Entities with Quantities

Gerhard Weikum
Max Planck Institute for Informatics

The Web as a Database

Unstructured content, like text in web pages, and semistructured content, like HTML tables in web pages, has much weaker search functionality compared to structured databases. For example, joins between text documents via co-occurring entity mentions or attribute values are infeasible, unless major efforts are taken to create markup for a structured view. As for web tables, filters on entity mentions allow users to look up data, but results are noisy and error-prone because of ad-hoc choices for names of entities and value encodings, with huge heterogeneity across tables and often even within a table.

In the last few years, large knowledge graphs (KG), machine learning (ML) techniques and advances in entity linking algorithms [12, 17] have enabled search engines to overcome these issues, to a large degree [13]. By detecting entity mentions in web content and normalizing them onto KG entries, it has become possible to answer entity-centric queries about people, places and products almost as precisely and concisely as a database query. The following examples work with all major search engines and return crisp entity-level answers:

Query	Results(s)
Height of the Eiffel Tower	324 meters
highest building in Paris	Eiffel Tower
CEO of Amazon	Jeff Bezos
Bezos worth	108.9 Billion USD
CEOs of IT companies	Jeff Bezos, Sundar Pichai, Ginny Rometti, Zhang Yong, . . .

Search engines leverage look-ups in back-end knowledge graphs, and run entity detection on both user inputs and page contents to provide these answers. It seems that entity-centric search on the web has become as easy and as effective as querying a structured and curated database!

The same methodologies, particularly, entity linking, are also key to joining data for the same entity across web tables and within heterogeneous data lakes [8, 19].

Quantity Queries

On the disillusioning side, there is an interesting and challenging type of queries that is underexplored and hardly supported: searching with *quantities*: quantitative measures of entities that capture financial, physical, technological or environmental properties. Examples are: a celebrity's personal wealth, a company's quarterly revenue, a car's energy consumption, a material's thermal conductivity, or the usual and maximal dosage of a medical drug. Quantities can be represented as $\langle measure, value, unit \rangle$ triples, such as $\langle height, 8848, meter \rangle$. The units can be simple, such as meters, light-years, US dollars, Euros etc., with well-defined conversion rules between different units for the same measure. But they can also be quite sophisticated such as *kWh/100km* for a car's energy consumption or *W/(mK)* for the thermal conductivity of materials, with more complex conversion rules, e.g., between kWh/100km and MPG (miles per gallon) for electric, hybrid and fuel-based cars. Conversions often require context information, such as date for currency conversions, or location for car properties

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

(incl. carbon footprint). The International System of Units (SI) is a rich reference for measures and conversions (`https://en.wikipedia.org/wiki/International_System_of_Units`).

Search engines perform well on looking up quantities for given entities, such as retrieving the height of the Eiffel Tower. In this regard, quantity properties are not different from other properties such as city or architect. The pain point, however, is *finding* all entities (of a certain type) that satisfy a *search condition for a quantity of interest*, for example, buildings taller than 500m or runners completing a marathon under 2:10h. With few exceptions where explicit lists are available, search engines fall back to returning page links only. The following examples illustrate this disappointing behavior.

Query	Results(s)
people worth 50 Billion USD	link to “List of Americans by net worth - Wikipedia”
. . . more than 50 Billion USD	links to pages such as “Meet the world’s 50 richest billionaires in 2019”
. . . between 10 and 50 Billion Euros	links to pages such as “Inequality and Wealth Distribution in Germany”

Search engines do not understand numbers and units (with a few exceptions regarding dates and money, sometimes). For example, “15 kW” and “15.000 W” are two different strings. Units like “l/100km”, “MPG”, “MPGe” and “kWh/100km” are also just strings, and the systems are ignorant about unit conversions.

These queries would be trivial to handle if all data resided in a single database with well-designed schema, standardized value encodings, and high-quality curation. However, these databases do rarely exist, or are outdated or incomplete. One would hope that this is where encyclopedic knowledge graphs kick in, such as DBpedia, Wikidata or Yago. However, quantitative properties are very sparse in these KGs, and often represented just as strings, e.g., “250 mi ± 10” for the range of a car model. Only Wikidata contains triples for the range of cars, but only for 4 models (as of Dec. 2019). As for other measures, like engine power, energy efficiency, carbon footprint etc., none of these KGs has any data. Only the Web as a whole contains the wealth of information that is needed to compute accurate and complete answers to many kinds of quantity queries.

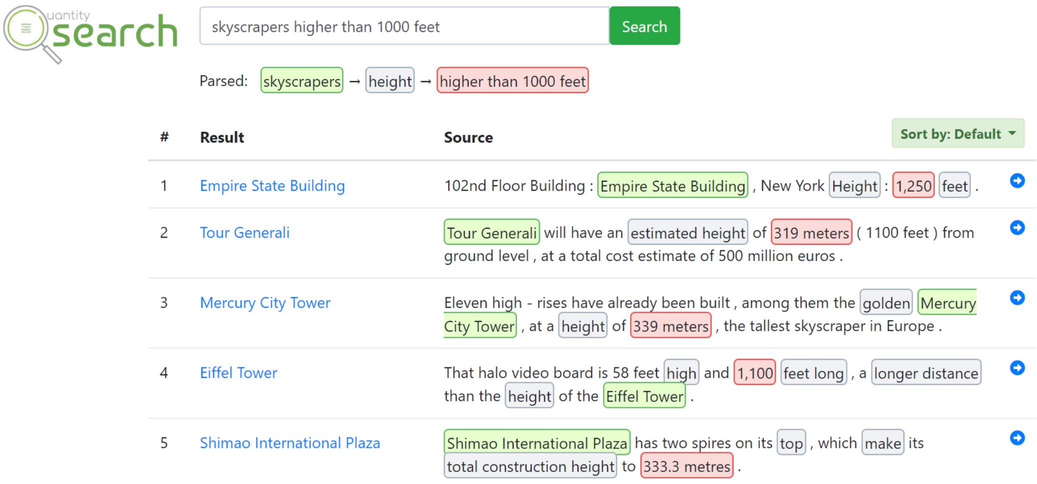
Initial Proof of Concept

Supporting quantity queries is easy over a single well-curated database. It is challenging over web page contents, web table collections or data lakes. In the latter cases, we need to overcome the obstacles of highly heterogeneous schemas, diverse and noisy value encodings, and widely varying degrees of coverage [10].

As an initial effort, we devised methods for a limited class of quantity queries over text document collections such as Wikipedia articles or news corpora. This work has led to an early prototype system, called *Qsearch* [4, 5]. The system consists of a *data preparation* stage with quantity extraction and indexing, and a *query processing* stage with matching and ranking. A Qsearch demonstrator is accessible at `https://qsearch.mpi-inf.mpg.de`. Figure 1 shows the top-ranked answers for an example query about buildings higher than 1000 ft.

Information Extraction: Qsearch uses machine learning for sequence tagging. It trains an LSTM neural network with distant supervision, and applies the learned model to tag each word in a sentence, identifying three components: i) an *entity* of interest, ii) a *quantity* that refers to this entity, and iii) *context cues* that capture what exactly the quantity denotes. For example, from the sentence “The hybrid Prius is sold in Germany for less than 30 thousand, and has a battery only range of 60 km.”, Qsearch extracts two assertions: first, related to price: i) Toyota Prius as key entity, ii) 30,000 Euros (upper bound) as quantity, iii) “sold in Germany” as cue words, and second, related to range: i) Toyota Prius as entity, ii) 60 km as quantity, iii) “battery only range” as cue words.

Query Analysis: At query time, Qsearch analyzes telegraphic queries or full questions and decomposes them into three components: *semantic target type* (e.g., buildings or hybrid cars etc.), *quantity condition* of the form $\langle comparison, value, unit\rangle$ (with comparisons like $\leq$, $\geq$, between, etc.), *context cues* that candidate results should match (e.g., “electric range in city traffic” for a query about hybrid cars).



Parsed: skyscrapers → height → higher than 1000 feet

#	Result	Source
1	Empire State Building	102nd Floor Building : Empire State Building , New York Height : 1,250 feet .
2	Tour Generali	Tour Generali will have an estimated height of 319 meters (1100 feet) from ground level , at a total cost estimate of 500 million euros .
3	Mercury City Tower	Eleven high - rises have already been built , among them the golden Mercury City Tower , at a height of 339 meters , the tallest skyscraper in Europe .
4	Eiffel Tower	That halo video board is 58 feet high and 1,100 feet long , a longer distance than the height of the Eiffel Tower .
5	Shimao International Plaza	Shimao International Plaza has two spires on its top , which make its total construction height to 333.3 metres .

Figure 1: Screenshot of Qsearch answers to query about buildings higher than 1000 ft

Matching and Ranking: Query processing aims to match all components of an assertion against the components of the query: the entity must be of the right type, the quantity condition must be satisfied, and the context cues must match as well as possible (leveraging word embeddings, e.g., to capture the relatedness of “battery only” and “electric range”). As the latter comes with uncertainty, Qsearch employs language-model-style ranking to compute the best answers.

Challenges and Opportunities

Quantity Filters: Even basic filters over quantities still pose enormous challenges. The extraction from text often faces complicated and misleading inputs, such as “The battery of the hybrid Toyoto Prius lasts well over 100,000 miles” as a spurious candidate for the electric range of this car. For more sophisticated measures such as the CO2 footprint of cars, it is crucial to consider elaborate context like the source of energy for electric cars, the driving situations (city vs. highway, summer vs. winter), and more. This will rarely be fully captured in a single sentence; so we need information extraction that combines and reconciles cues from entire paragraphs or even multiple documents. State-of-the-art work on quantity detection and extraction [1, 4, 6, 14, 15, 16] has disregarded such advanced settings so far.

Major sources for quantity information are also web tables and Open Data accessible on the Internet (e.g., `www.data.gov`, `data.gov.uk`, `data.europa.eu`, etc.). Tapping into this kind of (semi-) structured web data comes with huge challenges. Despite prior work on annotating cells in ad-hoc tables with entities and types [2, 3, 9, 18], understanding quantities and their relations to entities in this kind of online contents is way underexplored. The most notable prior endeavor is the work of Sarawagi et al. [16], which focused on a limited range of query types over web tables. Note that besides HTML tables in web pages, this direction should also consider spreadsheet data in enterprises as well as highly heterogeneous data lakes like Open Data. In addition, combining tables with cues from their surrounding text (in web pages or enterprise documents) could potentially be a powerful asset [7].

Quantity Joins: A next step would be tackling comparisons between quantities, either for the same entity or for different entities. For example, we could ask for 100m sprinters whose best time in Olympics finals is their personal record, or for such athletes whose time in the Olympics was worse than their personal best of the same year. These comparisons entail joins over the quantity values, in the second case even a non-equi join. The example may appear very special (of interest only to sports afficionados), but similarly structured queries appear in other domains as well; examples are comparing medical drugs and their usage (e.g., anti-coagulants for which the standard dosage is higher in the US than in the EU), or environmental properties of fuel-based, hybrid and electric cars in different geo-regions.

These queries are easy to express in SQL if the data resides in a single high-quality database. The challenge lies in applying them to extractions from text and tables (incl. scientific literature such as PubMed, ClinicalTrials, etc.) and to ad-hoc collections of many databases.

Quantity Aggregation: Given the inherent noise in extractions and the incompleteness of tables, it is often necessary to aggregate quantity information from multiple sources. For example, we may have to compute unions of entity sets as a basis for grouping and aggregate comparisons, or we have to combine many extractions to approximate proper values.

Such aggregations can be amazingly difficult even for seemingly simple cases. Already basic counting can be painful and challenging [11]. Consider the example of computing the total number of World Championship medals that Usain Bolt has won in his career (answer is 14). We may obtain cues from text and tables such as: he has won 100m three times, he won 11 gold medals between 2009 and 2017, he helped the Jamaican team to win the 4x100m relay race four times, 200m@2007 2nd place: Usain Bolt, 100m@2017 3rd place: Usain Bolt, etc. Can we infer the total, or at least lower and upper bounds? For prominent cases like Usain Bolt, this is not really necessary, as there are high-quality tables and lists already and we can look up the total rather than computing it. However, for less popular entities, the accessible information is often partial and spread across many sources. One difficulty is to avoid over-counting by disregarding that the 11 gold medals already include the four medals for the relay race. If we first specified a rule system, about sports medals, we could use reasoning to infer totals, but we want a solution that works out-of-the-box for all possible domains. Can we use machine learning to predict bounds for totals and other aggregates, with as little supervision as possible?

Obviously, the task gets only harder once we tackle quantities with units for realistic use cases. For example, what are the average blood lab values for diabetes patients of certain age groups in different parts of the world (as reported in clinical studies at PubMed, and other online sources)?

An Analyst's Dream

Quantity queries are often part of high-stakes information needs by advanced users, such as analysts, journalists, scientists and other knowledge workers. Ideally, an analyst would run her entire data analysis over web contents as easily as posing a keyword query or single-sentence question:

- Which runners have completed 10 marathons under 2 hours 10 minutes?
- Which is the most energy-efficient hybrid car model?
- How does the carbon footprint of Japanese cars compare to US-made cars when driven in the Bay Area?
- Which vaccinations have more than 80% coverage in the 20 population-wise largest countries?

The envisioned solution should support search engines over textual contents, web tables as well as heterogeneous data lakes. The key issues of extracting, normalizing, matching, ranking and aggregating quantities are the same regardless of whether we tap into textual contents or structured but fairly raw data.

More than 30 years ago, Bill Gates promised that "all information is at your fingertips" and Larry Page foresaw that "the ultimate search engine would understand exactly what you mean and give back exactly what you want". We have gone a long way towards these goals, but there are still many obstacles. This opinion paper is a call to overcome these issues for an interesting and valuable slice of information needs.

References

[1] Omar Alonso, Thibault Sellam: Quantitative Information Extraction From Social Data. SIGIR 2018

[2] Chandra Sekhar Bhagavatula, Thanapon Noraset, Doug Downey: TabEL: Entity Linking in Web Tables. ISWC 2015

[3] Michael J. Cafarella, Alon Y. Halevy, Hongrae Lee, Jayant Madhavan, Cong Yu, Daisy Zhe Wang, Eugene Wu: Ten Years of WebTables. PVLDB 11(12), 2018

[4] Vinh Thinh Ho, Yusra Ibrahim, Koninika Pal, Klaus Berberich, Gerhard Weikum: Qsearch: Answering Quantity Queries from Text. ISWC 2019

[5] Vinh Thinh Ho, Koninika Pal, Niko Kleer, Klaus Berberich, Gerhard Weikum: Entities with Quantities: Extraction, Search, and Ranking. Demo Paper, WSDM 2020

[6] Yusra Ibrahim, Mirek Riedewald, Gerhard Weikum: Making Sense of Entities and Quantities in Web Tables. CIKM 2016

[7] Yusra Ibrahim, Mirek Riedewald, Gerhard Weikum, Demetrios Zeinalipour-Yazti: Bridging Quantities in Tables and Text. ICDE 2019

[8] Oliver Lehmberg, Christian Bizer: Stitching Web Tables for Improving Matching Quality. PVLDB 10(11), 2017

[9] Girija Limaye, Sunita Sarawagi, Soumen Chakrabarti: Annotating and Searching Web Tables Using Entities, Types and Relationships. PVLDB 3(1), 2010

[10] Renee J. Miller, Fatemeh Nargesian, Erkang Zhu, Christina Christodoulakis, Ken Q. Pu, Periklis Andritsos: Making Open Data Transparent: Data Discovery on Open Data. IEEE Data Eng. Bull. 41(2), 2018

[11] Paramita Mirza, Simon Razniewski, Fariz Darari, Gerhard Weikum: Enriching Knowledge Bases with Counting Quantifiers. ISWC 2018

[12] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, Vijay Raghavendra: Deep Learning for Entity Matching: A Design Space Exploration. SIGMOD Conference 2018

[13] Natalya Fridman Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, Jamie Taylor: Industry-scale knowledge graphs: lessons and challenges. Commun. ACM 62(8), 2019

[14] Subhro Roy, Tim Vieira, Dan Roth: Reasoning about Quantities in Natural Language. TACL 3, 2015

[15] Swarnadeep Saha, Harinder Pal, Mausam: Bootstrapping for Numerical Open IE. ACL 2017

[16] Sunita Sarawagi, Soumen Chakrabarti: Open-domain quantity queries on web tables: annotation, response, and consensus models. KDD 2014

[17] Wei Shen, Jianyong Wang, Jiawei Han: Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. IEEE Trans. Knowl. Data Eng. 27(2), 2015

[18] Petros Venetis, Alon Y. Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, Chung Wu: Recovering Semantics of Tables on the Web. PVLDB 4(9), 2011

[19] Erkang Zhu, Dong Deng, Fatemeh Nargesian, Renée J. Miller: JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. SIGMOD 2019