Generating “Who Wants to Be a Millionaire?” Questions Sets Automatically from Wikidata

Markus Wohlan, Yannik Schröder, Frank Höppner

Ostfalia University of Applied Sciences
Dept. of Computer Science, D-38302 Wolfenbüttel, Germany

Abstract Quiz shows and apps have enjoyed great popularity in recent years, which increases the demand for fresh question sets. We investigate how to derive such sets automatically from the Wikidata knowledge graph. Utilizing the graph, node connectivity and diversity, we propose measures to identify *appealing wrong answers* and rate the difficulty of a question, which is a prerequisite to compose a full question set. First results align quite well with human perception of the questions.

1 Introduction

Various TV quiz shows (such as *Who Wants To Be A Millionaire?*) and mobile quiz apps attract people’s interest. The indispensable resource to run such a service is a large set of questions that can be posed to its users. A single game requires a question set of n (say 15) diverse questions with an increasing degree of difficulty. In this paper, we investigate how Wikidata [7] may be used to generate a virtually unlimited number of such questions sets and some means to let them appear as less computer-generated as possible.

2 Related Work

While Wikidata has been used in various occasions to *answer* questions (e.g. [6,2]), only little effort is documented regarding the *creation* of questions. We have found only two theses [1,3], which consider the construction of single questions for a given topic. Compared to them, our proposal generates questions faster, combines multiple questions to a diverse question set with varying degree of difficulty and puts a greater effort on more plausible wrong answers.

3 WikiData Dump Preprocessing

The Wikidata Toolkit [4] was used to access the JSON dump (gzip 50GB). To preserve a realistic chance of answering the generated questions we removed items of certain types (such as named proteins, specific events in a series, scientific articles, items without labels, etc.) as well as properties that were too specific (such as internal Wikipedia categories or global coordinates), as in [1,3].

The filtering process was a trade off between a small memory footprint (eliminate as much unnecessary information as possible to keep the filtered graph in main memory using a modified graph library [5]) and preserving the graph's connectivity (node connections will be used for the evaluation of questions). We finally settled at a hand-crafted selection of about 250 properties (out of approx. 1100 supported Wikidata item properties).

4 Question Set Generation

Question Template. The underlying idea is to generate questions based on templates $T = (q, p)$, where q is a query (e.g. a SparQL query) and p a phrase with placeholders completed by the query result. The query in Fig. 1, may be used together with the phrase "What is ⟨predicate⟩ of ⟨subject⟩? (a) ⟨object⟩ (b) ⟨candidate⟩ ...", where placeholders such as ⟨category⟩ have to be replaced by the corresponding variable of the query (`?category`). For instance "*What is the place of birth of Angela Merkel? (a) Hamburg (b) Dresden (c) ..*". The example query, however, delivers only a single candidate for wrong answers (usually we need three) and does not deliver additional information about the connectivity of the resulting nodes. We therefore did not use a SparQL engine, but a (filtered) graph in main memory (as described in the previous section) to speed up the querying for multiple answers as well as the collection of additional connectivity information. However, the SparQL query language illustrates the idea quite well and demonstrates that other patterns may be employed easily. To avoid trivial questions, we accept questions only if the subject does not already contain the correct answer (reject "Who organizes the FIFA world cup?"). For the same reason candidate answers being equal to the subject are eliminated, too ('Germany shares border with Germany').

Selecting Appealing Wrong Answers. For any graph node n, let $I(n)$ be the set of incoming edges (statement triples with n being the object) and $O(n)$ the set of outgoing edges (statement triples with n being the subject). By π_i we denote the projection of a set of triples to their i^{th} component (1=subject, 2=predicate, 3=object) and by σ_{val} we denote the selection of triples that take a certain value as predicate. Suppose the statement (s, p, o) serves as starting point for a question and c is a (wrong) candidate answer. In order to reduce the amount of candidates being evaluated, a candidate anwer c is only taken into consideration if c shares at least one rdf:type with o. To identify good candidate answers, we

```
SELECT ?subject ?predicate ?object ?candidate WHERE {
  ?subject ?predicate ?object.
  ?object rdf:type ?category.
  ?candidate rdf:type ?category.
 NOT EXISTS { ?subject ?predicate ?candidate. }
}
```

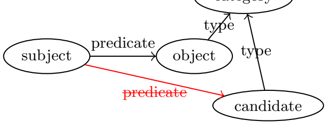

Figure 1. Example of a question template.

evaluate the similarity of c to the correct answer o and select the three candidates with highest similarity. We define the similarity between the correct answer o and candidate answer c as $\mathrm{sim}(o, c) := s(O(o), O(c))$ with $s(A, B) :=$

$$\frac{1}{5} \cdot \frac{|\pi_2(A) \cap \pi_2(B)|}{|\pi_2(A)|} + \frac{3}{5} \cdot \frac{|\pi_{2,3}(A) \cap \pi_{2,3}(B)|}{|\pi_{2,3}(A)|} + \frac{1}{5} \cdot \frac{|\pi_3(\sigma_{type}(A)) \cap \pi_3(\sigma_{type}(B))|}{|\pi_3(\sigma_{type}(A))|}$$

Thus $A = O(o)$ is the set of triples (statements) where the correct *answer* is the subject (e.g. (Hamburg,instance-of,Hanseatic City)) and likewise in $B = O(c)$ the *candidate answer* is the subject. The first term of $s(A, B)$ identifies the fraction of properties shared with the correct answer (Hamburg and Dresden share, e.g., the *population* property), the second term considers the fraction of shared properties including their value (Hamburg and Dresden are *located in time zone UTC+01:00*), and the third term the fraction of shared types only (e.g. Hamburg and Dresden are both *instance of Big City*). The second term is most specific (reflected by a higher coefficient), the first and third are more general and become useful with less-connected nodes.

Difficulty Ranking. Our idea for a difficulty ranking is based on the assumption that the difficulty of a question decreases the more familiar the contestant is with the question's components (s, p, o). To put this idea into practice, we introduce two different measures for nodes and one for properties. We measure the connectivity of a node n by $C(n) = |O(n)| + 2|I(n)|$, putting a higher emphasis on incoming edges, as they are a better indicator for a well known node within the graph. Secondly, we define a measure of homogeneity $H(n) = \frac{|O(n)|}{|\pi_2(O(n))|}$, which becomes larger the more outgoing edges share the same label. The rationale is that, say, athletes who won the same trophy several times receive higher attention in the news and are thus better known than those who have won a trophy only once. All nodes may now be ranked according to $C(n)$ and $H(n)$ and we intend to use the ranks $r(C(n))$ and $r(H(n))$ as ingredients for the final difficulty score. (The highest $C(n)$ value receives a rank of $r(C(n)) = 1$, the lowest value a rank of N with N being the number of nodes.) However, the distributions are very skewed, the first 40% of nodes share the same small connectivity value. As there will be millions of nodes that are barely known to the average contestant, we focus on the top γ% of ranks and assign a connectivity index $CI(n)$ as follows:

$$CI(n) = \begin{cases} 1 - \frac{r(C(n))-1}{\gamma N} & \text{if } r(C(n)) \leq \gamma N \\ 0 & \text{otherwise} \end{cases}$$

(and $HI(n)$ accordingly). We combine both indices to a node difficulty $DI(n) = \alpha CI(n) + \beta HI(n)$. Finally, the popularity of a property p may be evaluated by its popularity index $PI(p)$, which is simply the relative frequency of property p among all edges in the graph. The more frequently a property is used, the more likely the contestant is familiar with it. The difficulty of a question that was derived from a triple (s, p, o) is then assessed by combining the difficulties of both nodes s and o as well as the property p:

$$D(\ (s,p,o)\ ) = \frac{2DI(s) + PI(p) + DI(o)}{4}$$

The subject receives a higher weight as it is most specific for the question. Different choices for the parameters α, β and γ will be evaluated in section 5.

Diversified Question Set. The algorithm generates a large number of questions by picking a random node as starting point and then creating a question as described above. The degree of difficulty is evaluated and three lists of easy, medium and difficult questions are maintained. A final set of n diverse questions is generated by picking $\frac{n}{3}$ questions from each list, while taking care that a newly selected question uses a different property and subject.

5 Evaluation

Performance. The final graph consisted of 11M nodes and 57M edges. Its serialization to disk occupied 1.2GB; restoring the graph from disk takes 100 seconds (Intel i7 8700k, 32 GB). The program may generate thousands of questions per minute, but the time to generate a single question depends on the number of candidate answers: for every candidate answer we apply the similarity measure, which is cumbersome for persons as there are 3.6M persons in the graph (calculations take up to 15s then). There are ways to concentrate quickly on the most relevant nodes and prune the search, but they have not yet been elaborated.

Candidate Answers. Table 1 shows a few questions that were automatically generated by the system. For instance, question (c) asks for the title from the Lord of the Rings trilogy. If we were using the *instance-of* relationship only, any book title would do as a candidate answer. But the selected wrong answers consist of other book titles by the same author (Tolkien), although the author is not part of the question. Similarly, question (p) asks for the inventor of the special theory of relativity and the candidate answers include other well-known german physicists of that time.

Degree of Difficulty. To test our proposals for the degree of difficulty, we selected 18 automatically generated questions and asked 139 people (of which roughly 60% were computer science students) to assess the degree of difficulty in three levels (1:easy, 2:medium, 3:hard). Fig. 1 shows an excerpt together with the difficulty ranking obtained from averaging the responses. We experimented with parameter values $\alpha \in [1, 4]$, $\beta \in [0, 3]$ and $\gamma \in [0.02, 0.2]$ (224 configurations) and compared the rank (obtained from our difficulty measure) via Spearman correlation. Regarding γ, the best results were obtained for values close to 0.1. For $\alpha = 3, \beta = 1$ we obtained correlations of 0.5 (maximum) and 0.347 (on average). A closer inspection revealed that this score is mainly due to two questions (j,p), where especially the computer science students found the questions much

Table 1. Automatically generated questions with an empirically (ED) and algorithmically assessed difficulty rank (AR). Questions are ranked from 1 (easiest) to 18 (hardest) using $\alpha = 2, \beta = 1, \gamma = 0.09$ resulting in a correlation value of 0.48.

	Question	ED	AR
a	Which is the highest mountain of Tanzania? Mawenzi/Kibo/Tupungato/Calbuco	17	11
b	In which field does Angela Merkel have a degree ? natural science/genetics/pedagogy/physics	5	6
c	Which of the following is part of the Lord of the Rings trilogy? Smith of Wootton Major/The Adventures of Tom Bombadil/Unfinished Tales/The Return of the King	6	3
d	Which of these programming languages influenced C++? Fortran/BETA/Simula/Dart	13	9
e	What is the basic form of government in Austria ? constitutional monarchy/democracy/monarchy/federal parliamentary republic	7	4
f	In which country was Pink Floyd founded? USA/GB/Australia/Germany	8	1
g	Which sports team is Stephen Curry currently playing for? Los Angeles Clippers/Los Angeles Lakers/Milwaukee Bucks/Golden State Warriors	15	16
h	Who is one of the inventors of the TV series Two and a Half Men? Peter S. Fischer/Alan Taylor/Lee Aronsohn/Chris Gerolmo	11	8
i	Who wrote the book series Harry Potter? Neil Gaiman/Ursula K. Le Guin/Terry Pratchett/J. K. Rowling	1	2
j	Which unit measures the frequency? Bel/Radiant per Second/Newton/hertz	3	18
k	Which position is held by Bernie Sanders currently? US State Senator/Majority Leader in the US-Senate/US Minister of Trade/Member of US Senate	12	15
l	Who owns the F.C. Liverpool? Alibaba Group/Fenway Sports Group/General Motors/Pirelli	16	12
m	Which genre is assigned to Oasis? post-grunge/sadcore/shoegazing/indie pop	10	7
n	Who is the current German head of state? Heinz Riesenhuber/Angela Merkel/Hans-Peter Friedrich/Frank-Walter Steinmeier	2	5
o	Which position is held by Muhammadu Buhari currently? President of Nigeria/President of Gambia/President of Ecuador/President of South Africa	18	17
p	Who is considered the founder of the special theory of relativity? Niels Bohr/Albert Einstein/Max Planck-/Erwin Schrödinger	4	14
q	Which river flows into Lake Constance? Salzach/Yellow River/Dornbircher Ach/Inn	9	10
r	In which sport does the Stanley Cup take place? Weightlifting/Biathlon/Ice Hockey/Rowing	14	13

easier than other participants. When excluding these two questions only, the Spearman correlation coefficient rises to 0.9 (resp. 0.72 on average).

6 Conclusions

We have presented an approach to automatically generate question sets for quizzes. The selected wrong answers mimic hand-crafted options very well. First results on evaluating the difficulty score show promising results for many questions but also occassional misjudgements with the empirically assessed difficulty. One has to keep in mind, however, that the whole approach assumes a correlation of the Wikidata content with the general knowledge of the contestants (which will not hold for an arbitrary audience).

References

1. F. Bissig. Drawing questions from wikidata, 2015. ETH Zürich. Bachelor's Thesis.
2. D. Diefenbach, K. Singh, and P. Maret. Wdaqua-core0: A question answering component for the research community. In *Semantic Web Evaluation Challenge*, pages 84–89. Springer, 2017.
3. S. Geng. Drawing questions from wikidata, 2016. ETH Zürich. Bachelor's Thesis.
4. M. Krötzsch. Wikidata toolkit. https://www.mediawiki.org/wiki/Wikidata_Toolkit.
5. D. Michail, J. Kinable, B. Naveh, and J. V. Sichi. Jgrapht–a java library for graph data structures and algorithms. *arXiv preprint arXiv:1904.08355*, 2019.
6. H. Thakkar, K. M. Endris, J. M. Gimenez-Garcia, J. Debattista, C. Lange, and S. Auer. Are linked datasets fit for open-domain question answering? a quality assessment. In *Web Intelligence, Mining and Semantics*, pages 19–30. ACM, 2016.
7. D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.