Knowledge Graphs 2021: A Data Odyssey

Gerhard Weikum
Max Planck Institute for Informatics
Saarland Informatics Campus E1.4
D-66123 Saarbruecken, Germany
weikum@mpi-inf.mpg.de

ABSTRACT

Providing machines with comprehensive knowledge of the world's entities and their relationships has been a long-standing vision and challenge for AI. Over the last 15 years, huge knowledge bases, also known as knowledge graphs, have been automatically constructed from web data, and have become a key asset for search engines and other use cases. Machine knowledge can be harnessed to semantically interpret texts in news, social media and web tables, contributing to question answering, natural language processing and data analytics. This position paper reviews these advances and discusses lessons learned. It highlights the role of "DB thinking" in building and maintaining high-quality knowledge bases from web contents. Moreover, the paper identifies open challenges and new research opportunities. In particular, extracting quantitative measures of entities (e.g., height of buildings or energy efficiency of cars), from text and web tables, presents an opportunity to further enhance the scope and value of knowledge bases.

PVLDB Reference Format:
Gerhard Weikum. Knowledge Graphs 2021: A Data Odyssey. PVLDB, 14(12): 3233 - 3238, 2021.
doi:10.14778/3476311.3476393

1 INTRODUCTION

Enhancing computers with "machine knowledge" that can power intelligent applications is a long-standing goal for AI [14]. This formerly elusive vision has become practically viable, by major advances on the automatic construction of *large-scale high-quality knowledge bases (KBs)*, distilling noisy Internet content into crisp statements on entities, their attributes and relationships between them. Today, publicly available KBs, such as BabelNet (babelnet.org), DBpedia (dbpedia.org) , Wikidata (wikidata.org) or Yago (yago-knowledge.org), feature hundred millions of entities (such as people, organizations, locations and creative works like books, music etc.) and many billions of statements about them (such as who founded which company when and where, or which singer performed which song). Industrial KBs, deployed at major companies and widely referred to as *knowledge graphs (KGs)*, have an even larger scale, with one or two orders of magnitude more entities and facts [23].

A major use case where KGs have become a key asset is web search engines. When we send a query like "curie awards" to Baidu, Bing or Google, we obtain a crisp list of honors received by Marie Curie including her two Nobel prizes. The search engine taps into its background KG to automatically map the input strings "curie" and "awards" to individual entities, infer the implicitly stated relation (`won award`), and retrieves relevant facts accordingly. The returned list is strictly about Marie Curie's awards and does not erroneously conflate the result with prizes of her husband Pierre or her daughter Irene Curie (married Joliot). The KG carefully distinguishes entities, and thus enables precise and concise answering.

Further application areas of KGs include question answering, language understanding, text analytics and data cleaning – essentially, all areas where background knowledge about entities is beneficial. In data cleaning and database curation, master-data repositories of entities and their relations are a potential asset for discovering and repairing errors and for de-duplicating records. In health studies, for example, datasets can be cleaned or augmented by linking entities to a KB and leveraging the KB contents.

In addition to general-purpose encyclopedic knowledge (i.e., the "gist" of Wikipedia contents), there is also notable work on building domain-specific KGs for verticals like health and life sciences, food and nutrition, finance, consumer products, and more. Comprehensive surveys on KB construction and curation are [11, 25, 38]. In the following, Section 2 reviews history and (a subjective choice of) lessons learned, Section 3 discusses long-term challenges, and Section 4 outlines research opportunities for near-term progess.

2 LESSONS LEARNED

The first knowledge bases of notable scope and size were Cyc and WordNet, both completely hand-crafted by small teams in the 1990s. They were rich in taxonomic knowledge about types (aka classes) and general concepts, but were short of type instances, that is, individual entities. In the mid 2000s, the first generation of *automatically constructed* KBs revived the theme of machine knowledge and became a game-changer, by leveraging Wikipedia contents and applying information extraction algorithms at large scale. Notable projects were DBpedia, Freebase, KnowItAll, WikiTaxonomy and Yago. Later, further projects came along, such as DeepDive, Knowledge Vault, NELL, System T and Wikidata, and greatly advanced the methodological repertoire and the ability to construct huge KBs. These two decades of research and industrial practice on KB creation and curation provided insights on what works well and where the problems, pitfalls and risks lie. The following offers a subjective selection of lessons learned, highlighting where and why *DB thinking* is vital.

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476393

2.1 Knowledge Graphs are More Than Graphs

The term knowledge *graphs* is actually a misnomer and oversimplifies the structure and value of KBs. Graphs are binary relations, but KBs are not limited to such instances, called subject-predicate-object triples, or *SPO triples* for short. Hence, KB and not KG would be the appropriate terminology, but the term KG became widely established through press releases of big Internet stakeholders (e.g., [33]). Some research on KG embeddings even restricted itself to entity nodes and entity-to-entity relationship edges, disregarding attribute values with literals: numbers, strings, dates and textual descriptions (see survey [42] and references). KBs go beyond plain entity graphs in several important ways:

Higher-arity Relations for Context: For many facts, it is crucial to capture temporal, spatial and other context attributes, which leads to higher-arity tuples. Decomposing theses into binary-relation tuples would potentially lose information, a basic lesson in DB courses. For example, representing the two Nobel prizes of Marie Curie, one in Physics 1903 and one in Chemistry 1911, merely by SPO triples would lose the specific field-year combinations. Mature KBs overcome this issue by composite objects and qualifier predicates. These can be syntactically cast into the RDF model, but semantically this is no longer a plain graph.

Knowledge Provenance: Provenance of KB statements is another crucial case for higher-arity relations. We need to track from which sources by which methods at which time we extracted a statement. Without this information, it becomes impossible to maintain and curate the KB as its content evolves over long time horizons.

Consistency Constraints: KBs also include and leverage intensional data in the form of constraints and rules. The latter serve to derive updates by bots, such as ensuring the reciprocity of the `mother-of` and `child-of` relations. Constraints are essential for consistency checking and quality assurance: from type checking, functional and inclusion dependencies, all the way to temporal consistency and more. For example, knowing that Alan Turing is a person prevents us from erroneously placing him in the type `awards`, as people and abstractions (a supertype of awards) are disjoint. Likewise, it is impossible that he is born in both London and Princeton, detectable by a functional dependency. Consistency constraints are a key asset for knowledge acquisition from noisy sources with joint inference over multiple candidate statements (e.g., [21, 27, 35], and indispensible for KB cleaning by eliminating false positives.

All these are key points that *DB thinking* contributes to KBs.

2.2 Precision and Rigor Matter

KB construction inevitably faces a precision-recall trade-off: the more entities and facts the KB captures, the higher the likelihood that more statements will be incorrect. In prioritizing betweem *recall (KB coverage)* and *precision (KB correctness)*, we favor precision, aiming for a KB of *near-human quality*: comparable in error rates to expert curation, say 1 to 5 percent. The rationale is that the KB should provide reliable facts for all kinds of downstream use cases, and errors may get amplified through the application stack. To demonstrate this necessity, assume that a KB has 90% precision; with 1 billion statements this means 100 million errors. At this scale, fixing errors by curators or crowdsourcing is prohibitive.

A similar point can be made about the representation of entities and their types. A rigorous KB aims to *canonicalize* all entities so that they are uniquely identifiable. A canonicalized representation captures the alias names for each entity and groups statements per entity, not per name. Without this rigor, a KB could treat each of the following names as if they were different entities: “Alan Turing”, “Dr. Turing OBE”, “Alan Mathison Turing”, “Turing award”, “ACM Turing Award”, “Turing awardee” etc. This is a recipe for inconsistency and a cardinal sin from a DB perspective. The KB does not resolve this name ambiguity by itself as different entities share alias names, but the KB provides the foundation for downstream disambiguation: entity linking for mentions in texts or tables [32]. Many methods for entity linking, coreference resolution and related tasks have leveraged large KBs as a reference repository and source of distant-supervision signals.

Likewise, the type taxonomy of a KB needs to be rigorous. Once we include loose associations as class memberships or subclass-superclass subsumptions, all kinds of errors are possible. For example, including Alan Turing in the class `marathon` (as he was indeed a marathon runner), would put him also in the superclass `Greek inventions`, and placing the class `code breaker` as a subclass of `Internet crime` would lead to equally wrong inferences.

2.3 Input Data Quality is Key

It is easier to build a limited-scope *core KB* first and gradually augment and grow it, than to create a full-scale KB in a single shot. This staggered approach has the freedom of choosing which sources it taps into at different stages. Whenever possible, we would prioritize what we call *premium sources*, with the following characteristics:

- authoritative high-quality content about entities of interest,
- high coverage of many entities, and
- clean and uniform data representation.

This suggests first “picking low-hanging fruit” from structured or clean semi-structured sources (like web tables, lists etc.) and tackling text documents later and only when needed for coverage. For broad encyclopedic KBs, Wikipedia has served as such a premium source, most notably by its infoboxes and category system, but also by its clean markup, clear and largely unified headings and well-organized lists.

For vertical domains such as geography, finance or health, preexisiting databases and high-quality datasets, catalogs and rich web sites with uniform markup are the best choice. Consider the task of constructing a health KB with focus on lifestyle-induced diseases such as diabetes, hypertension or gastrointestinal disorders (possibly for the purpose of better understanding risks and complications of Covid cases). A seemingly natural source to capture entities like diseases, symptoms, risk factors, drugs and other treatments and their relationships would be PubMed articles. However, this is a formidable problem, already for recognizing and disambiguating entities; there is hardly a chance of achieving near-human-quality output. Instead, we advocate harvesting premium sources first, to create a high-quality core KB. For entities and types, the UMLS thesaurus and the MeSH taxonomy are good starting points, supplemented with biomedical databases on drugs (e.g., Drugbank), proteins (e.g., UniProt) and more. For relationships between the different entity types, e.g., which are the risk factors that trigger diabetes or aggravate it, the best choice is to tap into semi-structured contents as provided by patient-centric health portals such as mayoclinic.org or patient.info. They contain many lists with informative headings (e.g., symptoms (of diseases), side effects (for drugs) etc.), this way simplifying their extraction into relational tuples. DOM-tree labels (HTML headings, list items etc.) have been found most useful in various projects for large-scale KB construction (e.g., [4, 5]).

The take-home point is that judicious thinking about data source discovery and data quality assessment are crucial.

2.4 KB Construction is not End-to-End ML

It is tempting to think of KB construction as a single end-to-end machine learning (ML) task: collect enough training data, devise a neural network architecture (or just select Transformer), perform gradient descent to minimize a suitable loss function, and then deploy the trained model. This is wishful thinking. The task would require a huge amount of labeled training samples that cover a wide variety of cases.

KB construction is not a one-time task anyway. KBs serve as infrastructure assets, maintained over long timespans. The life-cycle involves correcting errors, adding new entities and facts, marking outdated statements (with temporal annotations), expanding the schema of attributes and relations, and further quality-assurance measures. For this bigger picture, a diverse toolbox and a substantial amount of engineering are required, with humans in the loop.

Taking the ML-only thought even further, we may not need explicit KBs at all, because we could just pull in suitable raw data into end-to-end-learning for whatever specific task arises. This is barely viable, as each task would repeatedly have to go through the demanding stages of data collection and preparation – the pain point of ML systems "in the wild". In fact, a major motivation for KBs has been to factor out these stages once and for all, so that high-quality background knowledge is already available when needed (e.g., for distant supervision or data augmentation). KBs do not become obsolete by ML; machine knowledge and machine learning complement and strengthen each other.

3 OPEN CHALLENGES

3.1 Expanding KB Coverage

Despite the impressive size of today's KBs, they have many gaps and are still far from the desired coverage of salient facts. The shortcomings have two different flavors:

- Long-tail entities or facts about them are missing, and also non-standard types are hardly covered, such as `climate activists`, `anti-war protest songs` or `aboriginal rock paintings`.
- Facts about entities are largely restricted to basic predicates regarding biography, family, awards, memberships and major works. However, many potentially interesting predicates are absent, examples being `song-is-about`, `book-features-location` or `software-deployed-at`.

The lack of capturing predicates has been referred to as a problem of "unknown unknowns" [18], as KBs and even KB architects are completely unaware of their existence and relevance. Addressing this gap requires new ways of identifying what constitutes salient knowledge about entities. For example, the following statements would be considered highly notable by most humans (and are prominently mentioned in Wikipedia articles):

- The Joan Baez song "Diamonds and Rust" is about Bob Dylan.
- A replica of the black monolith from the "2001" movie was found in a remote canyon in the Utah desert.
- Cixin Liu's book Three Body features interesting locations like Tsinghua University and Alpha Centauri.
- Frida Kahlo, the surrealistic Mexican painter, suffered her whole life from injuries in a bus accident.

Despite advances on Open IE to discover new predicates [20, 22] and on neural methods to extract relational instances [6], KB coverage is still far from where we would like it to be. Moreover, the pace at which the world evolves will widen the gap, unless we can come up with new methods for both discovering relevant data sources and reliably extracting crisp statements.

3.2 Supporting Analytic Tasks

KBs should also support knowledge workers like (data) journalists, (business and media) analysts, health experts, and more. Such advanced users go beyond finding entities or looking up their properties, and often desire to filter, compare, group, aggregate and rank entities based on *quantities*: financial, physical, technological, medical and other measures, such as annual revenue or estimated worth, distance or speed, energy consumption or CO2 footprint, blood lab values or drug dosages. Examples of quantity-centric information needs are:

- Which women have five or more marathons under 2:25:00 hours?
- How do the sales/downloads, earnings and wealth of male and female singers compare?
- How do Japanese electric cars compare to US-made models, in terms of energy efficiency, CO2 footprint and cost/km?
- Which are the top-10 countries with the highest coverage of vaccinations against virus diseases?

These kinds of analyses would be straightforward, using SQL or SPARQL queries and data-science tools, if the underlying data were stored in a single database or knowledge base. Unfortunately, this is not the case. KBs are notoriously sparse regarding quantities; for example, Wikidata contains several thousand marathon runners but knows their best times only for a few tens (not to speak of all their races). Instead of a KB, we could turn to domain-specific databases on the web, but finding the right sources in an "ocean of data" and assessing their quality, freshness and completeness is itself a big challenge.

3.3 Commonsense Knowledge

Commonsense knowledge (CSK) is the AI term for world knowledge that virtually all humans agree on. This comprises:

- Notable *properties of everyday objects*, such as: mountains are high and steep, they may be snow-covered or rocky (but they are never fast or funny).
- *Behavioral patterns* and *causality*, such as: children live with their parents, pregnancy leads to birth, and so on.

- *Human activities* and their typical settings, such as: concerts involve musicians, instruments and audience; rock concerts involve amplifiers and take place in big halls or open air (and not in bars).

CSK is difficult to acquire by machines, because of sparseness and bias in online contents, suggesting prejudiced or sensational statements such as: programmers are lonely, or programmers work 72 hours without sleep. CSK may be a crucial building block for next-generation AI, for use cases like QA and chatbots. Recent tutorials on CSK acquistion and reasoning, with ample references, are [12, 29].

A variation of CSK that matters for human-computer interaction is *socio-cultural knowledge*: behavioral patterns that do not necessarily hold universally, but are widely agreed upon within a large socio-cultural group. For example, there are preferred styles of eating meals, with different utensils (e.g., silverware vs. chopsticks) different ways of sharing, etc.

4 OPPORTUNITIES FOR DB RESEARCHERS

4.1 KB Coverage of Salient Facts

While there are good methods for handling the long tail of entities, the bottleneck is obtaining more informative facts about them. This calls for more aggressive extraction of relational tuples, in particular, for predicates outside the mainstream (e.g., `song-is-about`, `book-features-location` or `software-deployed-at`). Some of these may be extractable from lists in web pages (and their surrounding headings, captions etc.), but in general there is little hope that this mission can be achieved from semi-structured data alone. We need to turn to textual contents, while still exploiting markup if present (e.g., headings in DOM-tree paths). *Relation extraction (RE)* from text sources has been advanced over thirty years (see surveys [30, 34]), achieving good results in benchmarks such as TACRED (https://paperswithcode.com/dataset/tacred), but with precision below 80 percent and far from being robust "in the wild".

Distant Supervision for Relation Extraction: Unsurprisingly, state-of-the-art methods for RE from text (see overview by and references in [6]) are based on deep neural networks, including Transformer architectures. For training, *distant supervision* is key, to mitigate the bottleneck of fully labeled samples. This is done by leveraging existing KBs with correct SPO triples for a variety of predicates. The entity pair in a triple is matched in sentences or short passages, and this text is then considered as a positive sample. Negative samples are generated via entity pairs that are guaranteed to be counterexamples for a given predicate (incl. adversarially generated ones), and there are various ways of countering spurious matches (co-occurrence, but no relation). The methods have major limitations, calling for new departures:

- **Input Length:** Inputs are limited to short texts like single sentences or passages, due to the complexity of the neural network. This makes it impossible to extract triples when it is vital to combine cues from different parts of a long text (e.g., entire books, biographies, movie scripts).
- **Representative Samples:** The viability of distant supervision lives or dies with the amount and representativeness of samples, and also hinges on whether different predicates may be easy to confuse (e.g., song is about person vs. song is covered by person). It works well for predicates such as `spouse`, `member/employee-of` or `country-of-residence`, but struggles with less straightforward ones such as `song-is-about-person` or `movie-set-in-location`.
- **Zero-shot Extraction:** The methods cannot discover any predicates without distant-supervision samples. For example, `song-is-about-person` and `movie-set-in-location` are not included in any Wikipedia infoboxes, so it is not clear how to obtain samples. The conceivable solution is to employ some form of *transfer learning*, aiming to capture "hyper-patterns" (i.e., generic templates for patterns) that carry over from known predicates to previously unseen predicates. This has been shown to be effective for product attributes from structured websites with tables and lists [19]. For example, starting with samples of movie-book pairs for the `movie-based-on-book` relation, the method could identify a list with heading "movies based on books" and list items consisting of movie-book pairs, then learn a generic hyper-pattern to discover a list labeled "novels inspired by biographies" with book-person pairs as list items, and thus extract instances of a new relation `book-inspired-by-person`. However, this is a demanding example, and the method will not easily work robustly. Moreover, for this kind of zero-shot learning, it is widely open how to go about text sources with complex sentences rather than crisp noun phrases like list headings.

Creative Use of Language Models: With distant supervision being a bottleneck, a promising direction is to exploit *neural language models (LMs)* that have recently revolutionized the field of NLP; examples are ElMo, GPT-3 and T5, and most notably, BERT [3] and its variants (RoBERTa, ALBERT, BART, BioBERT etc.). These models, pre-computed and available at sites like https://huggingface.co/transformers/model_doc/bert.html, are Transformer networks with billions of parameters trained over huge text corpora (incl. Wikipedia articles, books, news). They are trained to minimize the loss for predicting masked-out words, subsequent sentences or similar NLP tasks. The beauty is that training data is for free without any effort to label samples: simply take a sentence or sentence pair, mask some part out and have the already known left-out part as ground truth for the loss function.

The fully trained models are then "fine-tuned" to all kinds of use cases such as question answering, sentiment classification, summarization, chatbots and more. By providing relation-specific patterns or generic hyper-patterns (i.e., templates for patterns) for cloze questions (or, equivalently, basic Who/Where/When questions), relation extraction can be cast into masked-word predictions over an LM, leveraging neural learning for machine reading comprehension [15]. For example, to infer the birthplace of Alan Turing, one would enter "X was born in [MASK]" with X being substituted by names of the respective entities, and obtain predictions for [MASK]. Potentially, the templates could be even generic, for example, "S was P in [MASK]" with S and P as placeholders for subject entities and predicates of interest.

There are other ways of incorporating BERT-like language models into RE machinery, for example, by pre-computing *BERT-based entity representations* from (carefully selected passages of) entire documents or large corpora and feeding the resulting vectors into a neural RE model. The enhanced RE model then takes as input a sentence or passage and the latent encodings of all entities that occur there, and predicts a scored list of predicate labels by putting classification layers on top [37, 39]. In fact, similar approaches have lately been leveraged for DB problems, like QA over DB tables [13, 36], QA over web tables and text [40], or entity matching for data integration [1, 16].

The bottom line is that neural language models are amazing data resources of great value and versatility. It has even been hypothesized that LMs can replace explicit KBs, by predicting components of SPO triples instead of looking them up [24]. This is quite a long shot, though, with several showstoppers coming to mind:

- **Predictions vs. Queries:** LMs always return a ranked list of predictions, with a confidence score distribution that is rarely calibrated and thus difficult to interpret. In other words, an LM will never return a definite answer. This is problematic when the number of ground-truth facts is variable and a priori unknown (e.g., founders of a company or rivers flowing through a city) or the correct answer would be empty (e.g., the cause of death for someone who is still alive).
- **Frequency Fallacies:** LM predictions are strongly affected by (direct or indirect) co-occurrence frequencies in the corpora on which the LM was trained. This induces a bias for prominent entities; predicting a rarely occurring output is a challenge. For example, when asking "Turing died in [MASK]", the LM will tend to return frequently observed answers like London (where he was born), Cambridge, England, his office, prison, Paris etc., and the correct answer Wilmslow (with a population of 30,000 people) will be way down in the ranking or completely missed.
- **Knowledge Life-cycle:** Since all knowledge is latently captured by the LM's neural network parameters, it is unclear how to maintain the knowledge: correcting errors, updating statements, adding new ones etc. Thus, the mission-critical issue of knowledge life-cycle management is disregarded.

Take-Away: What all this points to is that advancing the scope and quality of neural (or other) RE methods is largely a matter of being creative about the data sources that are leveraged at different levels: data for distant supervision, choice or discovery of best input texts to extract from, and data for contextualizing the input (like language models). This is less of an ML problem and more of a *DB-thinking* endeavor.

4.2 Entities with Quantities

Supporting analysts, journalists and other knowledge workers poses big challenges. Even if we simplify the task to merely provide building blocks towards analytic queries over KBs, there are major problems. Consider *quantity filter queries* that search for entities satisfying a condition on associated quantities (but without group-by aggregation), such as:

- women who ran a marathon under 2:25:00
- female singers worth more than 10 mio Euros,
- hybrid cars with battery range above 50 km and energy consumption below 25 kWh/100km,
- countries with CoVid vaccination rate above 50%.

Search engines handle quantity lookups for given entities fairly well (e.g., "Brigid Kosgei personal best"), but largely fail on quantity filter queries (e.g., "... under 2:25:00") due to their lack of understanding units and numeric comparisons. In addition, when matching query keywords in web-page text, it is often difficult to infer the proper entity to which a quantity mention refers. For example, the sentence "Kosgei won the race in London; compared to Emil Zapotek's time in his legendary 1952 Olympic marathon, finishing in 2:18:58 was more than four minutes faster". It is not easy for machines to understand that 2:18:58 is a marathon time and refers to Kosgei, not to Zapotek. The goal is to overcome the KB sparseness on quantities and populate KBs with more informative knowledge so as to answer the above kind of queries from a KB.

Extracting entity-quantity pairs from text has been tackled in [8], but many issues are widely open. Moreover, for quantities the more appropriate input data would probably be web tables: ad-hoc tables in HTML contents or spreadsheets or JSON files, typically small and hand-crafted without proper schema design or content curation. Information extraction from web tables has been addressed for a decade (e.g., [2, 17, 41]), but there is only limited work on the specific theme of quantities (e.g., [31] and our recent work [9, 10]). Key issues are:

- detecting and normalizing quantities that appear with varying values (e.g., estimated or stale), with different scales (e.g., with modifiers "thousand", "K" or "Mio") and units (e.g., MPG-e (miles per gallon equivalent) vs. kWh/100km);
- inferring to which entity and measure a quantity mention refers;
- contextualizing entity-quantity pairs with enough data for proper interpretation in downstream analytics – all this with very high quality and coverage.

Extracting Quantity Facts: The first issue – *quantity detection* – can be addressed by a combination of supervised learning and rules for pattern matching (e.g., [26, 28]). The second issue – *column alignment* – is surprisingly difficult when we consider complex tables. Prior works often assumed that a table has a single subject column with entities in its rows while all other columns are attribute values of these entities. However, quantities for technological or financial measures often appear in more complex tables with multiple entity columns and multiple quantity columns. This requires algorithms for inferring which quantity column refers to which entity column [9]. Nested tables with sophisticated layout further add to this problem.

Contextualizing Quantity Facts: Finally, the third issue – *contextualization of entity-quantity pairs* – is crucial for proper interpretation of statements and correct query answering. For example, for the query "hybrid cars with battery range above 50 km", the "battery" cue is decisive, as the total fuel-based range is not of interest. Likewise, interpreting financial numbers such as revenue or earnings mandates the extraction of temporal and spatial context (e.g., revenue in the last quarter of 2020 in EU countries). The right contextualization needs to consider cues from table headers and table captions, but should additionally tap into the text and structure of the page or document that contains the table. Surrounding paragraphs or headings on the DOM-tree path to the table can be informative.

Beyond Filter Queries: Quantity filter queries are just a building block and first step. Going beyond, poses further difficulties. How do we handle join queries that require stitching together multiple statements about entity-quantity statements? How does uncertainty of automatic extractions propagate to join results, and how can we ensure high confidence? For group-by aggregation, how can we ensure sufficient coverage without sacrificing precision? This is akin to approximate query processing over samples [7]). For example, to have high-confidence counts of each athlete's marathon races under 2:25:00, we need to find and extract enough instances. What is the query processing strategy over mixed-confidence statements, and how does this affect the choice of extraction strategies?

Take-Away: Capturing quantity properties of entities in sports, finance, technology and health is important for KB coverage towards analytic tasks. This is a big gap in today's major KBs, and search engines are not a good proxy. Extraction from tables, lists and possibly text sources poses difficult problems that require rethinking RE methodology, as none of the approaches in Section 4.1 seem applicable here. Quantities are not just numeric literals, but require understanding units, entities to which they refer, and contexts like spatio-temporal validity and other restrictions or refinements. While many models for neural learning come to mind for this setting, a DB-flavor key issue is the discovery and choice of the best data sources (for specific domains or even specific target entities or types), and the quality assurance towards KB population.

5 CONCLUSION

After nearly two decades on research and industrial practice with automatically constructed knowledge bases, this technology has become fairly mature. DB thinking, about data quality and consistency constraints, has played a substantial role in these advances. Nothwithstanding this success, there are new challenges and opportunities, most notably, extending KB coverage with non-standard predicates and better support for analytic tasks with quantities. While ML-fueled approaches are being pursued already, viable solutions need deeper thought on data discovery, data selection and data quality. Smart choice and creative use of data – as input sources, for training and for contextualization – remains a key issue. The data odyssey towards next-generation knowledge bases continues.

ACKNOWLEDGMENTS

Many thoughts in this paper have been shaped by intensive discussions and long-standing collaborations with Simon Razniewski and Fabian Suchanek.

REFERENCES

[1] U. Brunner, K. Stockinger: Entity Matching with Transformer Architectures - A Step Forward in Data Integration. EDBT 2020

[2] M.J. Cafarella, A.Y. Halevy, H. Lee, J. Madhavan, C. Yu, D.Z. Wang, E. Wu: Ten Years of WebTables. PVLDB 11(12), 2018

[3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL-HLT 2019

[4] X.L. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang: Knowledge Vault: a Web-scale Approach to Probabilistic Knowledge Fusion. KDD 2014

[5] P. Ernst, A. Siu, G. Weikum: KnowLife: a Versatile Approach for Constructing a Large Knowledge Graph for Biomedical Sciences. BMC Bioinformatics 16, 2015

[6] X. Han, T. Gao, Y. Lin, H. Peng, Y. Yang, C. Xiao, Z. Liu, P. Li, J. Zhou, M. Sun: More Data, More Relations, More Context and More Openness: A Review and Outlook for Relation Extraction. AACL/IJCNLP 2020

[7] J.M. Hellerstein, P.J. Haas, H.J. Wang: Online Aggregation. SIGMOD 1997

[8] V.T. Ho, Y. Ibrahim, K. Pal, K. Berberich, G. Weikum: Qsearch: Answering Quantity Queries from Text. ISWC 2019

[9] V.T. Ho, K. Pal, S. Razniewski, K. Berberich, G. Weikum: Extracting Contextualized Quantity Facts from Web Tables. WWW 2021

[10] V.T. Ho, K. Pal, G. Weikum: QuTE: Answering Quantity Queries from Web Tables. SIGMOD 2021

[11] A. Hogan et al.: Knowledge Graphs. CoRR abs/2003.02320, 2020

[12] F. Ilievski, A. Bosselut, S. Razniewski, M. Kejriwal: Commonsense Knowledge Acquisition and Representation. Tutorial with materials at https://usc-isi-i2.github.io/AAAI21Tutorial/, AAAI 2021,

[13] G. Katsogiannis-Meimarakis, G. Koutrika: A Deep Dive into Deep Learning Approaches for Text-to-SQL Systems. Tutorial video at http://www.inode-project.eu/blog/a-deep-dive-into-deep-learning-approaches-for-text-to-sql-systems/, SIGMOD 2021

[14] D. Lenat, E. Feigenbaum: On the Thresholds of Knowledge. Artificial Intelligence 47(1-3), 1991

[15] O. Levy, M. Seo, E. Choi, L. Zettlemoyer: Zero-Shot Relation Extraction via Reading Comprehension. CoNLL 2017

[16] Y. Li, J. Li, Y. Suhara, A. Doan, W.-C. Tan: Deep Entity Matching with Pre-Trained Language Models. PVLDB 14(1), 2020

[17] G. Limaye, S. Sarawagi, S. Chakrabarti: Annotating and Searching Web Tables Using Entities, Types and Relationships. PVLDB 3(1), 2010

[18] C. Lockard, P. Shiralkar, X.L. Dong, H. Hajishirzi: Multi-modal Information Extraction from Text, Semi-structured, and Tabular Data on the Web. Tutorial at WSDM 2020, ACL 2020, KDD 2020, https://sites.google.com/view/kdd-2020-multi-modal-ie

[19] C. Lockard, P. Shiralkar, X.L. Dong, H. Hajishirzi: ZeroShotCeres: Zero-Shot Relation Extraction from Semi-Structured Webpages. ACL 2020: 8105-8117

[20] Mausam: Open Information Extraction Systems and Downstream Applications. IJCAI 2016

[21] Tom M. Mitchell et al.: Never-Ending Learning. AAAI 2015

[22] C. Niklaus, M. Cetto, A. Freitas, S. Handschuh: A Survey on Open Information Extraction. COLING 2018

[23] N. Fridman Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor: Industry-scale Knowledge Graphs: Lessons and Challenges. Communications of the ACM 62(8), 2019

[24] F. Petroni, T. Rocktäschel, S. Riedel, P.S.H. Lewis, A. Bakhtin, Y. Wu, A.H. Miller: Language Models as Knowledge Bases? EMNLP/IJCNLP 2019

[25] R. Reinanda, E. Meij, M. de Rijke: Knowledge Graphs: An Information Retrieval Perspective. Foundations and Trends in Information Retrieval 14(4), 2020

[26] S. Roy, T. Vieira, D. Roth: Reasoning about Quantities in Natural Language. TACL 3, 2015

[27] C. De Sa, A. Ratner, C. Re, J. Shin, F. Wang, S. Wu, C. Zhang: Incremental Knowledge Base Construction using DeepDive. VLDB Journal 26(1), 2017

[28] S. Saha, H. Pal, Mausam: Bootstrapping for Numerical Open IE. ACL 2017

[29] M. Sap, V. Shwartz, A. Bosselut, Y. Choi, D. Roth: Commonsense Reasoning for Natural Language Processing. Tutorial with materials at https://homes.cs.washington.edu/~msap/acl2020-commonsense/, ACL 2020

[30] S. Sarawagi: Information Extraction. Foundations and Trends in Databases 1(3), 2008

[31] S. Sarawagi, S. Chakrabarti: Open-domain Quantity Queries on Web Tables: Annotation, Response, and Consensus Models. KDD 2014

[32] W. Shen, J. Wang, J. Han: Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. TKDE 27(2), 2015

[33] A. Singhal: Introducing the Knowledge Graph: Things, not Strings. Google blog post, May 16, 2012, https://blog.google/products/search/introducing-knowledge-graph-things-not/

[34] A. Smirnova, P. Cudre-Mauroux: Relation Extraction Using Distant Supervision: A Survey. ACM Computing Surveys 51(5), 2019

[35] F.M. Suchanek, M. Sozio, G. Weikum: SOFIE: a Self-organizing Framework for Information Extraction. WWW 2009

[36] J. Thorne, M. Yazdani, M. Saeidi, F. Silvestri, S. Riedel, A.Y. Halevy: From Natural Language Processing to Neural Databases. PVLDB 14(6), 2021

[37] D. Wang, W. Hu, E. Cao, W. Sun: Global-to-Local Neural Networks for Document-Level Relation Extraction. EMNLP 2020

[38] G. Weikum, L. Dong, S. Razniewski, F. Suchanek: Machine Knowledge: Creation and Curation of Comprehensive Knowledge Bases. Foundations and Trends in Databases 10(2), 2021

[39] I. Yamada, A. Asai, H. Shindo, H. Takeda, Y. Matsumoto: LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. EMNLP 2020

[40] P. Yin, G. Neubig, W.-T. Yih, S. Riedel: TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. ACL 2020

[41] S. Zhang, K. Balog: Web Table Extraction, Retrieval, and Augmentation: A Survey. ACM TIST 11(2), 2020

[42] R. Zhang, B.D. Trisedya, M. Li, Y. Jiang, J. Qi: A Comprehensive Survey on Knowledge Graph Entity Alignment via Representation Learning. CoRR abs/2103.15059, 2021