



HAL
open science

Commonsense Properties from Query Logs and Question Answering Forums

Julien Romero, Simon Razniewski, Gerhard Weikum, Jeff Z. Pan, Archit Sakhadeo, Koninika Pal

► To cite this version:

Julien Romero, Simon Razniewski, Gerhard Weikum, Jeff Z. Pan, Archit Sakhadeo, et al.. Commonsense Properties from Query Logs and Question Answering Forums. Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, 10.1145/3357384.3357955 . hal-02158602v1

HAL Id: hal-02158602

<https://telecom-paris.hal.science/hal-02158602v1>

Submitted on 18 Jun 2019 (v1), last revised 8 Sep 2019 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Commonsense Properties from Query Logs and Question Answering Forums

Julien Romero
Télécom ParisTech
romero@telecom-paristech.fr

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

Koninika Pal
Max Planck Institute for Informatics
kpal@mpi-inf.mpg.de

Jeff Z. Pan
University of Aberdeen
jeff.z.pan@abdn.ac.uk

Archit Sakhadeo
Max Planck Institute for Informatics
architsakhadeo@gmail.com

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

ABSTRACT

Commonsense knowledge about object properties, human behavior and general concepts is crucial for robust AI applications. However, automatic acquisition of this knowledge is challenging because of sparseness and bias in online sources. This paper presents Quasimodo, a methodology and tool suite for distilling commonsense properties from non-standard web sources. We devise novel ways of tapping into search-engine query logs and QA forums, and combining the resulting candidate assertions with statistical cues from encyclopedias, books and image tags in a corroboration step. Unlike prior work on commonsense knowledge bases, Quasimodo focuses on salient properties that are typically associated with certain objects or concepts. Extensive evaluations, including extrinsic use-case studies, show that Quasimodo provides better coverage than state-of-the-art baselines with comparable quality.

1 INTRODUCTION

1.1 Motivation and Goal

Commonsense knowledge (CSK for short) is an old theme in AI, already envisioned by McCarthy in the 1960s [16] and later pursued by AI pioneers like Feigenbaum [9] and Lenat [12]. The goal is to equip machines with knowledge of properties of everyday objects (e.g., bananas are yellow, edible and sweet), typical human behavior and emotions (e.g., children like bananas, children learn at school, death causes sadness) and general plausibility invariants (e.g., a classroom of children should also have a teacher). In recent years, research on automatic acquisition of such knowledge has been revived, driven by the pressing need for human-like AI systems with robust and explainable behavior. Important use cases of CSK include the interpretation of user intents in search-engine queries, question answering, versatile chatbots, language comprehension, visual content understanding, and more.

Examples: A keyword query such as “Jordan weather forecast” is ambiguous, but CSK should tell the search engine that this refers to the country and not to a basketball player or machine learning professor. A chatbot should know that racist jokes are considered tasteless and would offend its users; so CSK could have avoided the 2016 PR disaster of the Tay chatbot.¹ In an image of a meeting at an IT company where one person wears a suit and another person is in jeans and t-shirt, the former is likely a manager and the latter an engineer. Last but not least, a “deep fake” video where Donald

Trump rides on the back of a tiger could be easily uncovered by knowing that tigers are wild and dangerous and, if at all, only circus artists would do this.

The goal of this paper is to advance the automatic acquisition of salient commonsense properties from online content of the Internet. For knowledge representation, we focus on simple assertions in the form of subject-predicate-object (SPO) triples such as `children like banana` or `classroom includes teacher`. Complex assertions, such as Datalog clauses, and logical reasoning over these are outside our scope.

A major difficulty that prior work has struggled with is the sparseness and bias of possible input sources. Commonsense properties are so mundane that they are rarely expressed in explicit terms (e.g., countries or regions have weather, people don’t). Therefore, typical sources for information extraction like Wikipedia are fairly useless for CSK. Moreover, online contents, like social media (Twitter, Reddit, Quora etc.), fan communities (Wikia etc.) and books or movies, are often heavily biased and do not reflect typical real-life situations. For example, existing CSK repositories contain odd triples such as `banana located_in monkey’s_hand`, `engineer has_property conservative`, `child make choice`.

1.2 State of the Art and Limitations

Popular knowledge bases like DBpedia, Wikidata or Yago have a strong focus on encyclopedic knowledge about individual entities like (prominent) people, places etc., and do not cover commonsense properties of general concepts. The notable exception is the inclusion of SPO triples for the (sub-)type (aka. isa) predicate, for example, `banana type fruit`. Such triples are ample especially in Yago (derived from Wikipedia categories and imported from WordNet). Our focus is on additional properties beyond type, which are absent in all of the above knowledge bases.

The most notable projects on constructing commonsense knowledge bases are Cyc [12], ConceptNet [28], WebChild [34] and Mosaic TupleKB [7]. Each of these has specific strengths and limitations. The seminal Cyc project solely relied on human experts for codifying logical assertions, with inherent limitations in scope and scale. ConceptNet used crowdsourcing for scalability and better coverage, but is limited to only a few different predicates like `has_property`, `located_in`, `used_for`, `capable_of`, `has_part` and `type`. Moreover, the crowdsourced inputs often take noisy, verbose or uninformative forms (e.g., `banana type bunch`, `banana type herb`, `banana has_property good_to_eat`). WebChild tapped into book ngrams and image tags to overcome the bias in many Web sources.

¹www.cnbc.com/2018/03/17/facebook-and-youtube-should-learn-from-microsoft-tay-racist-chatbot.html

It has a wider variety of 20 predicates and is much larger, but contains a heavy tail of noisy and dubious triples – due to its focus on possible properties rather than typical ones (e.g., engineers are conservative, cool, qualified, hard, vital etc.). TupleKB is built by carefully generating search-engine queries on specific domains and performing various stages of information extraction and cleaning on the query results. Despite its clustering-based cleaning steps, it contains substantial noise and is limited in scope by the way the queries are formulated.

The work in this paper aims to overcome the bottlenecks of these prior projects while preserving their positive characteristics. In particular, we aim to achieve high coverage, like WebChild, with high precision (i.e., a fraction of valid triples), like ConceptNet. In addition, we strive to acquire properties for a wide range of predicates - more diverse and refined than ConceptNet and WebChild, but without the noise that TupleKB has acquired.

1.3 Approach and Challenges

This paper puts forward *Quasimodo*², a framework and tool for scalable automatic acquisition of commonsense properties. Quasimodo is designed to tap into non-standard sources where questions rather than statements provide cues about commonsense properties. This leads to noisy candidates for populating a commonsense knowledge base (CSKB). To eliminate false positives, we have devised a subsequent cleaning stage, where corroboration signals are obtained from a variety of sources and combined by learning a regression model. This way, Quasimodo reconciles wide coverage with high precision. In doing this, it focuses on salient properties which typically occur for common concepts, while eliminating possible but atypical and uninformative output. This counters the reporting bias - frequent mentioning of sensational but unusual and unrealistic properties (e.g., pink elephants in Walt Disney’s *Dumbo*).

The new sources that we tap into for gathering candidate assertions are search-engine query logs and question answering forums like Reddit, Quora etc. Query logs are unavailable outside industrial labs, but can be sampled by using search-engine interfaces in a creative way. To this end, Quasimodo generates queries in a judicious way and collects auto-completion suggestions. The subsequent corroboration stage harnesses statistics from search-engine answer snippets, Wikipedia editions, Google Books and image tags by means of a learned regression model. This step is geared to eliminate noisy, atypical, and uninformative properties.

A subsequent ranking step further enhances the knowledge quality in terms of typicality and saliency. Finally, to counter noisy language diversity, reduce semantic redundancy, and canonicalize the resulting commonsense triples to a large extent, Quasimodo includes a novel way of clustering the triples that result from the fusion step. This is based on a tri-factorization model for matrix decomposition.

Our approach faces two major challenges:

- coping with the heavy reporting bias in cues from query logs, potentially leading to atypical and odd properties,

²The name stands for: Query Logs and QA Forums for Salient Commonsense Definitions. Quasimodo is the main character in Victor Hugo’s novel “The Hunchback of Notre Dame” who epitomizes human preconception and also exhibits unexpected traits.

- coping with the noise, language diversity, and semantic redundancy in the output of information extraction methods.

The paper shows how these challenges can be (largely) overcome. Experiments demonstrate the practical viability of Quasimodo and its improvements over prior works.

1.4 Contributions

The paper makes the following original contributions:

- a complete methodology and tool for multi-source acquisition of typical and salient commonsense properties with principled methods for corroboration, ranking and refined grouping,
- novel ways of tapping into non-standard input sources like query logs and QA forums,
- a high-quality knowledge base of ca. 2.21 million salient properties for ca. 52,000 concepts, which will be made publicly available as a research resource³,
- an experimental evaluation and comparison to ConceptNet, WebChild, and TupleKB which shows major gains in coverage and quality, and
- experiments on extrinsic tasks like language games (Taboo word guessing) and question answering.

Our code will be made available on Github.

2 RELATED WORK

Commonsense Knowledge Bases (CSKB’s). The most notable projects on building large commonsense knowledge bases are the following.

Cyc: The Cyc project was the first major effort towards collecting and formalizing general world knowledge [12]. Knowledge engineers manually compiled knowledge, in the form of grounded assertions and logical rules. Parts of Cyc were released to the public as OpenCyc in 2002, but these parts mostly focus on concept taxonomies, that is, the (sub-)type predicate.

ConceptNet: Crowdsourcing has been used to construct ConceptNet, a triple-based semantic network of commonsense assertions about general objects [28, 29]. ConceptNet contains ca. 1.3 million assertions for ca. 850,000 subjects (counting only English assertions and semantic relations, i.e., discounting relations like *synonym* or *derivedFrom*). The focus is on a small number of broad-coverage predicates, namely, *type*, *locationOf*, *usedFor*, *capableOf*, *hasPart*. ConceptNet is one of the highest-quality and most widely used CSK resources.

WebChild: WebChild has been automatically constructed from book n-grams (and, to a smaller degree, image tags) by a pipeline of information extraction, statistical learning and constraint reasoning methods [33, 34]. WebChild contains ca. 13 million assertions, and covers 20 distinct predicates such as *hasSize*, *hasShape*, *physicalPartOf*, *memberOf*, etc. It is the biggest of the publicly available commonsense knowledge bases, with the largest slice being on part-whole knowledge [35]. However, a large mass of WebChild’s contents is in the long tail of possible but not necessarily typical

³<https://www.dropbox.com/sh/r1os5uo06v2xiac/AADinRFpUYsG1kQLm63pdMnOa?dl=0>

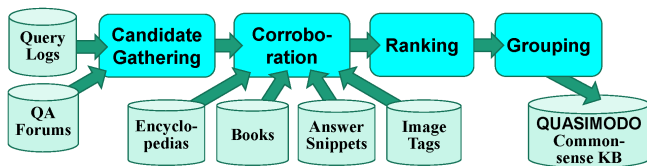


Figure 1: Quasimodo system overview.

and salient properties. So it comes with a substantial amount of noise and non-salient contents.

Mosaic TupleKB: The Mosaic project at AI2 aims to collect commonsense knowledge in various forms, from grounded triples to procedural knowledge with first-order logic. TupleKB, released as part of this ongoing project, is a collection of triples for the science domain, compiled by generating domain-specific queries and extracting assertions from the resulting web pages. A subsequent cleaning step, based on integer linear programming, clusters triples into groups. TupleKB contains ca. 280,000 triples for ca. 30,000 subjects.

Wikidata: This collaboratively built knowledge base is mostly geared to organize encyclopedic facts about individual entities like people, places, organizations etc. [13, 36]. It contains more than 400 million assertions for more than 50 million items. This includes some world knowledge about general concepts, like type triples, but this coverage is very limited. For instance, Wikidata neither knows that birds can fly nor that elephants have trunks.

Use Cases of CSK. Commonsense knowledge and reasoning are instrumental in a variety of applications in natural language processing, computer vision, and AI in general. These include question answering, especially for general world comprehension [42] and science questions [27]. Sometimes, these use cases also involve additional reasoning (e.g., [32]), where CSK contributes, too. Another NLP application is dialog systems and chatbots (e.g., [41]), where CSK adds plausibility priors to language generation.

For visual content understanding, such as object detection or caption generation for images and videos, CSK can contribute as an informed prior about spatial co-location derived, for example, from image tags, and about human activities and associated emotions (e.g., [5, 39, 40]). In such settings, CSK is an additional input to supervised deep-learning methods.

Information Extraction from Query Logs. Prior works have tapped into query logs for goals like query recommendation (e.g., [4]) and extracting semantic relationships between search terms, like synonymy and hypernymy/hyponymy (e.g., [1, 22, 24, 38]). The latter can be seen as gathering triples for CSK, but its sole focus is on the (sub-) type predicate – so the coverage of the predicate space is restricted to class/type taxonomies. Moreover, these projects were carried out on full query logs within industrial labs of search-engine companies. In contrast, Quasimodo addresses a much wider space of predicates and operates with an original way of sampling query-log-derived signals via auto-completion suggestions. To the best of our knowledge, no prior work has aimed to harness auto-completion for CSK acquisition (cf. [3]).

The methodologically closest work to ours is [23]. Like us, that work used interrogative patterns (e.g. “Why do ...”) to mine query logs – with full access to the search-engine company’s logs. Unlike us, subjects, typically classes/types such as “cars” or “actors”, were merely associated with salient phrases from the log rather than extracting complete triples. One can think of this as organizing CSK in SP pairs where P is a textual phrase that comprises both predicate and object but cannot separate these two. Moreover, [23] restricted itself to the extraction stage and used simple scoring from query frequencies, whereas we go further by leveraging multi-source signals in the corroboration stage and refining the SPO assertions into semantic groups.

3 SYSTEM OVERVIEW

Quasimodo is designed to cope with the high noise and potentially strong bias in online contents. It taps into query logs via auto-completion suggestions as a non-standard input source. However, frequent queries – which are the ones that are visible through auto-completion – are often about sensational and untypical issues. Therefore, Quasimodo combine a recall-oriented candidate gathering phase with two subsequent phases for cleaning, refining, and ranking assertions. Figure 1 gives a pictorial overview of the system architecture.

Candidate Gathering. In this phase, we extract candidate triples from some of the world’s largest sources of the “wisdom of crowds”, namely, search-engine query logs and question answering forums such as Reddit or Quora. While the latter can be directly accessed via search APIs, query logs are unavailable outside of industrial labs. Therefore, we creatively probe and sample this guarded resource by means of generating queries and observing auto-completion suggestions by the search engine. The resulting suggestions are typically among the statistically frequent queries. As auto-completion works only for short inputs of a few words, we generate queries that are centered on candidate subjects, the S argument in the SPO triples that we aim to harvest. Technical details are given in Section 4.

Corroboration. This phase is precision-oriented, aiming to eliminate false positives from the candidate gathering. We consider candidates as invalid for three possible reasons: 1) they do not make sense (e.g., programmers eat python); 2) they are not typical properties for the instances of the S concept (e.g., programmers drink espresso); 3) they are not salient in the sense that they are immediately associated with the S concept by most humans (e.g., programmers visit restaurants). To statistically check to which degree these aspects are satisfied, Quasimodo harnesses corroboration signals in a multi-source scoring step. This includes standard sources like Wikipedia articles and books, which were used in prior works already, but also non-standard sources like image tags and answer snippets from search-engine queries. Technical details are given in Section 5.

Ranking. To identify typical and salient triples, we devised a probabilistic ranking model with the corroboration scores as input signal. This stage is described in Section 6.

Grouping. For this phase, we have devised a clustering method based on the model of tri-factorization for matrix decomposition [8]. The output consists of groups of SO pairs and P phrases linked to each other. So we semantically organize and refine both the

concept arguments (S and O) in a commonsense triple and the way the predicate (P) is expressed in language. Ideally, this would canonicalize all three components, in analogy to what prior works have achieved for entity-centric encyclopedic knowledge bases (e.g., [10, 30]). However, commonsense assertions are rarely as crisp as facts about individual entities, and often carry subtle variation and linguistic diversity (e.g., *live in* and *roam in* for animals being near-synonymous but not quite the same). Our clustering method also brings out refinements of predicates. This is in contrast to prior work on CSK which has mostly restricted itself to a small number of coarse-grained predicates like *partOf*, *usedFor*, *locatedAt*, etc. Technical details are given in Section 7.

4 CANDIDATE GATHERING

The key idea for this phase is to utilize questions as a source of human commonsense. For example, the question “*Why do dogs bark?*” implicitly conveys the user’s knowledge that dogs bark. Questions of this kind are posed in QA forums, such as Reddit or Quora, but their frequency and coverage in these sources alone is not sufficient for building a comprehensive knowledge base. Therefore, we additionally tap into query logs from search engines, sampled through observing auto-completion suggestions. Although most queries merely consist of a few keywords, there is a substantial fraction of user requests in interrogative form [37].

4.1 Data Sources

Quasimodo exploits two data sources: (i) QA forums, which return questions in user posts through their search APIs, and (ii) query logs from major search engines, which are sampled by generating query prefixes and observing their auto-completions.

QA forums. We use four different QA forums: Quora, Yahoo! Answers⁴, Answers.com, and Reddit. The first three are online communities for general-purpose QA across many topics, and Reddit is a large discussion forum with a wide variety of topical categories.

Search engine logs Search engine logs are rich collections of questions. While logs themselves are not available outside of industrial labs, search engines allow us to glimpse at some of their underlying statistics by auto-completion suggestions. Figure 2 shows an example of this useful asset. Quasimodo utilizes Google and Bing, which typically return 5 to 10 suggestions for a given query prefix. In order to obtain more results, we recursively probe the search engine with increasingly longer prefixes that cover all letters of the alphabet, until the number of auto-completion suggestions drops below 5. For example, the query prefix “*why do cats*” is expanded into “*why do cats a*”, “*why do cats b*”, and so on.

We intentionally restrict ourselves to query prefixes in interrogative form, as these are best suited to convey commonsense knowledge. In contrast, simple keyword queries are often auto-completed with references to prominent entities (celebrities, sports teams, product names, etc.), given the dominance of such queries in the overall Internet (e.g., the query prefix “*cat*” is expanded into “*cat musical*”). These very frequent queries are not useful for CSK acquisition.

In total, we collected 11,603,121 questions from autocompletion.

⁴answers.yahoo.com and webscope.sandbox.yahoo.com

```

why do cats
why do cats purr
why do cats like boxes
why do cats meow
why do cats knead
why do cats sleep so much
why do cats hate water
why do cats like catnip
why do cats lick you
why do cats have whiskers

```

Figure 2: A glimpse into a search-engine query log.

| Pattern | In Query Logs | In QA Forums |
|-----------|---------------|--------------|
| how does | 19.4% | 7.5% |
| why is | 15.8% | 10.4% |
| how do | 14.9% | 38.07% |
| why do | 10.6% | 9.21% |
| how is | 10.1 % | 4.31% |
| why does | 8.97% | 5.46% |
| why are | 8.68% | 5.12% |
| how are | 5.51% | 1.8% |
| how can | 3.53% | 10.95% |
| why can't | 1.77% | 1.40% |
| why can | 0.81% | 0.36% |

Table 1: Question patterns for candidate gathering.

4.2 Question Patterns

We performed a quantitative analysis of frequent question words and patterns on Reddit. As a result, we decided to pursue two question words, *Why* and *How*, in combination with the verbs *is*, *do*, *are*, *does*, *can*, *can't*, resulting in 12 patterns in total. Their relative frequency in the question set that we gathered by auto-completion is shown in Table 1. For forums, we performed title searches centered around these patterns. For search engines, we appended subjects of interest to the patterns for query generation (e.g., “*Why do cats*”) for cats as subject. The subjects were chosen from the common nouns extracted from WordNet [17].

4.3 From Questions to Assertions

Open information extraction (Open IE) [15], based on patterns, has so far focused on assertive patterns applied to assertive sentences. In contrast, we deal with interrogative inputs, facing new challenges.

We address these issues by rule-based rewriting. As we need to cope with colloquial or even ungrammatical language as inputs, we do not rely on dependency parsing but merely use part-of-speech tags for rewriting rules. Primarily, rules remove the interrogative words and re-order subject and verb (al phrase) to form an assertive sentence. However, additional rules are needed to cast the sentence into a naturally phrased statement that Open IE can deal with. Most notably, auxiliary verbs like “*do*” need to be removed, and prepositions need to be put in their proper places, as they may appear at different positions in interrogative vs. assertive sentences. Table 2 shows some example transformations, highlighting the modified parts.

After transforming questions into statements, we employ the Stanford OpenIE tool [14] and OpenIE5.0 ([25], [26], [20], [6]) to extract triples from assertive sentences. When several triples with the same S and O are extracted from the same sentence, we retain only the one with the longest P phrase.

| Question | Statement |
|---|---|
| (1) why <i>is</i> voltmeter not connected in series | voltmeter <i>is</i> not connected in series |
| (2) why <i>are</i> chimpanzees endangered | chimpanzees <i>are</i> endangered |
| (3) why <i>do</i> men have nipples | men have nipples |
| (4) why <i>are</i> elephant seals mammals | elephant seals <i>are</i> mammals |
| (5) why <i>is</i> becoming a nurse in france hard | becoming a nurse in france <i>is</i> hard |

Table 2: Examples of questions and statements.

The resulting extractions are still noisy, but this is taken care of by the subsequent stages of corroboration, ranking and grouping, to construct a high-quality CSKB.

4.4 Output Normalization

The triples produced by OpenIE exhibit various idiosyncrasies. For cleaning them, we apply the following normalization steps:

- Replacement of plural subjects with singular forms.
- Replacement of verb inflections by their infinitive (e.g., *are eating* → *eat*).
- Removal of modalities (always, sometimes, occasionally, ...) in the predicate and object. These are kept as modality qualifiers.
- Removal of negation, put into a dedicated qualifier as negative evidence.
- Replacement of generic predicates like *are*, *is* with more specific ones like *hasColor*, *hasBodyPart*, depending on the object.
- Removal of adverbs and phatic expressions (e.g., “so”, “also”, “very much” etc.).

We completely remove triples containing any of the following:

- subjects outside the initial seed set,
- personal pronouns (“my”, “we”), and
- a short list of odd objects (e.g., xbox, youtube, quote) that frequently occur in search results but do not indicate commonsense properties.

The output of this phase are tuples of the form (*Subject, Predicate, Object, Modality, Negativity, Source, Score*), for instance, (*lion, hunts, zebra, often, positive, Google, 0.4*).

5 CORROBORATION

The output of the candidate gathering phase is bound to be noisy and contains many false positives. Therefore, we scrutinize the candidate triples by obtaining corroboration signals from a variety of additional sources. Quasimodo queries sources to test the occurrence and obtain the frequency of SPO triples. These statistics are fed into a logistic-regression classifier that decides on whether a triple is accepted or not.

The goal of this stage is to validate whether candidate triples are plausible, i.e., asserting them is justified based on several corroboration inputs:

Wikipedia and Simple Wikipedia. For each SPO candidate, we probe the article about S and compute the frequency of co-occurring

| Source | Fraction |
|--|----------|
| Google Auto-complete | 39.5% |
| CoreNLP | 52.3% |
| OpenIE5 | 42.3% |
| Answers.com | 10.5% |
| Reddit | 8.78% |
| Yahoo! Answers | 1.19% |
| Wikipedia | 62.6% |
| Simple Wikipedia | 53.6% |
| Flickr | 2.72% |
| OpenImage | 1.57% |
| Quora, Bing Auto-complete, Google Books, Answer snippets | <1% |

Table 3: Proportions of candidate triples by sources.

P and O within a window of n successive words (where $n = 5$ in our experiments).

Answer snippets from search engine. We generate Google queries using the S and O arguments of a triple as keywords, and analyze the top-100 answer snippets. The frequency of snippets containing all of S, P and O is viewed as an indicator of the triple’s validity. As search engines put tight constraints on the number of allowed queries per day, we can obtain this signal only for a limited subset of candidate assertions. We prioritize the candidates for which the other sources (Wikipedia etc.) yield high evidence.

Google Books. We create queries to the Google Books API by first forming disjunctions of surface forms for each of S, P and O, and then combining these into conjunctions. For instance, for the candidate triple (lion, live in, savanna), the query is “lion OR lions live OR lives in savanna OR savannas”. As we can use the API only with a limited budget of queries per day, we prioritized candidate triples with high evidence from other sources (Wikipedia etc.).

Image tags from OpenImages and Flickr. OpenImages is composed of ca. 20.000 classes used to annotate images. Human-verified tags exist for ca. 5.5 million images. Quasimodo checks for co-occurrences of S and O as tags for the same image and computes the frequency of such co-occurrences. For Flickr, we use its API to obtain clusters of co-occurring tags. Individual tags are not available through the API. We test for the joint occurrence of S and O in the same tag cluster.

Table 3 gives the fractions of candidate triples for which each of the sources contributes scoring signals.

Classifier training and application. We manually annotated a sample of 1700 candidate triples obtained in the candidate gathering phase.⁵ These are used to train a logistic regression model, which gives us a precision of 61%.

6 RANKING

We refer to the scores resulting from the corroboration stage as plausibility scores π . These plausibility scores are essentially linear combinations of frequency signals. Frequency is an important criterion for ranking CSK, yet CSK has other important dimensions.

⁵In comparison, TupleKB required crowd annotations for 70,000 triples.

In this section we propose two probabilistic interpretations of the scores π , referred to as τ (“typicality”) and σ (“saliency”). Intuitively, τ enables the ranking of triples by their informativeness for their subjects. Conversely, σ enables the ranking of triples by the informativeness of their p, o part.

To formalize this, we first define the probability of a triple spo .

$$P[s, p, o] = \frac{\pi(spo)}{\sum_{x \in KB} \pi(x)}.$$

Then, we compute τ and σ as:

- $\tau(s, p, o) = P[p, o | s] = \frac{P[s, p, o]}{P[s]}$.
- $\sigma(s, p, o) = P[s | p, o] = \frac{P[s, p, o]}{P[p, o]}$.

In each case, the marginals are $P[p, o] = \sum_{s \in subjects} P[s, p, o]$ and $P[s] = \sum_{p, o \in (predicates, objects)} P[s, p, o]$.

At the end of this stage, each triple spo is annotated with three scores: an internal plausibility score π , and two conditional probability scores τ and σ , which we subsequently use for ranking.

7 GROUPING

The corroboration stage of Quasimodo aimed to remove overly generic and overly specific assertions, but still yields diverse statements of different granularities with a fair amount of semantic redundancy. For instance, *hamsters are cute*, *hamsters are cute pets*, and *hamsters are cute pets for children* are all valid assertions, but more or less reflect the same commonsense property. Such variations occur with both O and P arguments, but less so with the subjects S as these are pre-selected seeds in the candidate gathering stage. To capture such redundancies while preserving different granularities and aspects, Quasimodo groups assertions into near-equivalence classes. At the top level, Quasimodo provides groups as entry points and then supports a meta-predicate refines for more detailed exploration and use-cases that need the full set of diversely phrased assertions.

Soft Co-Clustering. Our goal is to identify diverse formulations for both predicates P and subject-object pairs SO. The prior work on TupleKB has used ILP-based clustering to canonicalize predicates. However, this enforces hard grouping such that a phrase belongs to exactly one cluster. With our rich data, predicates such as “chase” or “attack” can refer to very different meanings, though: predators chasing and attacking their prey, or students chasing a deadline and attacking a problem. Analogously, S and O arguments also have ambiguous surface forms that would map to different word senses.

WebChild [33] has attempted to solve this issue by comprehensive word sense disambiguation (see [18] for a survey), but this is an additional complexity that eventually resulted in many errors. Therefore, we aim for the more relaxed and – in our findings – more appropriate objective of computing *soft clusters* where the same phrase can belong to different groups (to different degrees). As the interpretation of P phrases depends on the context of their S and O arguments, we cast this grouping task into a *co-clustering* problem where SO pairs and P phrases are jointly clustered.

Tri-Factorization of SO-P Matrix. Our method for soft co-clustering of SO pairs and P phrases is non-negative matrix tri-factorization; see [8] for mathematical foundations. We aim to compute clusters

for SO pairs and clusters for P phrases and align them with each other when meaningful. For example, the SO pairs *student problem* and *researcher problem* could be grouped together and coupled with a P cluster containing *attack* and a second cluster containing *solve*.

This goal alone would suggest a standard form of factorizing a matrix with SO pairs as rows and P phrases as columns. However, the number of clusters for SO pairs and for P phrases may be very different (because of different degrees of diversity in real-world commonsense), and decomposing the matrix into two low-rank factors with the same dimensionality would not capture this sufficiently well. Hence our approach is tri-factorization where the number of (soft) clusters for SO pairs and for P phrases can be different.

We denote the set of SO pairs and P phrases, as observed in the SPO triples after corroboration, as an $m \times n$ matrix $M_{m \times n}$, where element M_{ij} denotes the corroboration score of the triple with SO_i and P_j . We factorize M as follows:

$$M_{m \times n} = U_{m \times k} \times W_{k \times l} \times V_{l \times n}^T$$

where the low-rank dimensionalities k and l are hyper-parameters standing for the number of target SO clusters and target P clusters and the middle matrix W reflects the alignments between the two kinds of clusters. The optimization objective in this tri-factorization is to minimize the data loss in terms of the Frobenius norm, with non-negative U, W, V and orthonormal U, V :

$$\begin{aligned} \text{Minimize} \quad & \|M - U_{m \times k} \times W_{k \times l} \times V_{l \times n}^T\|_F \\ \text{s.t.} \quad & U^T U = I, \quad V^T V = I \\ & U, V, W \geq 0 \end{aligned} \tag{1}$$

We can interpret $U_{i\mu}$ as a probability of the membership of the i^{th} SO pair in the μ^{th} SO cluster. Similarly, $V_{j\nu}$ represents the probability of cluster membership of the j^{th} P phrase to the ν^{th} P cluster. The coupling of SO clusters to P clusters is given by the $W_{k \times l}$ matrix, where the μ^{th} SO cluster is linked to the ν^{th} P cluster if $W_{\mu\nu} > 0$.

Each SO pair and P phrase have a certain probability of belonging to an SO and P cluster, respectively. Hence, using a thresholding method, we assign SO_i to the μ^{th} cluster if $U_{i\mu} > \theta$ and P_j to the ν^{th} cluster if $V_{j\nu} > \theta$, in order to arrive at crisper clusters. In our experiments, we set the thresholds as follows: for the λ^{th} SO cluster, we set $\theta_\lambda = \delta \cdot \max_i U_{i\lambda}$, and for the λ^{th} P cluster, we set $\theta_\lambda = \delta \cdot \max_i V_{i\lambda}$. By varying the common thresholding parameter δ , we tune the cluster assignments of SO pairs and P phrases based on the empirical perplexity of the resulting clusters. This way, we found an empirically best value of $\delta = 0.1$.

The factor matrices in this decomposition should intuitively be sparse, as each SO pair would be associated with only a few P clusters, and vice versa. To reward sparsity, L_1 regularization is usually considered for enhancing the objective function. However, the L_1 norm makes the objective non-differentiable, and there is no analytic solution for the tri-factorization model. Like most machine-learning problems, we rely on stochastic gradient descent (SGD) to approximately solve the optimization in Equation 1. For this reason, we do not use L_1 regularization. Our SGD-based solver

| Domains | #SPO | k | l | ρ | # SO/SO cluster | | # P/P cluster | | # P clusters/P |
|----------------------------------|--------|--------|-------|--------|-----------------|-------|---------------|------|----------------|
| | | | | | avg. | max | avg. | max | avg. |
| Animals | 201942 | 3500 | 2000 | 0.10 | 38.46 | 383 | 2.8 | 24 | 1.5 |
| Persons | 218924 | 5000 | 2000 | 0.10 | 11.7 | 235 | 4.7 | 67 | 1.5 |
| Medicine | 91184 | 3000 | 1800 | 0.15 | 45.17 | 171 | 2.91 | 31 | 1.3 |
| Sport | 30794 | 1500 | 400 | 0.15 | 13.3 | 73 | 3.8 | 15 | 1.14 |
| macro-avg. (over all 49 domains) | | 1457.8 | 603.7 | 0.12 | 33.97 | 123.0 | 3.5 | 24.8 | 1.24 |

Table 4: Statistics for SO clusters and P clusters for vertical domains Animals and Occupations.

| P clusters | SO clusters |
|--|--|
| make noise at, be loud at, make noises at, croak in, croak at, quack at | fox-night, frog-night, rat-night, mouse-night, swan-night, goose-night, chicken-night, sheep-night, donkey-night, duck-night, crow-night |
| help in, help with, play part in | butterfly-environment, bee-ecosystem, butterfly-reproduction, butterfly-reproduction of plants, worm-ecosystem |
| misbehave in, talk in, sleep in, be bored in, act out in, be prepared for, be quiet in, skip, speak in | student-class, student-classes, student-lectures |
| diagnose, check for | doctor-leukemia, doctor-reflexes, doctor-asthma, doctor-diabetes, doctor-pain, doctor-adhd |

Table 5: Anecdotal examples of coupled SO clusters and P clusters from vertical domains Animals and Occupations.

initializes the factor matrices with a low density of non-zero values, determined by a hyper-parameter ρ for the ratio of non-zero matrix elements. The overall objective function then is the combination of data loss and sparseness:

$$\text{Maximize } \frac{\text{fraction of zero elements } (W)}{\text{data loss by Equation 1}}$$

All hyper-parameters – the factor ranks k and l and the sparseness ratio ρ – are tuned by performing a grid search.

8 EXPERIMENTAL EVALUATION

8.1 Implementation

Seeds. As seeds for subjects we use a combination of concepts from ConceptNet, combined with nouns extracted from WordNet, resulting in a total of around 120,000 subjects.

Candidate Gathering. In this phase Quasimodo collected ca. 14,000 questions from Quora (which has tight access restrictions), 600,000 questions from Yahoo! Answers, 2.5 million questions from Answers.com (via its sitemap), and 3.5 million questions from a Reddit dump (with a choice of suitable sub-reddits). From auto-completion suggestions, we obtained ca. 13 million questions from Google and 200,000 questions from Bing. After applying the rewriting of questions into statements and running Open IE, we obtained 24 million candidate triples; the subsequent normalization further reduced this pool to ca. 15 million triples.

Corroboration. The regression model assigned a mean π score of 0.61, with a standard deviation of 0.03. For high recall we do not apply a threshold in this phase, but utilize the scores for ranking in our evaluations.

Grouping. We performed this step on the top-50% triples, ordered by corroboration scores, amounting to ca. 2.1 million assertions. For efficient computation, we sliced this data into 49 *basic domains* based on the WordNet domain hierarchy [2]. To this end, we mapped the noun sense of each assertion subject to WordNet and assign all triples for the subject to the respective domain (e.g., animals,

plants, earth, etc.) The five largest domains are *earth, chemistry, animal, biology, and person*, containing on average 3.9k subjects and 198k assertions. We performed co-clustering on each of these slices, where hyper-parameters were tuned by grid search. Table 4 gives hyper parameter values and cluster-specific statistics of the co-clustering for three domains: number of assertions (#SPO); the co-clustering hyper-parameters SO clusters (k), P clusters (l) and sparseness ratio (ρ); the average number of elements per cluster for both SO and P clusters; and the average number of P-clusters per predicate. Additionally, we provide macro-averaged statistics for all 49 domains. Table 5 shows anecdotal examples of co-clusters for illustration.

Quasimodo CSKB. The resulting knowledge base contains ca. 4.6 million assertions for 87,000 subjects. A preliminary version is accessible by reviewers on Dropbox.⁶

Run-Time. One of the expensive component of Quasimodo is the probing of Google auto-completions. This was carried out within the allowed query limits over an entire week. Bing auto-completion was accessed through the Azure API. The another expensive component is co-clustering of all 49 domains, which takes total 142 hours in a Intel Xeon(R)(2 cores@3.20GHz) server (average 3.14 hours/ slice). All other components of Quasimodo run within a few hours at most.

8.2 Intrinsic Evaluation

We evaluate four aspects of Quasimodo: 1) size of the resulting CSKB, 2) quality, 3) recall, and 4) cluster coherence.

Size. We compare KBs in Table 6 left side by the number of subjects (#S), the number of predicates (#P), predicates occurring at least 10 times ($\#P \geq 10$), and the number of triples (#SPO). For Quasimodo we exclude all triples with *isa* / *type* predicate denoting subclass-of or instance-of relations, as these are well covered in traditional knowledge resources like WordNet, Wikidata and Yago. We also

⁶<https://www.dropbox.com/sh/r1os5uoo6v2xiac/AADinRfpUYSG1kQLm63pdMnOa?dl=0>

| | Full KB | | | | | animals | | occupations | |
|--------------------|---------|---------|--------------|------------|--------|---------|--------|-------------|--------|
| | #S | #P | #P \geq 10 | #SPO | #SPO/S | #S | #SPO | #S | #SPO |
| ConceptNet-full@en | 842,532 | 39 | 39 | 1,334,425 | 1.6 | 50 | 2,678 | 50 | 1,906 |
| ConceptNet-CSK@en | 41,331 | 19 | 19 | 214,606 | 5.2 | 50 | 1,841 | 50 | 1,495 |
| TupkeKB | 28,078 | 1,605 | 1,009 | 282,594 | 10.1 | 49 | 16,052 | 38 | 5,321 |
| WebChild | 55,036 | 20 | 20 | 13,323,132 | 242.1 | 50 | 27,223 | 50 | 26,257 |
| Quasimodo | 86,660 | 162,321 | 12,942 | 4,596,048 | 53.0 | 50 | 97,798 | 50 | 45,423 |

Table 6: Statistics for different KBs. Left side full KBs, right side two slices on animals and occupations.

compare, on the right side of the table, on two vertical domains: assertions for the 50 most popular animals and 50 most popular occupations, as determined by frequencies from Wiktionary. For ConceptNet, we report numbers for the full data including *isa* / *type* and *related* and other linguistic triples (e.g., on etymology) imported from DBpedia, WordNet and Wiktionary (ConceptNet-full), and for the proper CSK core where these relations are removed (ConceptNet-CSK).

Table 6 clearly conveys that Quasimodo has richer knowledge per subject than all other resources except for WebChild. The advantage over the manually created ConceptNet becomes particularly evident when looking at the two vertical domains, where ConceptNet-CSK contains less than 10% of the assertions that Quasimodo knows.

Quality. We asked MTurk crowd workers to evaluate the quality of CSK assertions along three dimensions: 1) meaningfulness, 2) typicality, 3) saliency. Meaningfulness denotes if a triple is conveys meaning at all, or is absurd; typicality denotes if most instances of the S concept have the PO property; saliency captures if humans would spontaneously associate PO with the given S as one of the most important traits of S.

For each evaluated triple, we obtained two judgments for the three aspects, each graded on a scale from 1 (lowest) to 5 (highest). A total 275 crowd workers completed the evaluation, with mean variance 0.70 on their ratings from 1 to 5 indicating good inter-annotator agreement. To ensure that crowd workers would not be distracted by KB-specific jargon, we translated predicates like *hasPrerequisite* or *hasProperty* into generic verb phrases like “requires” or “is”.

We sampled triples from the different CSKBs under two settings: In *comparative sampling*, we sampled triples for the same 100 subjects (50 popular occupations and 50 popular animals) across all KBs. For subject and each KB we considered the top-5-ranked triples as a pool, and uniformly sampled 100 assertions for which we obtain crowd judgement. For Quasimodo, as the rankings by typicality τ and by saliency σ differ, this sampling treated Quasimodo- τ and Quasimodo- σ as distinct CSKBs. This setting provides side-by-side comparison of triples for the same subjects.

In *horizontal sampling*, we sampled each KB separately; so they could differ on the evaluated subjects. We considered the top 5 triples of all subjects present in each KB as a pool, and picked samples from each KB uniformly at random. This evaluation mode gave us insights into the average quality of each KB. Note that it gives KBs that have fewer long-tail subjects an advantage, as triples for long-tail subjects usually receive lower human scores. Again, we considered Quasimodo rankings by τ and σ as distinct CSKBs.

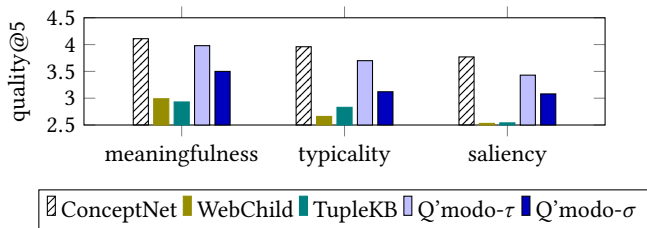


Figure 3: Quality for comparative sampling.

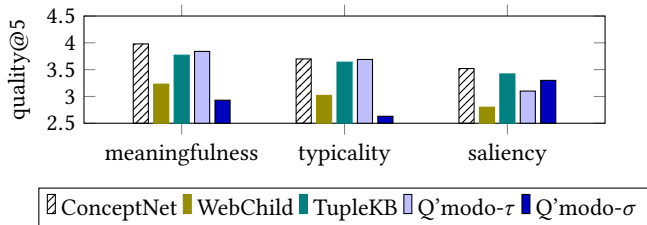


Figure 4: Quality for horizontal sampling.

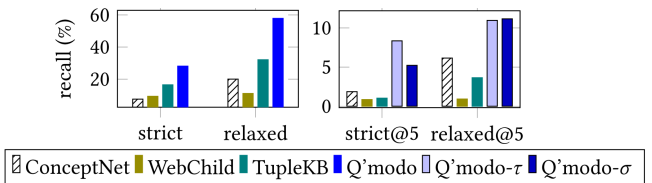


Figure 5: Recall evaluation.

The results of these evaluations are shown in Figure 3 and Figure 4. With comparative sampling, Quasimodo- τ significantly outperforms both WebChild and TupkeKB, and nearly reaches the quality of the human-generated ConceptNet. In horizontal sampling mode, Quasimodo- τ outperforms WebChild along all dimensions and outperforms TupkeKB in all dimensions but saliency. This is remarkable given that Quasimodo is 3 times bigger than ConceptNet, and is therefore penalized with horizontal sampling by its much larger number of long-tail subjects. In both evaluations, Quasimodo- τ significantly outperforms Quasimodo- σ in terms of meaningfulness and typicality. Regarding saliency the results are mixed, suggesting that further research on ranking models would be beneficial.

Recall. To compare the recall (coverage) of the different CSKBs, we asked crowd workers at MTurk to make statements about 50 occupations and 50 animals as subjects. We asked to provide short but general sentences, as spontaneous as possible so as to focus on typical and salient properties. Together with these instructions, we gave three examples for elephants (e.g., “elephants are grey”, “elephants live in Africa”) and three examples for nurses. For each

| Quasimodo | ConceptNet | WebChild | TupleKB |
|--------------------------|------------------------------------|----------------------|----------------------|
| (hasPhysicalPart, trunk) | (AtLocation, africa) | (quality, rare) | (has-part, brain) |
| (hasPhysicalPart, ear) | (HasProperty, cute) | (trait, playful) | (drink, water) |
| (live in, zoo) | (CapableOf, remember water source) | (size, large) | (prefer, vegetation) |
| (love, water) | (HasProperty, very big) | (state, numerous) | (eat, apple) |
| (be in, circus) | (CapableOf, lift logs from ground) | (quality, available) | (open, mouth) |

| Quasimodo | ConceptNet | WebChild | TupleKB |
|---------------------------|--|---------------------------|------------------------|
| (help, people) | (HasA, private life) | (emotion, euphoric) | (complete, procedure) |
| (stand long for, surgery) | (CapableOf, attempt to cure patients) | (quality, good) | (conduct, examination) |
| (learn about, medicine) | (AtLocation, golf course) | (trait, private) | (get, results) |
| (cure, people) | (CapableOf, subject patient to long waits) | (atlocation, hospital) | (has-part, adult body) |
| (can reanimate, people) | (AtLocation, examination room) | (hasproperty, aggressive) | (treat, problem) |

Table 7: Anecdotal examples (PO) for S elephant (top) and S doctor (bottom).

| KB | Elementary NDMC | Middle NDMC | CommonsenseQA | Trivia | Examveda |
|------------|--------------------|--------------------|--------------------|-------------------|-------------------|
| #Questions | 1,288 | 1,409 | 15,173 | 452 | 765 |
| Random | 27.1/27.1 | 24.6/24.6 | 33.6/33.6 | 25.9/25.9 | 25.4/25.4 |
| word2vec | 28.6/27.7 | 28.3/26.6 | 39.3/38.5 | 28.3/23.2 | 27.2/22.1 |
| Quasimodo | 38.0*/37.3* | 37.4*/35.9* | 38.4*/36.1* | 30.4/22.7* | 32.8/21.3* |
| ConceptNet | 32.5/21.4 | 25.9/14.6 | - (source) | 32.1/7.74 | 29.8/5.88 |
| TupleKB | 32.6/25 | 30.6/22.2 | 37.2/17.5 | 23.1/3.3 | 31.6/4.71 |
| WebChild | 24.8/21.1 | 28.9/24.3 | 31.1/26.3 | 27.6/10.6 | 25.5/8.24 |

Table 8: Precision and recall of answer selection in question answering. Statistically significant results (with p-value < 0.05 for paired t-test) of Quasimodo against other CSKBs are marked with an asterisk.

subject, crowd workers had 4 text fields to complete, which were pre-filled with “[subject] ...”. Each task was handed out 6 times; so in total, we obtained 2,400 simple sentences on 100 subjects.

We computed CSKB recall w.r.t. these crowd statements in two modes. In the *strict* mode, we checked for each sentence if the KB contains a triple (for the same subject) where both predicate and object are contained in the sentence and, if so, computed the word-level token overlap between PO and the sentence. In the *relaxed* setting, we checked separately if the KB contains an S-triple whose predicate appears in the sentence, and if it contains an S-triple whose object appears in the sentence. The results are shown in Figure 5. In terms of this coverage measure, Quasimodo outperforms the other CSKBs by a large margin, in both strict and relaxed modes and also when limiting ourselves to the top-5 highest-ranked triples per subject.

Cluster Coherence. We evaluate cluster coherence using an intruder task. For a random set of clusters that contain at least three P phrases, we show annotators sampled SO pairs from the cluster and samples of P phrases from the aligned cluster interspersed with an additional random intruder predicate drawn from the entire CSKB. For example, we show the SO pairs spider-web, mole-tunnel, rabbit-hole, along with the P phrases build, sing, inhabit, live in, where sing is the intruder to be found. We sampled 175 instances from two vertical slices, Animals and Persons, and used crowdsourcing (MTurk) to collect a total of 525 judgments on these 175 instances for the intruder detection task. We obtained an intruder detection accuracy of 64% for clusters in the Animals domain,

and 54% in Persons domain (compared with 25% for a random baseline). This is supporting evidence that our co-clustering method yields fairly coherent groups.

Anecdotal Examples. Table 7 provides a comparison of randomly chosen assertions for two subjects in each of the KBs: (elephant) (top) and doctor (bottom). WebChild assertions are quite vague, while TupleKB assertions are reasonable but not always salient. ConceptNet, constructed by human crowdsourcing, features high-quality assertions, but sometimes gives rather exotic properties. In contrast, the samples for Quasimodo are both typical and salient.

8.3 Extrinsic Evaluation

Answer Selection for QA. In this use case, we show that CSK helps in selecting answers for multiple-choice questions. We use five datasets: (i+ii) elementary school and middle school science questions from the AllenAI science challenge [19], (iii) common-sense questions generated from ConceptNet [31], (iv) reading comprehension questions from the TriviaQA dataset [11], and (v) exam questions from the Indian exam training platform Examveda [21].

To assess the contribution of CSKBs, for each multiple-choice answer and each word t in the answer phrase, we look for all triples of the form (t, p, o) or (s, p, t) in a KB; each time s or o appears in the respective question, we increase the score of that answer. In the end, the answer with the highest score is chosen (or the lexicographic first if two answers have equal scores). This is a basic strategy for multiple-choice QA and could be improved in many ways. However, it is sufficient to bring out the value of CSK and the differences between the CSKBs under test.

We compare four CSKBs against each other and against a word2vec baseline which computes the embeddings similarity between questions and answers. The results are shown in Table 8. Quasimodo

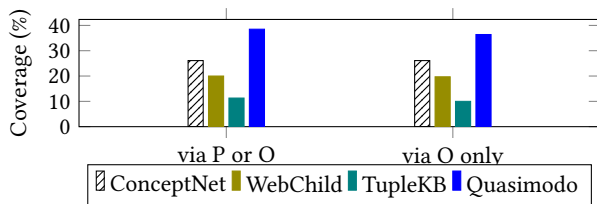


Figure 6: Coverage for word guessing game.

significantly outperforms the other CSKBs on three of the five datasets and is competitive on the other two.

Word Guessing Game. Taboo is a popular word guessing game in which a player describes a concept without using 5 taboo words, usually the strongest cues. The other player needs to guess the concept. We used a set of 578 taboo cards from the website *playtaboo.com* to evaluate the coverage of the different CSKBs.

Given a concept to be guessed, we compute the fraction of Taboo words that a KB associates with the concept, appearing in the O or P argument of the triples for the concept. This is a measure of a CSKB’s potential ability to perform in this game (i.e., not playing the game itself). The resulting coverage is shown in Table 6. Quasimodo outperforms all other KBs by this measure. TupleKB, the closest competitor on the science questions in the multiple-choice QA use case, has substantially lower coverage, indicating its limited knowledge beyond the (school-level) science domain.

9 CONCLUSION

This paper presented Quasimodo, a methodology for acquiring high-quality commonsense assertions, by harnessing non-standard input sources, like query logs and QA forums, in a novel way. As our experiments demonstrate, the Quasimodo knowledge base improves the prior state of the art, by achieving much better coverage of typical and salient commonsense properties (as determined by an MTurk study) while having similar quality in terms of precision. Extrinsic use cases further illustrate the advantages of Quasimodo. The Quasimodo data is available online⁷, and our code will be made available on Github.

REFERENCES

[1] Ricardo Baeza-Yates and Alessandro Tiberi. Extracting semantic relations from query logs. In *KDD*, 2007.

[2] Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. Revisiting the wordnet domains hierarchy: Semantics, coverage and balancing. In *COLING*, 2004.

[3] Fei Cai and Maarten De Rijke. A survey of query auto completion in information retrieval. *Foundations and Trends in Information Retrieval*, 2016.

[4] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. Context-aware query suggestion by mining click-through and session data. In *KDD*, 2008.

[5] Sreyasi Nag Chowdhury, Niket Tandon, Hakan Ferhatosmanoglu, and Gerhard Weikum. VISIR: visual and semantic image label refinement. In *WSDM*, 2018.

[6] Janara Christensen, Stephen Soderland, Oren Etzioni, et al. An analysis of open information extraction based on semantic role labeling. In *K-CAP*, 2011.

[7] Bhavana Dalvi, Niket Tandon, and Peter Clark. Domain-targeted, high precision knowledge extraction. In *TACL*, 2017.

[8] Chris H. Q. Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *KDD*, 2006.

[9] Edward A Feigenbaum. Knowledge engineering. *Annals of the New York Academy of Sciences*, 1984.

[10] Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. Canonicalizing open knowledge bases. In *CIKM*, 2014.

[11] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *ACL*, 2017.

[12] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 1995.

[13] Stanislav Malyshev, Markus Krötzsch, Larry González, Julius Gonsior, and Adrian Bielefeldt. Getting the most out of Wikidata: Semantic technology usage in Wikipedia’s knowledge graph. In *ISWC*, 2018.

[14] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL*, 2014.

[15] Mausam. Open information extraction systems and downstream applications. In *IJCAI*, 2016.

[16] John McCarthy. *Programs with common sense*. RLE and MIT computation center, 1960.

[17] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[18] Roberto Navigli. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 2009.

[19] Allen Institute of AI. AI2 science questions v2.1, 2017. <http://data.allenai.org/ai2-science-questions>.

[20] Harinder Pal et al. Demonyms and compound relational nouns in nominal open IE. In *AKBC*, 2016.

[21] Madhavi Parchure, M Sasikumar, and Ankit Garg. Veda: an online assessment and question banking system. *International Conference on Management Technology for Educational Practice*, 2009.

[22] Marius Pasca. Open-domain fine-grained class extraction from web search queries. In *EMNLP*, 2013.

[23] Marius Pasca. The role of query sessions in interpreting compound noun phrases. In *CIKM*, 2015.

[24] Marius Pasca and Benjamin Van Durme. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *ACL*, 2008.

[25] Swarnadeep Saha and Mausam. Open information extraction from conjunctive sentences. In *COLING*, 2018.

[26] Swarnadeep Saha, Harinder Pal, and Mausam. Bootstrapping for numerical open IE. In *ACL*, 2017.

[27] Carissa Schoenick, Peter Clark, Oyvind Tafjord, Peter Turney, and Oren Etzioni. Moving beyond the Turing test with the Allen AI science challenge. *Communications of the ACM*, 2017.

[28] Robyn Speer and Catherine Havasi. ConceptNet 5: A large semantic network for relational knowledge. In *Theory and Applications of Natural Language Processing*, 2012.

[29] Robyn Speer and Catherine Havasi. Representing general relational knowledge in ConceptNet 5. In *LREC*, 2012.

[30] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. SOFIE: a self-organizing framework for information extraction. In *WWW*, 2009.

[31] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *CoRR*, abs/1811.00937, 2018.

[32] Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. Reasoning about actions and state changes by injecting commonsense knowledge. In *EMNLP*, 2018.

[33] Niket Tandon, Gerard de Melo, Fabian M. Suchanek, and Gerhard Weikum. WebChild: harvesting and organizing commonsense knowledge from the web. In *WSDM*, 2014.

[34] Niket Tandon, Gerard de Melo, and Gerhard Weikum. WebChild 2.0: Fine-grained commonsense knowledge distillation. In *ACL*, 2017.

[35] Niket Tandon, Charles Hariman, Jacopo Urbani, Anna Rohrbach, Marcus Rohrbach, and Gerhard Weikum. Commonsense in parts: Mining part-whole relations from the web and image tags. In *AAAI*, 2016.

[36] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 2014.

[37] Ryen W. White, Matthew Richardson, and Wen-tau Yih. Questions vs. queries in informational search tasks. In *WWW*, 2015.

[38] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Probbase: a probabilistic taxonomy for text understanding. In *SIGMOD*, 2012.

[39] Frank F Xu, Bill Yuchen Lin, and Kenny Zhu. Automatic extraction of commonsense located near knowledge. In *ACL*, 2018.

[40] Mark Yatskar, Vicente Ordonez, and Ali Farhadi. Stating the obvious: Extracting visual common sense knowledge. In *NAACL*, 2016.

[41] Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. Augmenting end-to-end dialogue systems with commonsense knowledge. In *AAAI*, 2018.

[42] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *EMNLP*, 2018.

⁷www.dropbox.com/sh/r1os5uo06v2xiac/AAADinRfPUSg1kQLm63pdMnOa?dl=0