Enhancing Knowledge Bases with Quantity Facts

Vinh Thinh Ho
Max Planck Institute for Informatics
Saarbrücken, Germany
Bosch Center for Artificial Intelligence
Renningen, Germany
hvthinh@mpi-inf.mpg.de

Daria Stepanova
Bosch Center for Artificial Intelligence
Renningen, Germany
daria.stepanova@de.bosch.com

Dragan Milchevski
Bosch Center for Artificial Intelligence
Renningen, Germany
dragan.milchevski@de.bosch.com

Jannik Strötgen
Bosch Center for Artificial Intelligence
Renningen, Germany
jannik.stroetgen@de.bosch.com

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

ABSTRACT

Machine knowledge about the world's entities should include quantity properties, such as heights of buildings, running times of athletes, energy efficiency of car models, energy production of power plants, and more. State-of-the-art knowledge bases (KBs), such as Wikidata, cover many relevant entities but often miss the corresponding quantities. Prior work on extracting quantity facts from web contents focused on high precision for top-ranked outputs, but did not tackle the KB coverage issue. This paper presents a recall-oriented approach which aims to close this gap in knowledge-base coverage. Our method is based on iterative learning for extracting quantity facts, with two novel contributions to boost recall for KB augmentation without sacrificing the quality standards of the knowledge base. The first contribution is a query expansion technique to capture a larger pool of fact candidates. The second contribution is a novel technique for harnessing observations on value distributions for self-consistency. Experiments with extractions from more than 13 million web documents demonstrate the benefits of our method.

CCS CONCEPTS

• Information systems → Web mining; **• Computing methodologies → Information extraction**; **Knowledge representation and reasoning**.

KEYWORDS

Knowledge Bases, Information Extraction, Quantity Facts

ACM Reference Format:
Vinh Thinh Ho, Daria Stepanova, Dragan Milchevski, Jannik Strötgen, and Gerhard Weikum. 2022. Enhancing Knowledge Bases with Quantity Facts. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3485447.3511932

This work is licensed under a Creative Commons Attribution International 4.0 License.

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9096-5/22/04.
https://doi.org/10.1145/3485447.3511932

1 INTRODUCTION

Motivation and Problem. Large knowledge bases (KBs) [20, 46], such as Wikidata [43], DBpedia [24], YAGO [40] or NELL [10], contain many millions of entities and more than a billion of facts about them. The facts are typically represented in the form of subject-predicate-object (SPO) triples (along with qualifiers for higher-arity relations), and include, for example, properties of the Eiffel Tower like its location, architect, opening date, material, height, mass, and more.

The last two properties are examples of *quantity facts*: physical, technological or financial measures of interest, with proper units, associated with an entity. These are covered well for prominent entities, but KBs exhibit large gaps in quantities for less popular entities. For example, Wikidata knows more than 2 million buildings but has height entries for only less than 0.4% of these. For its 6183 sprinters, it has 100 meters race times, typically personal records, for only 38. For athletes running 400 meter hurdles, only 21 have information about their race times. Even Karsten Warholm, the Olympic champion and world record holder, falls into this huge gap (see https://www.wikidata.org/wiki/Q13900927 as of Oct 12, 2021).

The problem that we address in this paper is how to close this gap in KB coverage. To this end, we present new methods for automatically extracting quantity facts from web contents, so as to augment the KB with the missing properties.

State of the Art and Challenges. Information extraction (IE) from web contents like tables, lists and texts, has been greatly advanced over the past two decades [14, 28, 37]. However, it mostly focuses on identifying and classifying pairs of related entities for a given set of relations, typically based on patterns or distant supervision. There is not much work on extracting pairs of entities and quantities (other than for Wikipedia infoboxes and similarly semi-structured content). IE for quantity facts from text comes with the extra challenges of i) capturing the numeric value and proper unit as well as possible scaling factor from natural language, and ii) correctly connecting a quantity appearance to its entity mention in the same passage (not necessarily the same sentence).

These challenges have been studied in some prior works, most notably [27, 32, 34, 35]. However, while these methods address the first challenge, they do not solve the second one: their outputs are OpenIE-flavored with strings as arguments of SPO triples not properly mapped onto KB entities. Moreover, they do not scale to large input corpora.

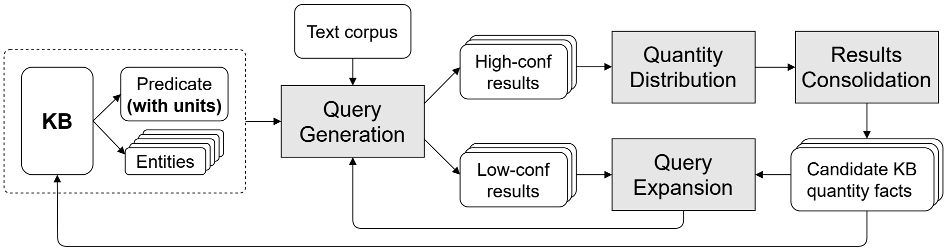

Figure 1: QL system overview.

The closest works to ours are the QEWT [35] and Qsearch [16] systems. QEWT taps into web tables as input and does not carry over to text input. Qsearch operates over large text corpora and satisfies the functional desiderata towards enhancing KB entities with quantities. However, both QEWT and Qsearch are geared for query answering rather than KB population. To this end, both optimize precision for top-ranked results of quantity-centric queries. Neither of them addresses the challenge of achieving high coverage of quantity facts for a given KB at large scale.

Approach and Contribution. To overcome the limitations of prior works, we present a novel method, called QL[1], for KB population with quantity facts. As input, our method takes a text corpus, a KB, a quantity relation to be populated (e.g., *height*) with its active domain given by a set of KB entities (e.g., *Eiffel_Tower*, *Empire_State_Building*, etc.), as well as the target units (e.g., *meter*, *feet*, *kilometer*) from the KB schema. Specifically, we use Wikidata as a KB, and produce output in the form of properly normalized quantity facts that can enhance the KB.

Our method proceeds in iterative rounds as follows. First, a set of fact candidates is computed from text by employing OpenIE [33, 34], and disambiguating entities and normalizing quantities [19, 32]. Guided by the KB schema and its entities, we first generate a *targeted query* to extract candidate facts from the text corpus, by using a combination of OpenIE for numeric expressions [33, 34] and rules [32] with entity disambiguation and quantity normalization [19]. After each round, the resulting SPO triples are split into a *high-confidence* and a *low-confidence* group, based on classifiers. The first group is considered mostly correct, while the second group mostly noisy due to many false positives. Subsequent rounds refine both sets, with the goal of expanding the *high-confidence* set.

To further increase the *precision* of the *high-confidence* group, we devise a new technique that captures the value distribution of the quantity of interest and employs a self-consistency test to demote assertions with extremely low likelihood.

To gradually improve the **recall** of the final results, we devise a new technique to expand the targeted query for gathering candidates, after each round. By analyzing contextual cues in the corpus occurrences of the current high-confidence candidates, our method learns informative expansion terms for reformulating the query. For example, given the initial query "⟨entity⟩ height" (with ⟨entity⟩ replaced by the KB entities), we may find snippets where the same relation is expressed by phrases like "stands … tall" and can then generate the query "⟨entity⟩ stands tall" for the next round.

By combining the distribution-based denoising with relation contextualization for query expansion, we successfully tackle the goal of high coverage without degenerating the KB quality standard in terms of correctness.

The key contributions of this paper are as follows:

- We present the first approach to scalably capturing quantity facts towards closing gaps in state-of-the-art knowledge bases.
- Our method includes two novel techniques for increasing recall, by automated query expansion, and ensuring high precision, by self-consistency checks against the quantity value distribution.
- Experiments with the Wikidata KB and a text corpus of more than 13 million documents demonstrate the viability of our method. For a set of 6 major predicates, we are able to extract a total of 11,687 quantity facts, with average precision of almost 80%, and above 30% of them missing in the KB. The code and data are available at https://www.mpi-inf.mpg.de/research/quantity-search/ql .

2 OVERVIEW OF THE QL SYSTEM

Given a predicate of interest and a set of KB entities, for which the respective quantities are missing in the KB, our goal is to extract these quantities from a text corpus. The QL system does this in the following steps, as depicted in Figure 1.

Step 0: Data Preparation. As pre-processing, we run Open IE [33, 34] on the corpus, recognize and disambiguate named entities (using [19] for entity linking, and [23] for coreference resolution), and detect and normalize quantities (using [32]). The output is cast into the **Qfact** representation following [16]: tuples of the form $\mathcal{F} = (e, q, X)$ where e is a KB entity, q is a quantity with a numeric value and a mapped-to-KB unit (the latter absent for mere counts), X captures context in the form of a (small) set of cue words that are informative for understanding the relation between e and q.

Example 2.1. Given the text snippet *"The Eiffel Tower is 1,063 ft high and costs about $1.5 million to construct."* with disambiguated entity: *"Eiffel Tower"* ⟶ ⟨*Eiffel_Tower*⟩ and quantities: *"1,063 ft"* ⟶ *(1063, feet)* and *"$1.5 million"* ⟶ *(1500000, $)*; Open IE generates two tuples *(The Eiffel Tower; is; 1,063 ft high)* and *(The Eiffel Tower; costs; about $1.5 million; to construct)*. Mapping them with the entities and quantities and dropping all stop words, we obtain:

- $\mathcal{F}_1$: *e* = ⟨*Eiffel_Tower*⟩; *q* = *(1063, feet)*; *X* = *"high"*
- $\mathcal{F}_2$: *e* = ⟨*Eiffel_Tower*⟩; *q* = *(1500000, $)*; *X* = *"costs construct"*

Our main contribution concerns automatically selecting, out of the set of Qfacts obtained in Step 0, those for which the context

[1] for **Q**uantity **L**eap – not to be taken too seriously

indicates the predicate of interest. To this end, we propose a query-driven technique as described next.

Step 1: Predicate-targeted Query Generation. The collected pool of candidate Qfacts are filtered and ranked by a *predicate-targeted query*, generated for the predicate at hand (e.g., *height*) leveraging the KB schema, i.e., the type signatures of predicates and the required units of quantities.

Definition 2.2 (Predicate-targeted Query). The *predicate-targeted query p* is a tuple $T(p) = (pd, pu, p\mathcal{X})$, where:

- pd is the predicate domain from the KB schema (e.g., *building*);
- pu is a set of possible units for the predicate values (e.g., *meter*, *feet*);
- $p\mathcal{X} = \{pX_0, pX_1, ...\}$ is the query context – a multiset where each pX_i is a bag of words expressing the predicate p (e.g., "*height*", "*stands tall*", etc.).

In the first iteration of QL, we construct the initial targeted query $T_0(p)$ with the fixed domain pd and units pu, taken from the KB schema, and the context comprising of only $p\mathcal{X} = \{pX_0\}$, with pX_0 being the KB label of the predicate (e.g., "height"). Subsequent iterations expand the $p\mathcal{X}$ set with further context tokens to achieve higher recall (e.g., $p\mathcal{X} = \{$"height", "stands tall", "rise", ...$\}$).

The targeted query is used to rank the candidate Qfacts in terms of their semantic relatedness. More specifically, we compute the relevance score of a Qfact $\mathcal{F} = (e, q, X)$ with respect to the query $T(p) = (pd, pu, p\mathcal{X})$ as:

$$rel(\mathcal{F}, T(p)) = \begin{cases} \max\limits_{pX_i \in p\mathcal{X}} sim(X, pX_i), & \text{if } e \in pd,\, q \in pu \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where *sim* denotes the semantic similarity between two bags of words. While various options for the choice of *sim* exist, we use the *context-embedding-distance* of [16], which is based on word embeddings. The introduced relevance score ranks all Qfacts, whose entity and quantity match with the domain and units of the target predicate, based on the semantic embedding distance between their context and the best-matched context in the query.

Based on a confidence-threshold parameter γ, we divide the ranked list of Qfacts into a *high-confidence* group H with the score $rel(\mathcal{F}, T(p)) \geq \gamma$ and a *low-confidence* group L with $rel(\mathcal{F}, T(p)) < \gamma$. For setting γ in a principled way, we employ the Deep Open Classification (DOC) method with Gaussian fitting [36], using distant supervision from a small set of ground-truth facts of the target predicate extracted from Wikidata. In practice, if no ground-truth facts are available, we could simply set the threshold γ to a high value, to ensure that the *high-confidence* group has mostly accurate results with high probability.

Step 2: Quantity Distribution. The majority of Qfacts in the *high-confidence* group are assumed to be likely correct: capturing the target predicate and having reasonable quantity values. However, a small fraction could still be spurious. To filter these out, we devise a denoising technique (Section 3), based on characterizing the value distribution of the *high-confidence* group. The idea is to spot outliers that are likely incorrect, such as buildings with height 1 meter or 5 km, which based on commonsense knowledge cannot be associated with building height. This way, we can eliminate many false positives.

Step 3: Results Consolidation. The previous step will still leave some incorrect or inaccurate Qfact candidates, due to the following: (1) for the same entity, different quantities can be stated at different precision levels (e.g., 302 m, ca. 300 m, more than 300 m); (2) different units can cause deviations after conversion (e.g., 1063 ft ⟶ 320 m); (3) false statements in the original text; (4) time-variant values or otherwise context-dependent differences in values (e.g., company revenues for a certain year or quarter, or for a certain sales region).

To resolve these kinds of noise and conflicts, we group Qfacts for the same entity-predicate pair by temporal scopes, obtained from the text passage via temporal tagging [39] or the document timestamp if available (e.g., for news articles). Within each of these groups, we select the most frequent value. The resulting Qfacts are the candidates for addition to the KB.

Step 4: Query Expansion. Since our overarching goal is to boost the recall without degrading precision, we reconsider the *low-confidence* group of Qfact candidates. This group might contain some further relevant statements, but additional sophisticated procedures are required for detecting them. To harvest positive instances from the *low-confidence* group, we propose a statistical method for query expansion, which exploits cleaned *high-confidence* facts from the previous step to automatically extend the predicate contexts $p\mathcal{X}$ with additional relevant phrases (Section 4).

Example 2.3. If (*Eiffel_Tower, height, 324m*) is added to the KB in Step 3, and *(Eiffel_Tower, 324m, "stand tall")* is a Qfact from the *low-confidence* pool, our query expansion mechanism collects the tokens *"stand tall"* as a paraphrasing of the target predicate *height*, and the initial targeted query $T_0(p)$ is expanded by setting $p\mathcal{X}$ to $p\mathcal{X} \cup \{$*"stand tall"*$\}$, which results in $T_1(p)$ with this updated context.

The above steps are repeated until a stopping criterion is met: the query cannot be expanded further, or we have reached the maximum number of iterations k (we set $k = 10$ in our implementation). We hypothesize that the introduced iterative method of breaking the quantity fact extraction task into smaller sub-problems, corresponding to spotting facts with different context phrases generated automatically would allow us to cautiously retrieve portions of likely correct facts for each computed context. Subsequently combining the results for every sub-problem should yield high overall recall.

3 DISTRIBUTION-BASED DENOISING

This section describes the denoising of the *high-confidence* group of Qfacts. To this end, all quantity values are normalized, by converting to the same standard unit (e.g., meters for height) and combining Qfacts with small differences between their normalized values (within less than 5 percent, e.g., taking the most frequent one from values like 300, 302 and 310 meters for the Eiffel Tower).

Given normalized values from the *high-confidence* group H of Qfacts, the goal is to filter out noisy values from H, based on the value distribution. The key idea is to compute the change in the distribution if a certain value is removed from H. Specifically, for each value $v \in H$, we compute two likelihood scores: (1) the *original likelihood score (o-score)* is the likelihood of v generated from the distribution constructed from the full set of values H (including v); and (2) the *consistency likelihood score (c-score)* is generated from

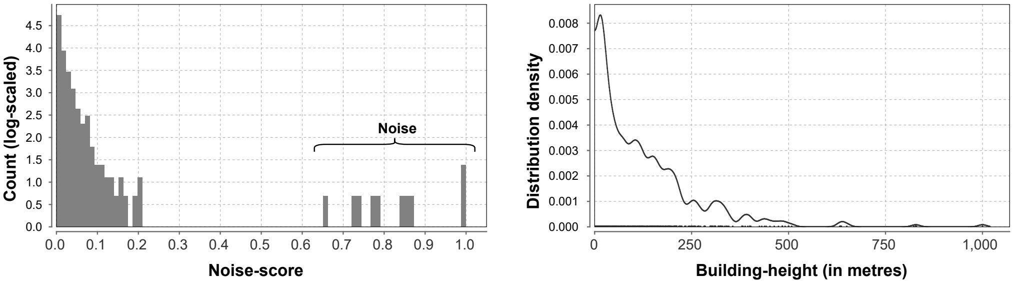

Figure 2: *(Left):* Histogram of noise-scores. *(Right):* Reconstructed distribution after denoising.

the distributions constructed from random subsets of H excluding v, computed based on a consistency learning technique. We consider the value v as noise if these two scores differ by a substantial amount:

$$\textit{noise-score}(v) = \frac{|\textit{o-score}(v) - \textit{c-score}(v)|}{\max(|\textit{o-score}(v)|, |\textit{c-score}(v)|)}$$

All Qfacts in the *high-confidence* group for which $\textit{noise-score}(v) \geq \mu$ for threshold parameter μ are filtered out.

Original Likelihood Score. For computing the original likelihood score *(o-score)*, we first construct a distribution f from H, with f being the probability density function (PDF), using Kernel Density Estimation (KDE):

$$f(v) = \frac{1}{|H|*b} \sum_{v' \in H} \Phi\left(\frac{v - v'}{b}\right)$$

where Φ is the kernel function. We use a Gaussian kernel, defined as $\Phi(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$, with bandwidth parameter b. We adopt the state-of-the-art method Improved-Sheather-Jones [4] for the automatic choice of the optimal bandwidth.

The *o-score* of value $v \in H$ is then defined as:

$$\textit{o-score}(v) = P(f \rightarrow v) = \int\limits_{x: f(x) \leq f(v)} f(x) dx \qquad (2)$$

In other words, we define the likelihood of v as the integral of f over all values whose density is not greater than $f(v)$. As the KDE could have multiple local extrema, we approximate this integral using Simpson's rule with segmentation.

Consistency Likelihood Score. For computing the *consistency likelihood score (c-score)*, we devise a technique inspired by earlier work on consistency learning [47] originally developed for image classification.

Intuitively, this is a form of self-validation, similar to the principle of cross-validation. We randomly sample a small probe set of values from H (10% of H in our implementation), and use the remaining values to construct a distribution. The constructed distribution is then used to measure the likelihood scores of the values in the probe set. This sampling and cross-validation process is repeated a large number of times. The consistency likelihood score (*c-score*) of a value v is computed as the average predicted likelihood, aggregated over all cases where v was in the probe set.

At each sampling iteration, the distribution construction and the value likelihood inference for the *c-score* are similar as for the *o-score*. The only difference is that the optimal bandwidth value b of f constructed from H when computing *o-score* is also used for constructing distributions from sample subsets of H. We hypothesize that the added noise changes only the shape of the distribution (defined by the samples), but not its smoothness (defined by b).

Denoising Output. The denoising has two results. First, we obtain the *positive* results H^+ after removing all noisy Qfacts from the high-confidence group H, which have a high $\textit{noise-score} \geq \mu$. In experiments, we set μ to 0.3. The positive results H^+ are subsequently consolidated and considered for addition to the KB. Second, we obtain a better estimation of the distribution f from H^+, which is used for query expansion, as described in the next section. For illustration, Figure 2 depicts the denoised output of the example predicate *building_height*.

4 QUERY EXPANSION

At this stage, we have cleaned the *high-confidence* group H of Qfacts and collected the positive results into H^+. While the Qfacts in the *low-confidence* group L are ranked low based on the semantic similarity function (1) from Section 2, we do not know whether they are actually wrong, thus we treat them as unknown.

In this section, we describe our approach for expanding the predicate-targeted query to achieve a better coverage of the fact extraction process. Specifically, with the current predicate-targeted query at round i: $T_i(p) = (pd, pu, pX = \{pX_0, ..., pX_i\})$, we learn a candidate context pX', which is then used to expand the query context for the next iteration.

Our query expansion technique relies on the redundancy in the data, i.e., the presence of the same entity and approximately similar quantities in both H^+ and L. To this end, we define the notion of *supported Qfacts*:

Definition 4.1 (Supported Qfact). A given Qfact $\mathcal{F} = (e, q, X)$ from the *low-confidence* group L is *supported* if in the cleaned *high-confidence* group H^+ there exists a Qfact $\mathcal{F}' = (e, q', X')$, such that $q \approx q'$ (i.e., $\mathcal{F}'$ has the same entity and approximately the same quantity as $\mathcal{F}$, after conversion to the same standard unit). The *supported set* is the set of all supported facts in L, which we denote by *supp-set*(L, H^+).

Example 4.2. Consider the *high-confidence* group H^+ with two Qfacts: H^+ = {(*Eiffel_Tower, 324 m, "height"*), (*Burj_Khalif, 2717 ft, "reached height"*)}. The following Qfacts are supported:

- $\mathcal{F}_1$ = (*Eiffel_Tower, 324 m, "stand tall"*),
- $\mathcal{F}_2$ = (*Eiffel_Tower, 1062 ft, "rise"*),

- $\mathcal{F}_3$ = (*Burj_Khalifa, 2722 ft, "originally tall"*) and
- $\mathcal{F}_4$ = (*Burj_Khalifa, 828 m, "rise height"*).

In contrast, the following facts are *not* supported:

- $\mathcal{F}_5$ = (*The_Shard, 1017 ft, "tall"*),
- $\mathcal{F}_6$ = (*Sydney_Tower, 309 m, "stand high"*) and
- $\mathcal{F}_7$ = (*Eiffel_Tower, 328 ft, "base wide"*).

The entities of $\mathcal{F}_5$ and $\mathcal{F}_6$ do not appear in H^+, while the quantity of $\mathcal{F}_7$ deviates too much. Given $L = \{\mathcal{F}_1, \ldots, \mathcal{F}_7\}$ from above, its supported set is as follows: $\textit{supp-set}(L, H^+) = \{\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4\}$.

For each candidate context pX' appearing in the *low-confidence* group L, we can compute the number of statements in L with this context that rephrase facts from the *high-confidence* group H^+. To this end, we define the notion of support as follows:

Definition 4.3 (Support). Given candidate context pX', its support is the number of Qfacts in the supported set of L whose context includes pX'.

$$\textit{supp}(pX', L, H^+) = |\{(e, q, X) \in \textit{supp-set}(L, H^+) : pX' \subseteq X\}|$$

Example 4.4. For the *high-confidence* group H^+ and the *low-confidence* group L of Example 4.2, we have: $\textit{supp}(\textit{"stand"}, L, H^+) = |\{\mathcal{F}_1\}| = 1$, $\textit{supp}(\textit{"tall"}, L, H^+) = |\{\mathcal{F}_1, \mathcal{F}_3\}| = 2$, and $\textit{supp}(\textit{"rise"}, L, H^+) = |\{\mathcal{F}_2, \mathcal{F}_4\}| = 2$. In general, the candidate context pX' is not limited to single tokens. For example, it holds that $\textit{supp}(\textit{"rise height"}, L, H^+) = |\{\mathcal{F}_4\}| = 1$.

We remove all candidate contexts with support lower than a predefined threshold. High support is important, but it is not sufficient for a candidate context to be a paraphrase or refinement of the original predicate p. Indeed, many uninformative words also have high support, for example "about", "during", "up", etc. These are words with low *inverse document frequency (idf)*, which can be filtered out by thresholding.

Support can be normalized by the highest support value among all the candidate contexts. We define the *relative support* of a candidate context as:

$$\textit{r-supp}(pX', L, H^+) = \frac{\textit{supp}(pX', L, H^+)}{\max\limits_{pX''} \textit{supp}(pX'', L, H^+)}$$

To effectively select promising candidate contexts for query expansion, we additionally take the quantities of the respective statements into account by exploiting the following proposed measures.

Definition 4.5 (Expansion Set). Given candidate context pX' and the *low-confidence* group L, the expansion set of pX' includes all Qfacts in L whose context contains pX':

$$\textit{exp-set}(pX', L) = \{(e, q, X) \in L \,|\, pX' \subseteq X\}$$

Example 4.6. Consider the *low-confidence* group L as in Example 4.2. We have: $\textit{exp-set}(\textit{"stand"}, L) = \{\mathcal{F}_1, \mathcal{F}_6\}$, $\textit{exp-set}(\textit{"tall"}, L) = \{\mathcal{F}_1, \mathcal{F}_3, \mathcal{F}_5\}$ and $\textit{exp-set}(\textit{"rise"}, L) = \{\mathcal{F}_2, \mathcal{F}_4\}$.

Intuitively, the expansion set comprises all Qfacts in the *low-confidence* group that contain pX', regardless of whether they are supported by any of the facts in the *high-confidence* group or not. These are potential statements that could be added to the *high-confidence* group if pX' is chosen to expand the query.

To measure the quality of an expansion set, we compare the quantity values of its Qfacts to the value distribution f as estimated in Section 3.

Definition 4.7 (Distribution Confidence). The distribution confidence is the average likelihood of the quantity values in the expansion set, generated by the distribution f constructed from the *high-confidence* group H^+:

$$\textit{d-conf}(pX', L, H^+) = \frac{1}{|\textit{exp-set}(pX', L)|} \sum_{(e,q,X) \in \textit{exp-set}(pX', L)} P(f \rightarrow q)$$

where $P(f \rightarrow q)$ is the integral function of Equation 2.

Intuitively, a good candidate context for paraphrasing or refining the original predicate should have an expansion set whose quantities comply with the reference distribution.

As a second signal for scoring the suitability of an expansion set, we use the original relevance scores of its Qfacts:

Definition 4.8 (Querying Confidence). The querying confidence is the average relevance score of the Qfacts in the expansion set relative to the predicate-targeted query $T_i(p)$ at the given round:

$$\textit{q-conf}(pX', L) = \frac{1}{|\textit{exp-set}(pX', L)|} \sum_{\mathcal{F} \in \textit{exp-set}(pX', L)} \textit{rel}(\mathcal{F} | T_i(p))$$

where *rel* is the function from Equation 1.

Finally, for a candidate context pX', its suitability for expanding the original query is computed using the following score:

Definition 4.9 (Candidate Expansion Score). The *expansion score* of a candidate context pX' is a (hyperparameter-)weighted sum of the *relative support*, the *querying confidence*, and the *distribution confidence*:

$$\begin{aligned} \textit{expansion-score}(pX', L, H^+) = w_1 \cdot \textit{r-supp}(pX', L, H^+) + \\ w_2 \cdot \textit{d-conf}(pX', L, H^+) + w_3 \cdot \textit{q-conf}(pX', L) \end{aligned}$$

Based on the $\textit{expansion-score}(pX', L, H^+)$ value, we rank candidate contexts pX' appearing in the *low-confidence* group L, and select the best one to expand the query for the next iteration.

5 EXPERIMENTS

We evaluated the quality of the QL output in two settings: *KB augmentation*, our major use case, and *quantity search* over web pages, as an extrinsic use case.

5.1 Experimental Setup

Input Data. We focus on the Wikidata KB, as it is the richest, among large publicly available KBs, in terms of quantity facts. We selected the following six numerical predicates, covering a spectrum of topical themes, with big gaps between potential facts and Wikidata coverage : *building-height (P2048)*, *mountain-elevation (P2044)*, *stadium-capacity (P1082)*, *river-length (P2043)*, *powerstation-capacity (P2109)*, and *earthquake-magnitude (P2528)*. We take entities from their corresponding domain types: *building*, *mountain*, *stadium*, *river*, *powerstation* and *earthquake*, respectively. These also include entities for which Wikidata has no triples on the above target predicates. We restrict ourselves to entities which are linked to a Wikipedia page, as only these are supported by the linking system

[19] that we rely on. Our goal is to acquire quantity values for these entities and predicates, regardless of whether Wikidata already has the target values or not. As input text corpus, we obtained the collection used in [16, 17] from the authors. This textual corpus comprises ca. 13 million web pages from the English Wikipedia and from news articles.

Quality Metrics. We measure the quality of the extracted quantity facts using the following three metrics: *precision*, *recall*, and *novelty*. For computing precision and recall, we use the Wikidata KB as ground-truth. Let G_p denote the set of triples stored in Wikidata for the target predicate p, and let S_p denote the set of facts extracted using our QL method. Then, precision is computed as:

$$prec(S_p) = \frac{|S_p \cap G_p| + |S_p \backslash G_p| \times prec_{sampled}(S_p \backslash G_p)}{|S_p|}$$

where $S_p \cap G_p$ and $S_p \backslash G_p$ are the sets of facts extracted by our method, that are inside and outside the ground-truth, respectively. We estimate the correctness of $S_p \backslash G_p$ by randomly sampling 30 facts and computing their precision (i.e., $prec_{sampled}$).

Recall is computed relatively to what Wikidata already contains as follows:

$$recall(S_p) = \frac{|S_p \cap G_p|}{|G_p|}$$

Obviously, this measure misses a key point that our QL system can acquire new facts from web sources that are absent in Wikidata. To evaluate this dimension, we compute the *novelty* metric as the fraction of correctly extracted facts, which are outside of the groundtruth set. This measure demonstrates the effectiveness of our approach for extending existing KBs with new knowledge:

$$novelty(S_p) = \frac{|S_p \backslash G_p| \times prec_{sampled}(S_p \backslash G_p)}{|S_p|}$$

Baselines. We compare the QL method to the following three baselines: *Qsearch* [16] and two methods based on pre-trained language models (*LMs*), RoBERTa [26] and GPT-3 [6].

• *Qsearch.*[2] Qsearch is designed to answer quantity-filter queries such as "buildings higher than 1000 ft". For each predicate, we manually create an input query for Qsearch, setting the filter condition to "> 0", to obtain the highest yield. Qsearch returns a list of confidence-ranked answers. For fair comparison, we consider only the top-N answers of Qsearch, where N is the number of facts extracted by QL.

• *LM RoBERTa.*[3] We use mask prediction for each target entity, exploiting per-predicate templates. For example, for building-height the template is: *"[ENTITY] has a height of [VALUE] [UNIT]"*, where "[ENTITY]" is the surface form of the entities and "[UNIT]" is filled from the units *pu* of the predicate-targeted query, like meter and feet for building-height. Due to the limitation of the mask prediction task, in this mode, the LM can predict only a single token, hence the need for helping by giving the unit as input. We let RoBERTa predict the value for each of the possible units, and treat the output as correct if the predicted value is correct for one of the units. In addition, we consider a *RoBERTa@5* configuration, where the output is considered correct if the correct value is among not just the top-1 but top-5 predictions.

• *GPT-3.*[4] GPT-3 has versatile interfaces including question answering. We probe GPT-3 with a short input text, beginning with 3 examples of question/answer pairs for target predicate and ending with an explicit question that GPT-3 has to answer. This is based on a hand-crafted template for each predicate. An example input is:

How tall is the Eiffel Tower? 324 meters.
How tall is Burj Khalifa? 2,717 feet.
How tall is Empire State Building? 381 m.
How tall is [ENTITY]?

where "[ENTITY]" is replaced by each of the target entities. Since GPT-3 is able to generate multi-token answers (which is in contrast to RoBERTa), we do not encode the target units in the input. The quantity extractor [32] is then used to parse the GPT-3 output to obtain the quantity result.

5.2 Results for KB Augmentation

Table 1 shows the main results: precision, recall and novelty for each predicate, as achieved by the methods under comparison. It also reports the total number of extracted quantity facts per predicate.

The total number of extracted facts by our QL method varies from hundreds to thousands, with precision reaching ca. 90 percent for the best cases. This shows the great potential for augmenting a high-quality KB with additional quantity facts. For some predicates, the precision is considerably lower, pointing out the need for further research. Nonetheless, we could choose more conservative thresholds to boost precision for a subset, at the expense of losing some recall. In any case, the QL method outperforms all baselines on precision by a huge margin (with Qsearch being second-best).

The recall, relative to Wikidata facts, is decent, but exhibits that QL could not find many of the Wikidata entities in its text corpus. This could be a limitation of what the corpus covers, and how difficult extraction from text passages is. The latter point holds particularly for Wikipedia articles, which often have complex sentences and long paragraphs with many pronouns rather than explicit entity mentions.

Finally, the novelty numbers underline the great opportunity to add new quantity facts to the KB: enriching entities with quantities that are so far absent. This potential is most pronounced for *building-height* and *earthquake-magnitude* predicates. Wikidata seems to contain many long-tail entities of these types, but misses out on the crucial quantity facts. Again, QL excels against all baselines in this regard.

To investigate whether the iterative approach of QL is productive, we also report the extraction quality measures after each round. Figure 3 plots the number of extracted facts and the precision per iteration. We do indeed see a steadily increase in the output size, thus acquiring more facts after each round. The precision lines stay fairly high throughout these iterations, showing that we barely lose precision while advancing recall and novelty. Table 2 reports the query context automatically expanded in each iteration of our method.

[2] https://qsearch.mpi-inf.mpg.de/

[3] https://huggingface.co/roberta-large

[4] https://beta.openai.com/

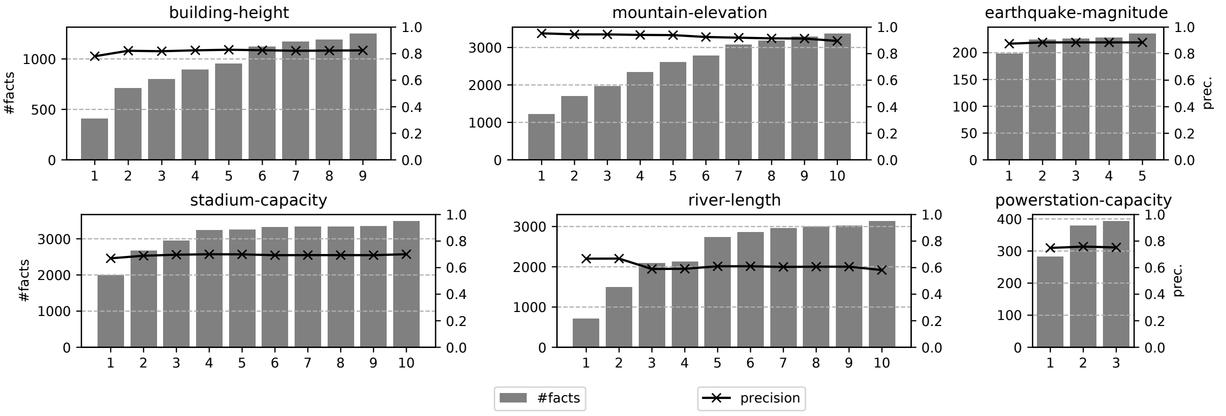

Figure 3: #Facts and precision after each round.

Table 1: #Facts, precision, recall, novelty for our proposed method QL and baselines.

Predicate	*Method*	*#Facts*	*Prec.(%)*	*Rec.(%)*	*Nov.(%)*
building-height	Qsearch	1253	56.14	9.91	48.48
	RoBERTa	1253	7.02	2.06	5.42
	RoBERTa@5	1253	20.35	5.99	15.72
	GPT-3	1253	4.56	1.34	3.52
	QL	1253	**82.46**	**24.25**	**63.70**
mountain-elevation	Qsearch	3244	57.34	20.32	10.64
	RoBERTa	3289	1.76	0.64	0.30
	RoBERTa@5	3289	8.16	2.98	1.41
	GPT-3	3289	5.99	2.19	1.03
	QL	3289	**91.36**	**33.35**	**15.78**
stadium-capacity	Qsearch	3496	31.10	19.25	16.51
	RoBERTa	3496	1.18	0.91	0.49
	RoBERTa@5	3496	6.58	5.06	2.75
	GPT-3	3496	2.99	2.30	1.25
	QL	3496	**70.10**	**53.91**	**29.26**
river-length	Qsearch	3019	14.36	5.82	5.84
	RoBERTa	3019	4.14	1.27	2.28
	RoBERTa@5	3019	19.05	5.84	10.51
	GPT-3	3019	1.11	0.34	0.61
	QL	3019	**60.71**	**18.62**	**33.48**
powerstation-capacity	Qsearch	319	53.50	7.12	27.17
	RoBERTa	394	2.75	0.51	1.23
	RoBERTa@5	394	15.14	2.80	6.76
	GPT-3	394	5.96	1.10	2.66
	QL	394	**75.23**	**13.90**	**33.60**
earthquake-magnitude	Qsearch	236	53.33	16.49	46.55
	RoBERTa	236	23.08	12.37	17.99
	RoBERTa@5	236	67.31	36.08	52.48
	GPT-3	236	38.46	20.62	29.99
	QL	236	**88.46**	**47.42**	**68.97**

Run-Time. With our implementation, the total fact extraction time for each predicate ranges from one to fifteen minutes, depending on the number of executed iterations (from three to ten), and facts extracted from them (from 236 to almost 3500, for predicates *powerstation-capacity* and *stadium-capacity*, respectively).

Table 2: Expansion of predicate-targeted queries per round.

Predicate	Expanded query context in each iteration
building-height	1) *"height"*, 2) *"tall"*, 3) *"tallest"*, 4) *"rise"*, 5) *"skyscraper"*, 6) *"high"*, 7) *"build"*, 8) *"build tallest"*, 9) *"stand"*
mountain-elevation	1) *"elevation"*, 2) *"peak"*, 3) *"highest"*, 4) *"high"*, 5) *"level sea"*, 6) *"rise"*, 7) *"height"*, 8) *"summit"*, 9) *"altitude"*, 10) *"locate"*
stadium-capacity	1) *"capacity"*, 2) *"hold"*, 3) *"seat"*, 4) *"spectator"*, 5) *"people"*, 6) *"stadium"*, 7) *"seat stadium"*, 8) *"hold people"*, 9) *"capacity seat"*, 10) *"multi-purpose seat"*
river-length	1) *"length"*, 2) *"tributary"*, 3) *"flow"*, 4) *"state"*, 5) *"river"*, 6) *"stream"*, 7) *"run"*, 8) *"long tributary"*, 9) *"longest"*, 10) *"north"*
powerstation-capacity	1) *"capacity"*, 2) *"power"*, 3) *"generate"*
earthquake-magnitude	1) *"magnitude"*, 2) *"measure"*, 3) *"scale"*, 4) *"moment"*, 5) *"estimate"*

Note that our method runs offline in batch mode. For scalability, the method can be easily parallelized by data partitioning.

5.3 Extrinsic Use Case: Quantity Search

We aim to compute top-ranked answers with quantity filter conditions such as: "buildings with height above 1000 ft", or "sprinters who ran 100 meters under 9.9 seconds". To this end, we first run the QL extraction pipeline, and then evaluate the queries against its output.

Baselines. The original Qsearch system also serves as our main baseline. In addition, we compare to results obtained from top-10 result-page snippets from Google. These are evaluated manually, by considering a snippet as correct if it contains a correct entity and a quantity that satisfies the query filter. This gives Google an

Table 3: Quantity search results.

System	*Prec.@10*	*Recall@10*	*mAP@10*
Google	0.167	0.076	0.041
Qsearch	0.290	0.177	0.119
QL	0.301	0.189	0.129

advantage, as it does not have to explicitly extract the target entity and quantity.

Benchmark. We use the benchmark queries Q150 from the work [18]. This comprises 150 queries spanning the following four domains: finance, transport, sports, and technology. Each query has ground-truth answer entities, along with their quantity values, based on manually identified Wikipedia list pages.

Results. Table 3 reports the results for this experiment: *precision@10*, *recall@10* and *mean average precision (mAP)* over the top-10 ranks. The table shows that QL can indeed improve on the – already very good – results of Qsearch, on all metrics. Both QL and Qsearch outperform the search-engine baseline, which underlines the need for this research.

6 RELATED WORK

Numerical Fact Detection. Detecting numerical expressions with units in textual data has been well addressed in prior works [2, 32, 35]. This alone is insufficient, though: for quantity facts, we also need to infer to which entity the expression refers.

The NumberTron method [27] specifically tackled numerical facts for geo-political entities, using a probabilistic graphical model. However, the approach does not scale to large text corpora and achieves only moderate precision.

The authors of [35] proposed the QEWT method for answering numerical lookup queries such as *"co2 emissions of china"*, or *"net worth of zuckerberg"*. This work uses only data from web tables and does not carry over to text input. Moreover, it does not directly solve the problem of KB population as we do.

To this end, the work on Qsearch [16, 17] customized a distantly supervised LSTM network for extracting quantities, entities and their context from individual sentences. However, Qsearch is geared for capturing a small number of top-ranked facts as responses to quantity-filter queries. Beyond the top ranks, its precision degrades drastically, by design. In contrast, our method balances larger recall with high precision, for the goal of augmenting a high-quality KB.

Recent works on numerical IE include [7, 8, 11, 12, 29]. They tackle a variety of specific settings such as commonsense assertions (e.g., "lions have 4 legs"), properties of commercial products (e.g., price, shipping weight), or quantities in clinical narratives and patient records (e.g., lab values or drug dosages). All of these prior works focus on the IE task itself, without consideration of a KB.

Numerical Embeddings. In [21, 30, 38, 42, 44] word embeddings are adapted to account for numerical expressions. The work [9] uses bi-directional and DAG-structured LSTM networks to learn simple formulas from verbal descriptions of numerical claims. None of these methods address the task of extracting full-fledge quantity facts, as needed for augmenting a KB.

Language Models for Numeracy. A direction that has gained great attention is the potential role of pre-trained language models as knowledge bases [15, 25, 31]. The idea is to prompt huge LMs like BERT, GPT-3 or T5 for numerical facts via cloze questions with masked parts to be completed through LM inference [3, 25]. However, none of these methods reaches sufficiently high precision to be considered for enhancing a high-quality KB.

Knowledge Graph Completion. Graph-structured KB embeddings [45] have been considered for completing gaps in KBs. However, their precision is far from reaching the quality standards of KBs like Wikidata. Moreover, even the exceptional works that specifically tackle numerical predictions [22] only consider the knowledge base itself as input. They are not geared for extraction of quantity facts from text.

Iterative learning has been used for fact extraction from text in various works (e.g., [1, 5, 10, 13, 41]). However, none of these works is specifically designed for extracting quantity facts.

7 CONCLUSION

We have presented a method for augmenting knowledge bases with quantity facts extracted from text at large scale. The evaluation of the method's effectiveness showed that it can indeed boost the recall of previous works while retaining high precision, even after several iterative rounds and acquiring thousands of new facts.

There are several important directions for future work. First, we plan to extend our method to support also higher-arity quantity facts as well as account for complex qualifiers. Second, targeting specific domains (e.g., medical or scientific) is another important research direction. Last but not least, exploiting the developed methods for multilingual inputs, numerical question answering tasks or quantity fact checking are promising future works.

REFERENCES

[1] Eugene Agichtein and Luis Gravano. 2000. *Snowball*: extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries, June 2-7, 2000, San Antonio, TX, USA.*

[2] Omar Alonso and Thibault Sellam. 2018. Quantitative Information Extraction From Social Data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.*

[3] Taylor Berg-Kirkpatrick and Daniel Spokoyny. 2020. An Empirical Investigation of Contextualized Number Prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020.*

[4] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. 2010. Kernel density estimation via diffusion. *The Annals of Statistics* (2010).

[5] Sergey Brin. 1998. Extracting Patterns and Relations from the World Wide Web. In *The World Wide Web and Databases, International Workshop WebDB'98, Valencia, Spain, March 27-28, 1998, Selected Papers (Lecture Notes in Computer Science).*

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

[7] Tianrun A. Cai, Luwan Zhang, Nicole Yang, Kanako K. Kumamaru, Frank J. Rybicki, Tianxi Cai, and Katherine P. Liao. 2019. EXTraction of EMR numerical data: an efficient and generalizable tool to EXTEND clinical research. *BMC Medical Informatics Decis. Mak.* (2019).

[8] Yixuan Cao, Dian Chen, Zhengqi Xu, Hongwei Li, and Ping Luo. 2021. Nested relation extraction with iterative neural network. *Frontiers Comput. Sci.* (2021).

[9] Yixuan Cao, Hongwei Li, Ping Luo, and Jiaquan Yao. 2018. Towards Automatic Numerical Cross-Checking: Extracting Formulas from Text. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018.*

[10] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010.*

[11] Chung-Chi Chen, Hen-Hsen Huang, Yow-Ting Shiue, and Hsin-Hsi Chen. [n. d.]. Numeral Understanding in Financial Tweets for Fine-Grained Crowd-Based Forecasting. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018, Santiago, Chile, December 3-6, 2018.*

[12] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How Large Are Lions? Inducing Distributions over Quantitative Attributes. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers.*

[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 *Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004.*

[14] Xu Han, Tianyu Gao, Yankai Lin, Hao Peng, Yaoliang Yang, Chaojun Xiao, Zhiyuan Liu, Peng Li, Jie Zhou, and Maosong Sun. 2020. More Data, More Relations, More Context and More Openness: A Review and Outlook for Relation Extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020.*

[15] Benjamin Heinzerling and Kentaro Inui. 2021. Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021.*

[16] Vinh Thinh Ho, Yusra Ibrahim, Koninika Pal, Klaus Berberich, and Gerhard Weikum. 2019. Qsearch: Answering Quantity Queries from Text. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part I (Lecture Notes in Computer Science).*

[17] Vinh Thinh Ho, Koninika Pal, Niko Kleer, Klaus Berberich, and Gerhard Weikum. 2020. Entities with Quantities: Extraction, Search, and Ranking. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020.*

[18] Vinh Thinh Ho, Koninika Pal, Simon Razniewski, Klaus Berberich, and Gerhard Weikum. 2021. Extracting Contextualized Quantity Facts from Web Tables. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021.*

[19] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL.*

[20] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutiérrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan F. Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* (2021).

[21] Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, and Kewei Tu. 2020. Learning Numeral Embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020.*

[22] Bhushan Kotnis and Alberto García-Durán. 2019. Learning Numerical Attributes in Knowledge Bases. In *1st Conference on Automated Knowledge Base Construction, AKBC 2019, Amherst, MA, USA, May 20-22, 2019.*

[23] Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers).*

[24] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* (2015).

[25] Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020.*

[26] 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. *CoRR* (2019). arXiv:1907.11692

[27] Aman Madaan, Ashish Mittal, Mausam, Ganesh Ramakrishnan, and Sunita Sarawagi. 2016. Numerical Relation Extraction with Minimal Supervision. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*

[28] José-Lázaro Martínez-Rodríguez, Aidan Hogan, and Ivan López-Arévalo. 2020. Information extraction meets the Semantic Web: A survey. *Semantic Web* (2020).

[29] Kartik Mehta, Ioana Oprea, and Nikhil Rasiwasia. 2021. LATEX-Numeric: Language Agnostic Text Attribute Extraction for Numeric Attributes. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, NAACL-HLT 2021, Online, June 6-11, 2021.*

[30] Aakanksha Naik, Abhilasha Ravichander, Carolyn Penstein Rosé, and Eduard H. Hovy. 2019. Exploring Numeracy in Word Embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers.*

[31] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language Models as Knowledge Bases?. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019.*

[32] Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about Quantities in Natural Language. *Transactions of the Association for Computational Linguistics* (2015).

[33] Swarnadeep Saha and Mausam. 2018. Open Information Extraction from Conjunctive Sentences. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018.*

[34] Swarnadeep Saha, Harinder Pal, and Mausam. 2017. Bootstrapping for Numerical Open IE. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers.*

[35] Sunita Sarawagi and Soumen Chakrabarti. 2014. Open-domain quantity queries on web tables: annotation, response, and consensus models. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014.*

[36] Lei Shu, Hu Xu, and Bing Liu. 2017. DOC: Deep Open Classification of Text Documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017.*

[37] Alisa Smirnova and Philippe Cudré-Mauroux. 2019. Relation Extraction Using Distant Supervision: A Survey. *ACM Comput. Surv.* (2019).

[38] Georgios P. Spithourakis and Sebastian Riedel. 2018. Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers.*

[39] Jannik Strötgen and Michael Gertz. 2016. *Domain-Sensitive Temporal Tagging.* Morgan & Claypool Publishers.

[40] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007.*

[41] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. 2009. SOFIE: a self-organizing framework for information extraction. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009.*

[42] Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for Numeracy-Preserving Word Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020.*

[43] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a Free Collaborative Knowledge Base. *Commun. ACM* (2014).

[44] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019.*

[45] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* (2017).

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

[47] Jay Yagnik and Atiq Islam. 2007. Learning people annotation from the web via consistency learning. In *Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR 2007, Augsburg, Bavaria, Germany, September 24-29, 2007.*