a simple token overlapping method to pull out relevant assertions from a CSKB. First, we only take into account assertions whose subjects are mentioned in the query. We rank these assertions by the number of distinct tokens occurring in the input query (ignoring stop words). For each query, we pick up the top ranked assertions and concatenate them to build the context. For comparability, we limit the length of every context to 256 characters. As rank tie-breaker, we use original ranks in the CSKBs.

Task construction. Previous work has generated masked sentences based on templates from ConceptNet triples [43]. However, the resulting sentences are often unnatural, following the idiosyncrasies of the ConceptNet data model. We therefore built a new dataset of natural commonsense sentences for *masked prediction*. We use the CSLB property norm dataset [9] which consists of short human-written sentences about salient properties of general concepts. We hide the last token of each sentence, which is usually the object of that sentence. Besides, we remove sentences that contain less than three words. The resulting dataset consists of 19,649 masked sentences.

For the *generative and extractive settings*, we use the Google Search Auto-completion functionality to collect commonsense questions about the aforementioned set of 150 engineering concepts, animals and occupations. For each subject, we feed the API with 6 prefixes: "what/when/where are/do <subject>", then we collect

Context	FG		GG		SP		MP
	C	I	C	I	C	I	P@5
No context	2.44	2.22	2.87	2.57	-	-	17.9
ConceptNet	2.74	2.39	3.03	2.61	2.34	2.16	24.5
TupleKB	2.84	2.53	**3.46**	**3.03**	1.82	1.62	23.7
Quasimodo	2.58	2.31	3.06	2.72	2.22	2.05	25.1
GenericsKB-Best	2.89	**2.71**	3.13	2.77	2.39	2.20	24.8
ASCENT^{tri}	**2.91**	2.68	3.41	3.01	**2.61**	**2.34**	**25.9**

Table 5: Results of our QA evaluation. Metrics: C - correctness, I - informativeness, P@5 - precision at five (%). ASCENT^{tri} contains only triples in ASCENT.

all auto-completed queries returned by the API. We got 8,098 auto-completed queries for these subjects. Next, we drew samples from that query set, then manually removed jokes and other noise (e.g., "where do cows go for entertainment") obtaining 50 questions for evaluation. The answers from each KB in each generative or extractive setting were then posted on Amazon MTurk, along with test questions that ensured answer quality.

Evaluation scheme. For commonsense topics, questions often have multiple valid answers. Additionally, given that answers in our settings of generative and extractive QA are very open, creating an automated evaluation is difficult. We therefore use human judgements for evaluating all settings except masked prediction. Specifically, given a question and set of answers, we ask humans to assess each answer based on two dimensions, *correctness* and *informativeness*, on a scale from 1 (lowest) to 5 (highest). Each question is evaluated by three annotators in Amazon MTurk. For evaluating masked prediction, we use the mean precision at k ($P@k$) metric, following [43].

Results. The evaluation results are shown in Table 5. We can see that all KBs contribute contexts that improve LM response quality. ASCENT performs significantly better than the no-context baseline in both FG, GG and MP settings (p-values of paired t-test below 0.013), Besides, in the span prediction (SP) setting, where answers come directly from retrieved contexts, ASCENT outperforms all competitors, indicating that our assertions have very high quality compared to other KBs – with statistically significant gains (p-value below 0.038) over TupleKB on both metrics, and Quasimodo on correctness. Notably, our structured resource also outperforms the text-based GenericsKB in all but one case. For the MTurk assessments, we obtained a mean score variance of 0.76 and a mean Pearson correlation coefficient of 0.58, which indicate high agreement among annotators. We demonstrate three examples for the retrieved contexts and answers generated by GPT-2 in Table 6.

	Question + Retrieved contexts	Answer
	WHEN ARE RATS AWAKE?	
∅	-	The rats are awake.
CN	Awakes have yawn. Awakes need open eyes. Awakes need sleep. Awaking has rub eyes. Awakes need set alarm clock. Ratss are rodents...	Rats are awake.
TP	Rats have bone. Rats have brain. Rats have cell. Rats have chest cavity. Rats have chest. Rats have cytoplasm. Rats have ear.	Rats are awake.
QS	Rats carry rabies. Rats can jump. Rats get electrocuted. Rats get rabies. Rats are pet. Rats have gallbladder. Rats have bladder...	Rats are awake.
GB	Rats are omnivorous, which means they eat both plant and animal material. Rats are nocturnal animals. Rats are monophyodont, whi...	Rats are awake at night.
AT	Rats are intelligent. Rats are clean. Rats are nocturnal. Rats are active. Rats are careful. Rats grind their teeth. Rats are om...	Rats are awake.
AQ	Rats are very intelligent. Rats are very clean. Rats are mainly nocturnal. Rats are active at night. Rats are very careful. Rats...	Rats are awake at night.
	WHERE DO SHEPHERDS LIVE?	
∅	-	They live in the woods.
CN	Lives have die. Lives have breathe. Lives need born. Living causes dying. Living causes death. Living causes life. Lives have bo...	They live in the wilds of the North.
TP	Shepherds have arm. Shepherds have body substance. Shepherds have cell membrane. Shepherds have cell. Shepherds have corpus. She...	Shepherds live in the wild.
QS	Shepherds can live. Noes can live. Living is thing. Lives are illegal in uk. Living is alone. Living is abroad. Living is good.	Shepherds live in the wild.
GB	Shepherds care for their sheep but also expect to make a living. Shepherds require proteins, carbohydrates, minerals, vitamins a...	Shepherds live in the wild, but they are not allowed to breed in the wild.
AT	Shepherds live in crop fields. Shepherds care for the sheep. Shepherds are the flock of god. Being is living. Shepherds keep the...	Shepherds live in crop fields.
AQ	Shepherds live in crop fields. Shepherds care for the sheep to green pastures. Shepherds are the flock of god. Being is living f...	Shepherds live in crop fields.
	WHAT ARE CARS USED FOR?	
∅	-	The most common car used for driving is a car with a front end that is not a front end.
CN	Cars are used for fun. Cars are used for getting to work. Carss can use fuel. Cars are used for drive. Cars are used for transpo...	Cars are used for fun.
TP	Cars use chemical energy. Cars use energy. Cars use gasoline. Cars use suspension. Cars use independent suspension. Cars use pis...	Cars used for: 1. A car that is used for: 1. A car that is used for: ...
QS	Cars use gas. Cars use diesel. Cars use fuel. Cars use circuit breakers. Cars use transmission. Cars use engines. Cars use propa...	Cars used for the purpose of transporting goods.
GB	Cars are complex devices used in a complex environment. Cars use up energy and cause pollution. Cars use up energy and cause pol...	Cars are complex devices used in a complex environment.
AT	Cars are used for transportation. Cars use parallel washington street. Cars use part. Cars use lever. Cars use less manpower. Ts...	Cars are used for transportation.
AQ	Cars are used for transportation. Cars use parallel washington street to a crest at hyde street. Cars use part by drew winter an...	Cars are used for transportation.

Table 6: Examples of retrieved KB assertions and answers generated by GPT-2. Abbreviations: No context (∅), ConceptNet (CN), TupleKB (TP), Quasimodo (QS), GenericsKB-Best (GB), ASCENT^{tri} (AT), ASCENT^{quad} (AQ). AT contains only triples, while in AQ the most frequent facet in every triple is involved.

5.4 Evaluation of Facets

To answer RQ3, we evaluate facets both intrinsically and extrinsically.

For *intrinsic evaluation*, as there are no existing CSKBs with facets, we provide comparisons with two baselines, a random permutation of facet values in ASCENT, and facets generated by GPT-2. First, we randomly drew 100 assertions with facets from our KB. Next, we translate each statement into a sentence prefix and ask GPT-2 to fill in the remaining words to complete the sentence. For example, given the quadruple ⟨*elephant, use, their trunks, PURPOSE: to suck up water*⟩, the sentence prefix will be "Elephants use their trunk to" and for this, GPT-2's continuation is "to move around" (see also Table 7 for more examples of ASCENT vs. GPT-generated

No.	Prefix	Ascent	GPT-2
1	Lawyers represent clients in	courts [location]	the case
2	Elephants use their trunks to	suck up water [purpose]	move around
3	Artificial intelligence has a number of applications in	today's society [location]	the field of artificial intelligence
4	Waiters deliver food to	a table [trans-obj]	the homeless in the city of San Francisco
5	Hogs roll in mud to	keep cool [purpose]	the ground
6	Wine is high in	alcohol [other-qty.]	the mix

Table 7: Examples of Ascent's facet types and values along with predictions of GPT-2 given sentence prefixes.

	Correctness	Informativeness
Random	1.47	1.29
GPT-2	2.85	2.22
Ascent	**3.99**	**3.50**

Table 8: Assessment of Ascent and LM-generated facets.

facets). We show each sentence prefix along with three answers (from Ascent, GPT-2 and random permutation) to crowd workers and ask them to evaluate each answer along two dimensions: *correctness* and *informativeness*, based on a scale from 1 (lowest) to 5 (highest). Each statement is assessed by three annotators. The evaluation results are reported in Table 8. Ascent outperforms the baselines by a large margin, indicating that the facets provide valuable information to better understand the assertions. For the MTurk assessments, we obtained a mean variance score of 0.77 and a mean Pearson correlation coefficient of 0.63, indicating good agreement between annotators.

For *extrinsic evaluation*, we reused the four question answering tasks from Section 5.3. We incorporated facets in the context in two ways: Once based on a 256-character limit (so adding facets means that in total, fewer statements can be given as context), once by expanding the top-5-ranked statements with their facets. Note that the sets of questions in each case were different, so the absolute scores are not directly comparable. The results are shown in Table 9, and the insights are twofold. On the one hand, within the fixed character-limit setting, facets do not improve results, presumably because expanding statements by facets means that some statements relevant for question answering fall out of the size limit. On the other hand, expanding a fixed number of statements by facets gives a consistent improvement in three of the four evaluation settings (FG, GG, SP), with the biggest effect being observed for informativeness in the least constrained setting (11% relative improvement in informativeness in free generation). An example where facets are crucial is shown in Table 6 with the query "When are rats awake?".

5.5 Per-module Evaluation

Open information extraction. We report the yield of our OIE method in comparison with StuffIE [45] and Graphene [4] in Table 10 on a sample dataset of Wikipedia articles for ten random concepts, consisting of 2,557 sentences. Nested facets (i.e., linked contexts in Graphene) are not considered. It can be seen that our extractor can identify significantly more assertions and facets than

Context	FG		GG		SP		MP
	C	I	C	I	C	I	P@5
256-character limit							
Ascent^{tri}	**2.91**	2.68	**3.41**	**3.01**	2.61	2.34	**25.9**
Ascent^{quad}	2.84	2.59	3.20	2.81	**2.68**	**2.44**	25.6
Top-5-statement limit							
Ascent^{tri}	2.73	2.26	2.91	2.41	2.20	1.89	**25.8**
Ascent^{quad}	**2.93**	**2.53**	**3.04**	**2.57**	**2.23**	**1.96**	25.5

Table 9: Extrinsic evaluation of facets by correctness (C) and informativeness (I).

Method	#spo	#facets	avg. length
StuffIE [45]	6,078	4,281	6.83
Graphene [4]	5,708	2,112	10.10
Ascent	**6,690**	**4,911**	**6.28**

Table 10: Yield statistics of different OIE methods.

Task	#train	#test	Acc.
Triple-pair classification	21,569	5,392	0.958
Facet type labeling	3,962	991	0.928

Table 11: Corpus accuracy statistics for two RoBERTa-based tasks.

the comparison systems. Besides, the conciseness of our output improves, as average assertion length without facets (measured in words) decreases.

RoBERTa-based tasks. We report the sizes of annotated corpora and performance of our two RoBERTa classification models in Table 11. Since these tasks are specific to our pipeline, there are no external baselines to be compared. For both tasks, we use the pretrained RoBERTa-Base model for initialization and other specifications as follows: Adam optimizer with learning rate of 2×10^{-5} and Adam epsilon of 10^{-8}; batch size of 32; and maximal sequence length of 32. We train the model for 10 epochs for the facet labeling task, and 4 epochs for the triple pair classification task. Both models obtain very high accuracy.

6 CONCLUSION

This paper presented Ascent, a methodology to collect advanced commonsense knowledge about generic concepts. Our refined knowledge representation allowed us to identify considerably more informative assertions, and avoid common limitations of previous works. The technique for generating web search queries and filtering results shows that CSK extraction from general web content is feasible with high precision and recall. Intrinsic and extrinsic evaluations confirmed that the resulting CSKB is a significant advance over existing resources.

We hope that our approach revives the long-standing vision of structured CSKBs [26] and provides a cutting-edge resource that can drive forward knowledge-centric AI applications. Code, data, and a web interface are available at `https://www.mpi-inf.mpg.de/ascent`.

ACKNOWLEDGMENTS

We are thankful to Kyle Richardson for suggestions on extrinsic comparisons of CSKBs.

REFERENCES

[1] Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. GenericsKB: A Knowledge Base of Generic Statements. *arXiv:2005.00660* (2020).
[2] Sumithra Bhakthavatsalam, Kyle Richardson, Niket Tandon, and Peter Clark. 2020. Do Dogs have Whiskers? A New Knowledge Base of hasPart Relations. *arXiv:2006.07510* (2020).
[3] Tom B. Brown et al. 2020. Language Models are Few-Shot Learners. *arXiv:2005.14165* (2020).
[4] Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018. Graphene: Semantically-linked propositions in open information extraction. *COLING* (2018).
[5] Yohan Chalier, Simon Razniewski, and Gerhard Weikum. 2020. Joint Reasoning for Multi-Faceted Commonsense Knowledge. *AKBC* (2020).
[6] Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain Natural Language Inference. In *ACL*.
[7] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv:1803.05457* (2018).
[8] James Clarke, Vivek Srikumar, Mark Sammons, and Dan Roth. 2012. An NLP Curator (or: How I Learned to Stop Worrying and Love NLP Pipelines). In *LREC*.
[9] Barry J Devereux, Lorraine K Tyler, Jeroen Geertzen, and Billi Randall. 2014. The Centre for Speech, Language and the Brain (CSLB) concept property norms. *Behavior research methods* (2014).
[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL* (2019).
[11] Mauro Dragoni, Soujanya Poria, and Erik Cambria. 2018. OntoSenticNet: A Commonsense Ontology for Sentiment Analysis. *IEEE Intell. Syst.* (2018).
[12] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. *ACL* (2019).
[13] Oren Etzioni, Michael J. Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2004. Web-scale information extraction in Knowitall: (preliminary results). In *WWW*.
[14] Edward A Feigenbaum. 1984. Knowledge engineering. *Annals of the New York Academy of Sciences* (1984).
[15] Samuel Fernando and Mark Stevenson. 2012. Mapping WordNet synsets to Wikipedia articles.. In *LREC*. 590–596.
[16] Dov M. Gabbay. 2003. *Many-Dimensional Modal Logics: Theory and Applications*. Elsevier North Holland.
[17] Roxana Girju, Adriana Badulescu, and Dan I. Moldovan. 2006. Automatic Discovery of Part-Whole Relations. *Comput. Linguistics* (2006).
[18] Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *AKBC*, Fabian M. Suchanek, Sebastian Riedel, Sameer Singh, and Partha Pratim Talukdar (Eds.).
[19] Jonathan Gordon, Benjamin Van Durme, and Lenhart K. Schubert. 2010. Learning from the Web: Extracting General World Knowledge from Noisy Text. *AAAI Workshops* (2010).
[20] Jonathan Gordon and Lenhart K. Schubert. 2010. Quantificational Sharpening of Commonsense Knowledge. In *Commonsense Knowledge, AAAI Fall Symposium*.
[21] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *ICML* (2020).
[22] Sven Hertling and Heiko Paulheim. 2017. WebIsALOD: Providing Hypernymy Relations Extracted from the Web as Linked Open Data. In *ISWC*.
[23] Aidan Hogan et al. 2020. Knowledge Graphs. *ArXiv* (2020).
[24] Filip Ilievski, Pedro Szekely, and Daniel Schwabe. 2020. Commonsense Knowledge in Wikidata. *Wikidata workshop* (2020).
[25] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite Bert for self-supervised learning of language representations. *ICLR* (2020).
[26] Douglas B Lenat. 1995. CYC: A large-scale investment in knowledge infrastructure. *CACM* (1995).
[27] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *EMNLP*.
[28] Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *EMNLP*.
[29] Bang Liu, Weidong Guo, Di Niu, Jinwen Luo, Chaoyue Wang, Zhen Wen, and Yu Xu. 2020. GIANT: Scalable Creation of a Web-scale Ontology. In *SIGMOD*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.).
[30] Hugo Liu and Push Singh. 2004. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal* (2004).
[31] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692* (2019).
[32] Mausam. 2016. Open Information Extraction Systems and Downstream Applications. In *IJCAI*.
[33] John McCarthy. 1960. *Programs with common sense*. RLE and MIT computation center.
[34] George A. Miller. 1995. WordNet: A Lexical Database for English. *CACM* (1995).
[35] Bhavana Dalvi Mishra, Niket Tandon, and Peter Clark. 2017. Domain-targeted, high precision knowledge extraction. *TACL* (2017).
[36] Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.* (2009).
[37] Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence* 193 (2012), 217–250.
[38] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. *COLING* (2018).
[39] Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic Role Labeling*. Morgan & Claypool Publishers. https://doi.org/10.2200/S00239ED1V01Y200912HLT006
[40] Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging Generic Patterns for Automatically Harvesting Semantic Relations. In *ACL*.
[41] Marius Pasca and Benjamin Van Durme. 2008. Weakly-Supervised Acquisition of Open-Domain Classes and Class Attributes from Web Documents and Query Logs. In *ACL*.
[42] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How Context Affects Language Models' Factual Predictions. *AKBC* (2020).
[43] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *EMNLP* (2019).
[44] Simone Paolo Ponzetto and Michael Strube. 2011. Taxonomy induction based on a collaboratively built knowledge repository. *Artif. Intell.* (2011).
[45] Radityo Eko Prasojo, Mouna Kacimi, and Werner Nutt. 2018. StuffIE: Semantic tagging of unlabeled facets using fine-grained information extraction. In *CIKM*.
[46] Julien Romero, Simon Razniewski, Koninika Pal, Jeff Z. Pan, Archit Sakhadeo, and Gerhard Weikum. 2019. Commonsense Properties from Query Logs and Question Answering Forums. *CIKM* (2019).
[47] Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2018. Atomic: An atlas of machine commonsense for if-then reasoning. *AAAI* (2018).
[48] Lenhart Schubert. 2002. Can we derive general world knowledge from texts. In *HLT*.
[49] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. 2016. A Large DataBase of Hypernymy Relations Extracted from the Web. In *LREC*.
[50] Vered Shwartz and Chris Waterson. 2018. Olive oil is made of olives, baby oil is made for babies: Interpreting noun compounds using paraphrases in a neural model. *NAACL* (2018).
[51] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences*.
[52] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic Taxonomy Induction from Heterogenous Evidence. In *ACL*.
[53] Robyn Speer and Catherine Havasi. 2012. ConceptNet 5: A large semantic network for relational knowledge. In *Theory and Applications of Natural Language Processing*.
[54] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *NAACL*.
[55] Niket Tandon, Gerard de Melo, Fabian M. Suchanek, and Gerhard Weikum. 2014. WebChild: harvesting and organizing commonsense knowledge from the web. In *WSDM*.
[56] Niket Tandon, Charles Hariman, Jacopo Urbani, Anna Rohrbach, Marcus Rohrbach, and Gerhard Weikum. 2016. Commonsense in parts: Mining part-whole relations from the web and image tags. In *AAAI*.
[57] Nathaniel Weir, Adam Poliak, and Benjamin Van Durme. 2020. Probing Neural Language Models for Human Tacit Assumptions. *CogSci* (2020).
[58] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. 2012. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD*.
[59] Jiangnan Xia, Chen Wu, and Ming Yan. 2019. Incorporating Relation Knowledge into Commonsense Reading Comprehension with Multi-task Learning. In *CIKM*.
[60] Hongming Zhang, Daniel Khashabi, Yangqiu Song, and Dan Roth. 2020. TransOMCS: From Linguistic Graphs to Commonsense Knowledge. In *IJCAI*.
[61] Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *TACL* (2017).