How much is a Triple?

Estimating the Cost of Knowledge Graph Creation

Heiko Paulheim

Data and Web Science Group, University of Mannheim, Germany
heiko@informatik.uni-mannheim.de

Abstract. Knowledge graphs are used in various applications and have been widely analyzed. A question that is not very well researched is: what is the price of their production? In this paper, we propose ways to estimate the cost of those knowledge graphs. We show that the cost of manually curating a triple is between $2 and $6, and that the cost for automatically created knowledge graphs is by a factor of 15 to 250 cheaper (i.e., 1¢ to 15¢ per statement). Furthermore, we advocate for taking cost into account as an evaluation metric, showing the correspondence between cost per triple and semantic validity as an example.

Keywords: Knowledge Graphs, Cost Estimation, Automation

1 Estimating the Cost of Knowledge Graphs

With the increasing attention larger scale knowledge graphs, such as DBpedia [5], YAGO [12] and the like, have drawn towards themselves, they have been inspected under various angles, such as as size, overlap [11], or quality [3]. However, one dimension that is underrepresented in those analyses is *cost*, i.e., the prize of creating the knowledge graphs.

1.1 Manual Curation: Cyc and Freebase

For manually created knowledge graphs, we have to estimate the effort of providing the statements directly.

Cyc [6] is one of the earliest general purpose knowledge graphs, and, at the same time, the one for which the development effort is known. At a 2017 conference, Douglas Lenat, the inventor of Cyc, denoted the cost of creation of Cyc at $120M.[1] In the same presentation, Lenat states that Cyc consists of 21M assertions, which makes a cost of **$5.71 per statement**. As a footnote, the development time of 1,000 person years boils down to 9.5 minutes per assertion.

[1] http://www.ttivanguard.com/conference/Napa2017/4-Lenat.pdf

Freebase has been collaboratively created by volunteers [1], and hence, its development effort is more difficult to assess. To assess the time for curating the statements in Freebase, we follow the assumption that adding a statement should be approximately as much effort as adding a sentence to Wikipedia.[2]

In [4], the time of creating the English language Wikipedia up to April 2011 has been estimated to a total of 41M working hours. At that time, Wikipedia contained 3.6M pages,[3] at an average of 36.4 sentences each [10]. This boils down to 18.7 minutes per sentence.[4] Since the majority of Wikipedians is US-based,[5] we use the US federal minimum wage of $7.25 per hour[6] as an estimate for labor cost, leading us to the cost of $2.25 per sentence.

Therefore, we also assume a cost of **$2.25 per statement** in Freebase. This is less than half of the price of a statement in Cyc – which is reasonable since Cyc was created by experts, while Freebase was created by laymen users.

In total, given that the last version of Freebase contains 3B facts [9], the cost of creating Freebase totals to $6.75B.

1.2 Automatic Creation: DBpedia, YAGO, and NELL

The estimation of effort for creating a knowledge graph automatically is different. We consider the software used for creating the knowledge graph and estimate its development effort based on the lines of code (LOC).[7] We follow the findings in [2], stating that in a software development project, an average of 37 LOC are produced by hour.[8] Furthermore, since YAGO has been developed by a German research institute and the majority of developers of DBpedia is also based in German research institutions, we use the researcher salaries proposed by the German national research funding agency DFG[9] for our cost estimates.

DBpedia is created from Wikipedia dumps by running the DBpedia Extraction Framework[10], which uses mappings to a central ontology[11] for creating the knowledge graph. They account for 4.9M and 2.2M LOC, respectively. Using the numbers above, this leads to a total development cost of $5.1M. Given the 400M statements in the English language DBpedia [11], this boils down to **1.85¢ per statement**.[12] Comparing this to the $2.25 per statement for manual curation, the automation leads to savings by a factor of around 100.

[2] *Disclaimer:* this is a debatable assumption, and it is by far not the only one in this paper.

[3] `https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia`

[4] This number may seem high, but it includes revisions and, since the measurement is based on the length of edit sessions, even research for facts to a certain extent.

[5] `https://en.wikipedia.org/wiki/Wikipedia:Wikipedians`

[6] `https://www.dol.gov/whd/minimumwage.htm`

[7] Determined using *GitHub SLOC* `https://github.com/martianyi/github-sloc`

[8] The authors measured the *total* software development cost, not only the coding.

[9] `http://www.dfg.de/formulare/60_12/60_12_en.pdf`

[10] `https://github.com/dbpedia/extraction-framework`

[11] `https://github.com/dbpedia/mappings-tracker`

[12] We do not include the cost of creating Wikipedia in the first place. Otherwise, assuming that each statement in DBpedia has its root in one infobox entry (which

YAGO is a knowledge graph which combines information extracted from Wikipedia with the ontology WordNet [7]. For a fair comparison, the development cost of WordNet is included. The YAGO codebase[13] has 1.6M LOC (including rules to map infoboxes to an ontology), which makes a total of $1.6M. WordNet itself consists of 117k *synsets*,[14] each including a gloss, and we estimate the cost of defining such a synset roughly the same as producing a Wikipedia page, i.e., $10M on top. Given that YAGO has 1.4B statements [11], this totals to **0.83¢ per statement**. Compared to the manual curation, automation leads to savings by a factor of around 250 here.

NELL is a system that learns patterns for relation extraction [8]. Its core technology encompasses 103k LOC,[15] which accounts for an estimated development cost of $109k. Furthermore, 1,467 statements are manually validated per month. Assuming that manually *validating* a statement costs as much as *creating* it, this accounts for another $376k, i.e., a total development cost of $485k. Given the size of NELL, this totals to **14.25¢ per statement**, i.e., a savings factor of 16 compared to manual curation.

2 Towards new Evaluation Metrics

Introducing cost as a measure for knowledge graph creation can also pave the way for other kinds of evaluation. For example, a new method for adding missing knowledge to a knowledge graph [9] can be inspected by cost: e.g., an approach developed by one person over half a year should add significantly more than 2,800 statements, which, according to the numbers used in this paper, would be the amount of triples that person would produce manually in that time. Furthermore, approaches that propose the creation a custom knowledge graph for improving the performance of a specific task can estimate the cost of that improvement more efficiently.

Another interesting consideration is the relation between development effort and data quality. In figure 1, we graphed the error rate of the knowledge graphs discussed in this paper against the cost per triple. While the general trend that can be observed is that triples created at higher expenses also have a higher likelihood of being correct, NELL is an outlier here, depicting a much worse relation between accuracy and cost.

3 Concluding Remarks

In this paper, we have shown estimates for the cost of the creation of popular knowledge graphs, an aspect of knowledge graph creation that is currently underrepresented in the literature. We have quantified the gain of automatic over

we could understand as a "'sentence"') in Wikipedia, the cost would always be higher than that of manual curation.

[13] `https://github.com/yago-naga/yago3`

[14] `https://wordnet.princeton.edu/`

[15] `https://groups.google.com/forum/#!topic/cmunell/aAZVG9zVwSU`

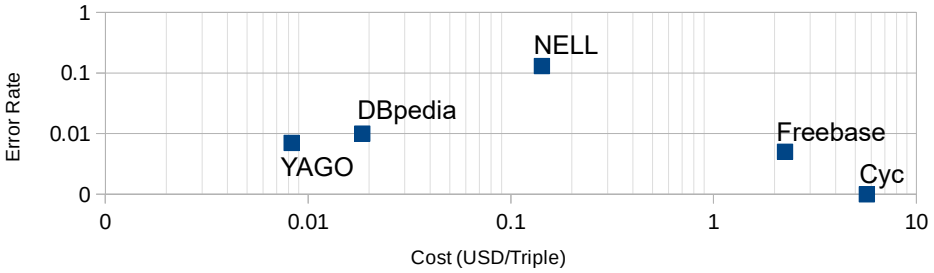

Fig. 1: Error rate (according to [3, 8]) graphed against cost per triple

manual curation (i.e., 2-3 orders of magnitude), and proposed using cost for the definition of new evaluation metrics, e.g., trading off cost for accuracy.

That being said, we are aware that many of the assumptions and approximations we took for computing those estimates are questionable (e.g., we did not consider the cost of third party software libraries used by the approaches, or the infrastructure cost), and one could have used other numbers in most of the cases. Moreover, the cost of *providing* the knowledge graphs is currently not considered. Nevertheless, we are confident that shedding light at the cost aspect of knowledge graph creation is valuable.

References

1. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD. pp. 1247–1250 (2008)
2. Devanbu, P., Karstu, S., Melo, W., Thomas, W.: Analytical and empirical evaluation of software reuse metrics. In: ICSE. pp. 189–199 (1996)
3. Färber, M., Bartscherer, F., Menne, C., Rettinger, A.: Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. Semantic Web 9(1), 77–129 (2018)
4. Geiger, R.S., Halfaker, A.: Using edit sessions to measure participation in wikipedia. In: CSCW. pp. 861–870 (2013)
5. Lehmann, J. et al.: DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. Semantic Web 6(2) (2013)
6. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
7. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM 38(11), 39–41 (1995)
8. Mitchell, T. et al.: Never-ending learning. In: AAAI (2015)
9. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web 8(3), 489–508 (2017)
10. Ringland, N., Nothman, J., Murphy, T., Curran, J.R.: Classifying articles in english and german wikipedia. In: ALTA. pp. 20–28 (2009)
11. Ringler, D., Paulheim, H.: One knowledge graph to rule them all? analyzing the differences between dbpedia, yago, wikidata & co. In: KI. pp. 366–372 (2017)
12. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In: WWW. pp. 697–706 (2007)