Beyond Aggregations: Understanding Count Information for Question Answering

Shrestha Ghosh

Max Planck Institute for Informatics, 66123 Saarbruecken, Germany
ghoshs@mpi-inf.mpg.de

Abstract. General-purpose knowledge bases (KBs), like DBpedia and Wikidata, contain billions of facts but, are still incomplete, use messy redundant relations and have inconsistent data. These pitfalls manifest themselves in applications like question answering (QA), where, the state-of-the-art commercial search engines answer popular user queries in information cards and carousels when the answers are available in the KBs. However, the moment a query concerns a lesser known entity, the system defaults to the classical web search and information retrieval with text highlighting. In this work we focus on the aspect of count information. Understanding count information on a semantic level is important not only for back-end data curation but also user-end applications like QA. The end goal being to move beyond aggregations into a more systematic approach to dealing with count information. Even though count information is found abundantly in text sources, extraction is non-trivial since it does not follow the general subject-predicate-object principle (*Obama has two children* → ⟨Obama, children, 2⟩ instead of ⟨Obama, has, 2 children⟩). Count information can be modelled as integers as in the previous sentence or through entity enumerations (*Obama's children are Sasha and Malia.*). We define and model count information and address its role in data curation and QA.

Keywords: Count information · Question answering · Semantic web · Knowledge base.

1 Problem Statement

Understanding data on a semantic level is essential not only for data curation, like forming KBs with class hierarchies and property constraints, but also user end applications like QA. We focus on the aspect of count information in the form of *set-valued predicates*, which conceptually model the relationship between an entity and a set of entities. Set predicates come in two variants, as i) *enumerating predicates* listing individual objects for a given subject and as ii) *counting predicates* giving the total object counts, where, both enumerating and counting variants complement each other. The major challenge lies in consolidation of count information obtained from different forms (as enumerations and counts) and facets (through multiple interpretations).

As an example, information such as *"the children of Barack Obama"* can be represented as a count 2 and as a list of entities comprising his two children. Sometimes, the count is more readily available with a few enumerations due to data privacy. For instance, for *"approximately 150,000 employees of Microsoft"*, we may have enumeration data only on the more prominent entities (CEO or others holding important positions at Microsoft) which are publicly available. Other times, when both count and enumerations are available, say in the case of *"songs by the Beatles"*, a comprehensive approach is presenting both the count ("213 songs") and enumerations.

We formulate our problem statement in the domain of QA as

"Given a natural language count query, provide a ***correct, informative*** *and* ***explainable*** *answer."*

A count query can be defined as a query regarding a set predicate - children, employees, songs - of an entity such that a natural language count query is structured as "How many .. ?" or "What is the number of .. ?". We define three requirements for the answer to a count query in the backdrop of the aforementioned query regarding songs by The Beatles.

1. Correct. Provide a reasonable estimate or the exact true value measured via precision, such that both the answers, *The Beatles have 213 songs*, or, *227 songs* are considered correct but in varying degrees.
2. Informative. Provide all or representative enumerations of the set predicate measured via precision and recall. Songs by The Beatles include 'Hey Jude' and 'Here Comes the Sun'.
3. Explainable. Provide relevant context used to derive the count and enumerations, such as a Wikipedia excerpt - "The group's main catalogue .. consists of 213 songs: 188 originals and 25 covers ..". The quality of explanation could then be judged through human annotators.

2 Importance

In QA, count queries are often processed ad-hoc by aggregating the entities in the answer in the absence of matching counting predicate. This is driven by the fact that there is only one true correct answer possible. While this assumption works perfectly well for popular entities and relations, instances with less popular entities suffer. We approach this problem by exploiting the redundancy of relevant information in the web for consolidation and present information more comprehensively through enumerations and counts.

Therefore, for a query on the *"number of songs by The Beatles"*, which has contentious answers (213, 227, 185) based on whether they were originals, covers, released and so on, only providing a list of popular songs, which is a gross underestimation of the actual number, or only the count of songs, which naturally leads to the next question regarding the constituent entities, an answer supported by enumerations and counts with explainable context is more comprehensive.

Measurement information such as *height, length, temperature*, is out of scope of count information since they do not form a relation between an entity and a set of entities. Unlike count queries regarding non-entities, such as *"number of floors in Burj Khalifa"* or *"number of votes in 2016 US Election"*, enumerating entities of set predicates support and enhance the count. Count information semantics can enhance the available knowledge by giving it more structure. For instance, "number of languages in India" is not just a statistic "121" but descriptive - "22 scheduled and 99 non-scheduled" or "21 Indo-European, 17 Dravidian, ..".

3 Related Work

This research comes close to ontology matching, count information modelling in description logics and KBs. Ontology matching focuses on alignment of entities and relations across KBs as a part of data integration [27] where the additional challenge lies in adhering to the taxonomies and ontological constraints [9,32].

Description logics models count information through qualifying number and role restrictions [15,4]. Count information in the OWL standard [21] is modelled in cardinality assertions, lower bounds and upper bounds, which comes at a complexity trade-off in both ontology ontology [14,3] as well as query language [10,25,5]. The construct of the ontological modelling languages restricts these constraints and restrictions to be relation-specific, such that limited systematic inter-relation connections exist (hierarchical or inverse relations) but enumerating and counting predicates cannot be used to reinforce each other.

Recall is an important measure for KB quality affecting applications that rely on KBs for the data [26] and can be estimated statistically, using sample overlap [20], digit distribution [33] or association rules [11]; relatively from related entities [16,33]; and from text extractions [30,29]. Our work on aligning counts with entity enumerations complements these techniques.

Count information in QA. Findings in [23] report that 5% to 10% of the questions in popular TREC QA datasets concern counts. KB QA systems like AQQU [1] and QAnswer [7] evaluated on WebQuestions [2] and LC-QuAD [35] benchmarks, perform ad-hoc special translations for typical count queries starting with *"How many ..?"*. State-of-the-art systems like Google and Bing answer count queries with both counts and enumerations, however, being limited to very few popular entities with no observable systematic behaviour. Recent advances in transformer networks [6,18], which have been applied to QA benchmark datasets including SQuAD [28] and DROP [8], predict a single answer. There is still scope for a more general setup of answer consolidation in QA over KB and texts.

Count information in IE. In the context on numerical information, conventional IE has focused on temporal [19] and measurement information [31]. Research also exists on detecting errors and outliers [36] and organizing and annotating measurement units [24,34] in the numeric information, which is present in considerable amount, in general-purpose KBs. Recent research on count IE [22,23] uses it to assess and improve KB recall [30,29]. However, such extractions are limited to manually identified count predicates.

4 Research Questions

From our guiding research question in Sec. 1, we derive the following questions.

(Q1.) How to *identify* and *consolidate* count information?
Given the structured nature of KB, it is easier to identify set predicates by analysing predicate domain, range and predicate value statistics. However, extracting count information from text is more complex with ambiguity in surface forms (integer or word representations, approximations and bounds), interpretations (number of languages in a country may be derived geographically, from official status or language families). Especially in the case of multiple interpretations, consolidation becomes challenging.

(Q2.) How to ensure an answer that is (A) *correct*, (B) *informative* and (C) *explainable*? We design our evaluation metrics to test our proposals on these three characteristics. We expand on our baselines and metrics in the Section 6.

Our aim is to solve Q1 in both KB and text setting and to subsequently draw on the strengths of each setting to finally answer a natural language count query (Q2). We begin with a preliminary investigation of identifying count information through set predicates in KBs (Q1) and aligning enumerating predicates with counting predicates for subsequent purposes of KB recall and QA (Q2). In the context of a KB which consists of subject-predicate-object (SPO) triples, a set predicate that models sets via binary membership is called an enumerating predicate. A set predicate whose values represent counts of entities modelled in the given KB is called a counting predicate.

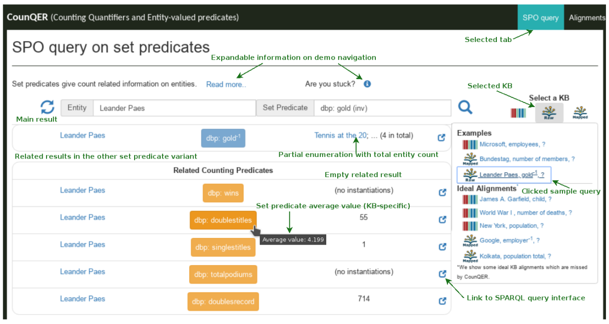
In the text setting, we gather relevant documents by offloading the information retrieval task to the available state-of-the-art and focus on dynamic extraction and consolidation of count information from the text (Q1). We exploit redundancy to predict the correct answer supported by confidence scores (Q2). Time is another critical but tangential aspect and a full-fledged problem on its own. Hence, we do not tackle time relevancy in our current work.

5 Preliminary Results

5.1 Count information in KBs

To answer Q1 in the KB setting, we first conceptualize set predicates and the enumerating and counting variants. We look into popular general-purpose KBs, namely, two versions of DBPedia (raw and mapping-based extractions), Wikidata and Freebase. We set up a methodology, named CounQER (short for "**Coun**ting **Q**uantifiers and **E**ntity-valued P**R**edicates"), to identify set predicate using supervised classification and rank set predicate alignments based on statistical and lexical measures (Q1) [12]. We provide the first step towards answering count queries through our CounQER demonstrator[1] on KB SPO triples (Q2.B,C: informative and explainable) [13] illustrated in Fig. 1 on a count query

[1] https://counqer.mpi-inf.mpg.de/spo



CounQER (Counting Quantifiers and Entity-valued predicates)

SPO query on set predicates

Set predicates give count related information on entities. Read more..

Are you stuck?

Entity: Leander Paes Set Predicate: dbp: gold (inv)

Leander Paes	dbp: gold^{-1}	Tennis at the 20; ... (4 in total)

Related Counting Predicates

Leander Paes	dbp: wins	(no instantiations)
Leander Paes	dbp: doublestitles	55
Leander Paes	dbp: singlestitles	1
Leander Paes	dbp: totalpodiums	(no instantiations)
Leander Paes	dbp: doublesrecord	714

Average value: 4.199

Examples
- Microsoft, employees, ?
- Bundestag, number of members, ?
- Leander Paes, gold^{-1}, ?

Ideal Alignments*
- James A. Garfield, child, ?
- World War I , number of deaths, ?
- New York, population, ?
- Google, employer^{-1}, ?
- Kolkata, population total, ?

*We show some ideal KB alignments which are missed by CounQER.

SPO query
Alignments
Selected tab
Expandable information on demo navigation
Selected KB
Select a KB
Raw
Mapped
Main result
Related results in the other set predicate variant
Partial enumeration with total entity count
Empty related result
Set predicate average value (KB-specific)
Clicked sample query
Link to SPARQL query interface

Fig. 1. Annotated snapshot of CounQER taken from [13].

which returns results including supporting evidence from semantically related set predicates of the other variant. We highlight the main findings below. For complete details refer to [12].

Supervised Classification. We train two logistic classifiers for identifying enumerating and counting predicates on a crowd-annotated dataset of 800 predicates chosen from the four KBs. The classifiers use i) frequency-based *textual features*, ii) predicate *type information* to encode the predicate domain and range, and iii) *KB statistics*, to capture the observed functionality through descriptive statistics and the overall datatype distribution.

Our best performing model obtained an F1 score of 0.35 for counting predicates and 0.55 for enumerating predicates. We observe that unbalanced data distribution of counting predicates, which contains only 11% positive cases is a reason why the counting classifier suffers from low precision score (0.23) despite a high recall (0.71).

Heuristic Predicate Alignment. We introduce three families of unsupervised ranking metrics for predicate pairs (e, c), where e and c belong to the predicted enumerating and counting predicate sets respectively. i) set predicate co-occurrences, *i.e.,* the number of KB subjects for which the pair (e, c) co-occur, ii) set predicate value distribution, where we compare the distribution of the number of objects e takes with the value of c for the co-occurring subjects, and iii) linguistic similarity, where we measure the relatedness of the human-readable labels of e and c. We propose averaging the best-performing heuristic from each family for final alignment scores as a trade-off between robustness and ensemble learning which requires larger evaluation data. We obtained an average of 0.75 for NDCG [17] scores at positions 1 and 3.

5.2 Count information in text

As discussed in Sec. 4 we collect the top snippets returned by search engines for a given query. The reason behind using snippets instead of the entire document is that the relevant information is typically featured in snippets. Our method is capable of supporting full text documents if required.

Extraction of relevant count values (Q1). For each query we get the top 50 search snippets. Given a text snippet we identify all noun phrases containing a count value. We use SpaCy's[2] dependency parser to identify noun phrases and NER tagger to identify counts (labelled as cardinals). We then map the cardinals to the corresponding noun phrases keeping only those i) which contain a cardinal and ii) whose head noun matches the query noun. A head noun is considered a match if the maximum path similarity, *i.e.,* the shortest path, between the synsets of the two nouns in the WordNet[3], is above a threshold.

In our present proposal we predict the final answer as the median of all the identified counts in noun phrases with matching head nouns. This method already predicts 7 exact and 5 approximate answers (48%) of the simple queries and, 1 exact and 4 approximate answers (20%) of the complex queries.

6 Evaluation

Dataset. Generating or gathering data is of utmost importance in order to conduct a proper evaluation. In our experiments on analysing count information in KBs, annotated data for training and evaluating the alignment rankings were based on human judgements collected from crowd annotations since there exist no known datasets in this regard. The natural language count queries is a diverse but small dataset of 25 simple, which have straightforward answers, and 25 complex questions, which have approximations, bounds and compositions. We expect to create a larger dataset of around 10k queries by scraping autocomplete suggestions from search engine APIs using query prompts such as *"How many* `<prefix>`*"* with varying prefixes. We would further automate the answer annotation by scraping the search engine answer box. Some manual intervention required would be for quality control - checking if queries are regarding at least one entity and a predicate. While our current focus is on queries with one count answer, we will eventually consider complex compositions.

Baselines. We set two baselines, i) *first match*, the first count, with head noun matching the query, that we encounter in the snippets and ii) *transformer prediction* which returns a median of the answers extracted from the snippets by pre-trained BERT models fine-tuned on the SQuAD2.0 dataset. The first match baseline as reflected in its performance has an unfair advantage due to internal ranking (heavily influenced by click feedback) performed by the search engine.

Metrics. We measure *correctness* (Q2.A) using the metrics - (i) *Precision*, a binary indicator which is true only when the answer matches the gold answer, (ii) *Proximity*, a ratio of the minimum of the two answers (system and gold) to the maximum, measuring the closeness of the system's answer to the gold answer, and (iii) *Confidence interval*, which gives the range of the 95% confidence interval of the system's answer. We use proximity to calculate *relaxed precision* by setting a tolerance level between $[0, 1]$. Measuring *informativeness* (Q2.B)

[2] https://spacy.io/usage/linguistic-features

[3] https://www.nltk.org/howto/wordnet.html

of enumerations is quite subjective and difficult to measure quantitatively. We have two proposals i) enumerate all entities if the count is within a threshold, say 100 else provide a list of 100 most representative enumerations and ii) get crowd-judgements on the quality of enumerations. The 95% confidence intervals provide some *explanation* (Q2.C) of the predicted answer which can be further supplemented by providing the count distribution. A qualitative measure could include highlighting parts of the snippets from which the counts are derived.

7 Discussion and Future Work

Based on preliminary results of median and first match predictions, we are aiming to develop a learning-to-rank model which takes into consideration various factors such as head-noun match, snippet rank and positive identification by transformer networks. Another path worth exploring is to fine-tune BERT with the signals mentioned above in order to predict the final answer. We aim to deal with facet consolidation in the future for dealing with popular counts and multiple interpretations. Popular count values, which when different from the true count, affects the final outcome. For example, the query *"number of books written by J.K. Rowling"*, returns multiple occurrences of the count 7, which corresponds to the popular Harry Potter series thereby overshadowing her 15 other works. Similarly, multiple surface form interpretations, such as *"number of translations of the Bible"* which can mean translations of the multiple texts (the Old/New Testament), or their types (dynamic, formal, idiomatic), leads to multiple correct answers.

Acknowledgement. I would like to thank my supervisors Simon Razniewski and Gerhard Weikum for their invaluable support and inputs.

References

1. Bast, H., Haussmann, E.: More accurate question answering on Freebase (CIKM 2015)
2. Berant, J., Chou, A., Frostig, R., Liang, P.: Semantic parsing on freebase from question-answer pairs (EMNLP 2013)
3. Calvanese, D., Eiter, T., Ortiz, M.: Regular path queries in expressive description logics with nominals (AAAI 2009)
4. Calvanese, D., Lenzerini, M., Nardi, D.: Description logics for conceptual data modeling (Logics for databases and information systems 1998)
5. Dell, H., Roth, M., Wellnitz, P.: Counting answers to existential questions (ICALP 2019)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (NAACL-HLT 2019)
7. Diefenbach, D., et al.: QAnswer: A question answering prototype bridging the gap between a considerable part of the LOD cloud and end-users (WWW 2019)
8. Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., Gardner, M.: Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs (NACL-HLT 2019)

9. Euzenat, J., Shvaiko, P.: Ontology matching (Springer 2007)
10. Fan, W., Wu, Y., Xu, J.: Adding counting quantifiers to graph patterns (SIGMOD 2016)
11. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases (WSDM 2017)
12. Ghosh, S., Razniewski, S., Weikum, G.: Uncovering hidden semantics of set information in knowledge bases. Journal of Web Semantics (2020)
13. Ghosh, S., et al.: CounQER: A system for discovering and linking count information in knowledge bases (ESWC 2020)
14. Glimm, B., Lutz, C., Horrocks, I., Sattler, U.: Conjunctive query answering for the description logic SHIQ (JAIR 2008)
15. Hollunder, B., Baader, F.: Qualifying number restrictions in concept languages. (KR 1991)
16. Hopkinson, A., Gurdasani, A., Palfrey, D., Mittal, A.: Demand-Weighted Completeness Prediction for a Knowledge Base (NAACL 2018)
17. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques (TOIS 2002)
18. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations (ICLR 2019)
19. Ling, X., Weld, D.S.: Temporal information extraction (AAAI 2010)
20. Luggen, M., et al.: Non-parametric class completeness estimators for collaborative knowledge graphs—the case of wikidata (ISWC 2019)
21. McGuinness, D.L., Van Harmelen, F., et al.: OWL web ontology language overview (W3C recommendation 2004)
22. Mirza, P., Razniewski, S., Darari, F., Weikum, G.: Cardinal virtues: Extracting relation cardinalities from text (ACL 2017)
23. Mirza, P., Razniewski, S., Darari, F., Weikum, G.: Enriching knowledge bases with counting quantifiers (ISWC 2018)
24. Neumaier, S., Umbrich, J., Parreira, J.X., Polleres, A.: Multi-level semantic labelling of numerical values (ISWC 2016)
25. Nikolaou, C., Kostylev, E.V., Konstantinidis, G., Kaminski, M., Grau, B.C., Horrocks, I.: Foundations of ontology-based data access under bag semantics (AI 2019)
26. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods (SWJ 2017)
27. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching (VLDB 2001)
28. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad (ACL 2018)
29. Razniewski, S., Jain, N., Mirza, P., Weikum, G.: Coverage of information extraction from sentences and paragraphs (EMNLP 2019)
30. Razniewski, S., Suchanek, F., Nutt, W.: But what do we actually know? (AKBC 2016)
31. Saha, S., et al.: Bootstrapping for numerical Open IE (ACL 2017)
32. Shvaiko, P., Euzenat, J.: Ontology matching: state of the art and future challenges (TKDE 2013)
33. Soulet, A., Giacometti, A., Markhoff, B., Suchanek, F.M.: Representativeness of knowledge bases with the generalized benford's law (ISWC 2018)
34. Subercaze, J.: Chaudron: extending DBpedia with measurement (ESWC 2017)
35. Trivedi, P., Maheshwari, G., Dubey, M., Lehmann, J.: LC-Quad: A corpus for complex question answering over knowledge graphs (ISWC 2017)
36. Wienand, D., Paulheim, H.: Detecting incorrect numerical data in DBpedia (ESWC 2014)