Published in the Proceedings of the 16th Joint ACL - ISO Workshop on Interoperable SemanticAnnotation (ISA-16), 12 may 2020, Marseille France, which sould be cited to refer to this work.

A Consolidated Dataset for Knowledge-based Question Generation using Predicate Mapping of Linked Data

Johanna Melly[*]
Cortexia S.A.
Route de Vevey 105a
1618 Châtel-Saint-Denis
Switzerland
johanna@melly.me

Gabriel Luthier
HEIG-VD / HES-SO
Route de Cheseaux 1, CP 521
1401 Yverdon-les-Bains
Switzerland
gabriel.luthier@heig-vd.ch

Andrei Popescu-Belis
HEIG-VD / HES-SO
Route de Cheseaux 1, CP 521
1401 Yverdon-les-Bains
Switzerland
andrei.popescu-belis@heig-vd.ch

Abstract

In this paper, we present the ForwardQuestions data set, made of human-generated questions related to knowledge triples. This data set results from the conversion and merger of the existing SimpleDBPediaQA and SimpleQuestionsWikidata data sets, including the mapping of predicates from DBPedia to Wikidata, and the selection of 'forward' questions as opposed to 'backward' ones. The new data set can be used to generate novel questions given an unseen Wikidata triple, by replacing the subjects of existing questions with the new one and then selecting the best candidate questions using semantic and syntactic criteria. Evaluation results indicate that the question generation method using ForwardQuestions improves the quality of questions by about 20% with respect to a baseline not using ranking criteria.

Keywords: Question generation, linked data, knowledge triples, semantic mapping.

1. Introduction

Question generation from linked data is a promising approach for producing large corpora of questions and answers. A primary use of these corpora is for training and evaluating question answering systems (Duan et al., 2017), while other uses are for education (Pham et al., 2018), tutoring (Su et al., 2019), or entertainment. Automatic question generation can be based on texts or on large repositories of linked data. In the latter case, an initial set of human-generated questions is often necessary to generate new ones, but such data sets are strongly related to specific linked data formats, and are difficult to port to new repositories.

In this paper, we present the ForwardQuestions corpus of human-generated questions associated to knowledge triples from the Wikidata knowledge base. We constructed this corpus by converting and merging two partially overlapping corpora of questions, SimpleDBPediaQA and SimpleQuestionsWikidata, which were separately derived from subsets of the SimpleQuestions corpus. These three data sets are respectively based on DBpedia, Wikidata and Freebase, but the latter resource is no longer available.

Specifically, we enriched SimpleQuestionsWikidata with a substantial number of questions from SimpleDBPediaQA, by converting DBpedia predicates to Wikidata ones and keeping only the 'forward' questions, given our final goal of quiz generation.[1] The overlap between these two resources is only of 32%, showing that the resulting data set has considerable novelty. As a result, we make available, under the Creative Commons Attribution license (BY), the ForwardQuestions corpus of 38k questions related to 94 different Wikidata predicates.[2]

Furthermore, we show how ForwardQuestions can be used to generate new questions from previously unseen triples, by replacing the subjects of existing questions with new ones, and then ranking candidate questions on semantic and syntactic criteria. The questions can be used, for instance, in a chatbot that generates quizzes on any topic indicated by a user, thanks to a strategy for selecting relevant triples from Wikidata. The evaluation results with human subjects who rate the quality of the questions show that the best questions generated by our method reach about 80% approval, of which 10 points are due to the question ranking method.

The paper is organized as follows. In Section 2, we review related work and present the SimpleQuestions, SimpleDBpediaQA and SimpleQuestionsWikidata resources. In Section 3 we explain how the latter two data sets were converted and merged into the new ForwardQuestions data set. In Section 4, we describe our template-based question generation method and the semantic and syntactic ranking strategies, used in a chatbot presented briefly in Section 5. In Section 6, we define the evaluation protocol and quantify the improvements brought by our resource and question generation method.

2. Relation to Previous Work

Question answering (QA) has been extensively researched in the past. Many methods use textual documents to find answers, while others consider knowledge bases, such as large sets of knowledge triples (*subject*, *predicate*, *object*). QA over knowledge bases requires data sets with questions and their answers, for training and evaluation. For instance, the data sets used for the QALD evaluations (Unger et al., 2016) typically include hundreds of questions, most

[*] Work conducted while the first author was at HEIG-VD.

[1] 'Forward' questions are those bearing on the object of a (*subject*, *predicate*, *object*) triple. They typically have smaller sets of correct answers than 'backward' questions (see 2.2 and 3.2). Note that 'subject' and 'object' refer to the entities appearing in first and third position in the triples, but depending on how the predicate is expressed in a sentence, their grammatical functions can be reversed.

[2] github.com/johannamelly/ForwardQuestions.

of which can be answered based on a single triple, while others require a combination of triples.

In the past, triple stores such as Freebase (with around 40 million entities) or DBpedia (an order of magnitude smaller) have been used to design QA systems (Bast and Haussmann, 2015). The termination of the Freebase repository raised the question of resource conversion to DBpedia, or to the more recent Wikidata triple store,[3] which is a knowledge graph derived from Wikipedia infoboxes and allows data querying with SPARQL (Malyshev et al., 2018). The main challenge remains however the generation of questions from triples, which is a costly process that has been partially automated in the past, as we briefly review hereafter.

2.1. Automatic Question Generation

Existing methods for question generation start either from textual data or from knowledge triples. Heilman and Smith (2010) defined rules for syntactic transformation of declarative sentences into questions, which were then ranked by a logistic regression model, reaching an acceptance rate of about 50% for the 20% top-ranked questions. Chali and Hasan (2015) used named entity and predicate-argument information to generate questions, but evaluated them only automatically. They used LDA to estimate topic relevance, and syntactic tree kernels for grammaticality judgments. A rule-based approach to generate questions from relative subordinate sentences extracted from Wikipedia was proposed by Khullar et al. (2018). This method generated better questions than Heilman and Smith, but relied crucially on the availability of relative pronouns and adverbs.

More recent models attempt to generate questions from sentences using deep neural networks, e.g. starting from a sentence and the intended answer word (Sun et al., 2018; Zhao et al., 2018). Currently, their accuracy on long sentences such as those from Wikipedia is sufficient for quiz generation, especially since they were only evaluated by quantitative comparisons to the SQuAD data set (Rajpurkar et al., 2016)). Recent improvements aim at predicting the question type from the answer and then add this prediction to the neural generator (Zhou et al., 2019).

Serban et al. (2016) proposed two methods for question generation from Freebase. The neural network approach used TransE multi-relational embeddings (Bordes et al., 2013) and leveraged conditional language generation models. They generated a corpus of 30 million questions based on Freebase triples, which were evaluated with the BLEU metric and partly with human judges. Their template-based baseline model scored only slightly below, but is applicable also when TransE embeddings are not available – hence, it is the starting point of our present proposal.

2.2. Human-generated Questions from Triples

The SimpleQuestions dataset (Bordes et al., 2015)[4] features 108,442 questions in English obtained through a crowdsourcing platform. Each question is accompanied by the knowledge triple from Freebase on which it is based, which also provides its answer. For instance, one question is "What does Jimmy Neutron do?", and the triple ('Jimmy Neutron', 'fictional character occupation', 'inventor') indicates that the answer is "inventor".

An important distinction introduced by SimpleQuestions, coming from the observation of human-generated questions, is between forward and backward questions. A forward question bears on the object of a triple, while a backward one bears on its subject, and is often formulated using passive voice. For instance, from the triple ('The Dishwasher: Dead Samurai', 'publisher', 'Xbox Game Studios'), someone generated the question "What company published The Dishwasher: Dead Samurai?", which is a forward one. However, from the triple ('Rampage', 'publisher', 'Midway Games'), someone wrote the question "What game is published by Midway Games?", which is a backward one. One reason to consider this distinction is that predicates do not appear in both active and passive forms in the triple store, so questions are allowed to bear on the subject or or object of a triple.

Due to the termination of Freebase, the triples of a subset of questions from SimpleQuestions have been converted to DBpedia triples, resulting in the SimpleDBpediaQA data set (Azmy et al., 2018).[5] Two formatted questions from this data set are presented in Table 1. 'Query' is the original question formulated over a Freebase triple, whose former predicate URL is given under 'Freebase Predicate'. 'Subject' points to the URL of the concept on DBpedia. There are three subfields under 'predicate list': the DBpedia URL of the predicate, the direction of the question (forward or backward), and a constraint on the expected answer type for backward questions.

A subset of SimpleQuestions different from the one above has been converted to Wikidata triples, resulting in the SimpleQuestionsWikidata set (Diefenbach et al., 2017).[6] The resource is available as a text document, formatted as shown in Table 2. Each line has four tab-separated fields, containing the Wikidata identifier of the triple's subject, predicate, and object, and the question itself. The predicates of forward questions have Wikidata identifiers prefixed with 'P', e.g. 'P413' refers to the Wikidata property at `www.wikidata.org/wiki/Property:P413` with the English label "position played on team / speciality". Backward questions are indicated by predicates whose initial letter was changed from 'P' to 'R', as in the third example from Table 2, where 'R509' indicates the fact that the 'P509' property ("cause of death", `www.wikidata.org/wiki/Property:P509`) holds between the object and the subject and not vice-versa, and that the actual triple in Wikidata is (Q6371569, R509, Q12152), "Karl Anton Rickenbacher died of myocardial infarction."

Finally, a smaller set of about 700 questions collected from users over Wikidata triples is also available as the WDAquaCore0Wikidata set (Diefenbach et al., 2017).[7]

[3] `www.wikidata.org`

[4] Part of the bAbI evaluation tasks from Facebook Research: `research.fb.com/downloads/babi/`.

[5] `github.com/castorini/SimpleDBpediaQA`

[6] Data set available at `github.com/askplatypus/wikidata-simplequestions`.

[7] `github.com/WDAqua/WDAquaCore0Questions`.

ID	00035
Query	what is the place of birth of sam edwards?
Subject	`http://dbpedia.org/resource/Sam_Edwards_(physicist)`
Freebase Predicate	`www.freebase.com/people/person/place_of_birth`
Predicate List	
Predicate	`http://dbpedia.org/ontology/birthPlace`
Direction	forward
Constraint	null
ID	00042
Query	which home is an example of italianate architecture?
Subject	`http://dbpedia.org/resource/Italianate_architecture`
Freebase Predicate	`www.freebase.com/architecture/architectural_style/examples`
Predicate List	
Predicate	`http://dbpedia.org/ontology/architecturalStyle`
Direction	backward
Constraint	`http://dbpedia.org/ontology/ArchitecturalStructure`

Table 1: Examples of SimpleDBpediaQA entries: a forward and a backward question.

Subject	Pred.	Object	Question
Q2747238	P413	Q5059480	what position does carlos gomez play?
Q1176417	P136	Q37073	what type of music does david ruffin play
Q12152	R509	Q6371569	which swiss conductor's cause of death is myocardial infarction?

Table 2: Examples of SimpleWikidataQA entries.

2.3. Comparison of SimpleDBpediaQA and SimpleQuestionsWikidata

From the 108,442 entries in SimpleQuestions, 43,086 were included in SimpleDBpediaQA, while 49,202 were included in SimpleQuestionsWikidata. There is therefore a potential to select more of the original questions for inclusion in ForwardQuestions.

Questions in SimpleDBpediaQA are not accompanied by the object of their triple, which means that their correct answer cannot be verified directly from the data set, unlike those from SimpleQuestionsWikidata, as it appears when comparing Table 1 with Table 2. This is not a major problem, nevertheless, because: (1) in general, the correct answer may not be unique even if the question is based on a single triple (e.g. "who are the children of Barack Obama?"), so the underlying triple store is still needed to verify the answer; (2) for our intended use, the questions from the database are only used as templates to generate new questions from new triples (see Section 4), therefore the objects of the original triples are never needed.

Qualitatively, the questions in SimpleDBpediaQA cover a smaller range of predicates than those in SimpleQuestions-Wikidata, and contain fewer questions per triple. The latter set uses Wikidata predicates, which are often more fine-grained than the DBpedia ones (for instance distinguishing 'father' and 'mother' where DBpedia has only 'parent'). Another qualitative observation is that SimpleDBpediaQA contains a somewhat larger proportion of triples that are not useful for question generation, as they correspond to various numeric identifiers of entities in 3rd party repositories.

3. The ForwardQuestions Data Set

3.1. Motivation for ForwardQuestions

DBpedia or Wikidata triples represent only small subsets of the knowledge embodied in Wikipedia, which is why it may seem that generating questions directly from Wikipedia sentences could lead to more varied questions (Heilman and Smith, 2010; Chali and Hasan, 2015). However, our pilot experiments in this direction pointed to strong limitations. For instance, we considered identifying patterns such as *verb + named entity* in sentences from Wikipedia, and then reversing them to build a question, e.g. from "World War II ended in 1945" we aimed to derive "When did World War II end?" However, several difficulties appeared: (1) the VB+NE pattern also applies to relative clauses (e.g. "Billie Joe Armstrong took two years to write American Idiot") from which questions cannot be easily generated; (2) the interrogative word is hard to predict; (3) pronouns lead to unintelligible questions; (4) answers should not be limited to named entities.

Therefore, we turned to the use of knowledge triples, following the template-based baseline proposed by Serban et al. (2016). Triples enable a straightforward generation method: transform the triple (*subject*, *predicate*, *object*) into a question bearing on the 'predicate' property of the 'subject', knowing that 'object' one of the correct answers. For instance, from ('Harry Potter', 'mother', 'Lily Potter') one can construct "Who is the mother of Harry Potter?". Note that 'subject' and 'object' do not necessarily have these grammatical functions in the sentence from which the triple was generated, as these functions depend on the form of the predicate. In the above example, the natural formulation "Lily Potter is the mother of Harry Potter" actually reverses these roles.

It appeared however that, in general, the specific wordings describing the subject, the predicate, and the expected type of answer are difficult to generate correctly. For instance, from ('Harry Potter', 'composer', 'John Williams'), the derived question "Who is the composer of Harry Potter?" is incorrectly formulated – a correct version is "Which composer wrote the music for the film Harry Potter?". This is why we use template-based generation from questions written by humans in the ForwardQuestions data set.

3.2. Construction of the Data Set

SimpleDBpediaQA and SimpleQuestionsWikidata are both subsets of SimpleQuestions. Hence, they have a certain amount of overlapping questions, but also some that are specific to each set. Therefore, merging the two subsets results in a larger one, named ForwardQuestions. Given their different formats, we decided to convert them to a new format, which preserves all the information from both data sets. This format also includes a template derived from each question, which can be used for question generation.

The main added value of the resource is the conversion of DBpedia predicates to Wikidata ones, resulting in a resource that is enriched with respect to both of its sources, although it still cannot recover all original SimpleQuestions items based on Freebase predicates, as not all of them have mappings in Wikidata.

We do not include backward questions, because they are not convenient for generating new questions. Indeed, they typically accept a much larger number of possible answers than forward ones, and may therefore appear as either too open or too easy. Indeed, asking about a property of a subject makes a good question, as subjects have a limited number of properties. However, asking which subjects have a given property is generally not a good question because the same property can potentially apply to a very large number of subjects. For instance, "In what country is Geneva?" is a good question, while "What city is in Switzerland?" is not, although both are based on ('Geneva', 'country', 'Switzerland'). Given our goal of quiz generation, we exclude backward questions, of which there are 14,632 in SimpleDBpediaQA (34%) and 12,420 in SimpleQuestionsWikidata (25%).

We now describe the mapping process for the SimpleQuestionsWikidata entires, and discuss afterwards the differences with SimpleDBpediaQA. We process each (*subject*, *predicate*, *object*, *question*) line as follows. We first exclude predicate starting with an 'R' (backwards question). Then, we query the Wikidata API to find the English labels of the subject, predicate, and object, and exclude questions for which the subject or the predicate cannot be found. Next, we build a template from each question, for question generation. We identify the position of the subject in the question, and replace it with the string '<placeholder>'. As different referring expressions were sometimes used for subjects, we allow for some flexibility when matching subject labels. For instance, we replace dashes, apostrophes and non-ASCII characters with white spaces, to increase the number of matches. Still, due to misspellings, simplifications, confusion of subject or object, or insertion of external knowledge about the subject, no match can be identified for about 4% of the questions, which are excluded.

A similar conversion was performed for SimpleDBpediaQA entries, but this required a mapping of DBpedia predicates to Wikidata ones, which we explain in the next subsection. The subjects were also mapped to their Wikidata equivalent, using requests to the APIs and matching the English labels of the entities (as stated above, objects are missing in this case).

As a result, each item appears in ForwardQuestions as follows:

- Question: full text and template based on it;
- Subject: label (English words) and Wikidata code;
- Predicate: label and Wikidata code;
- Object: if available, label and Wikidata code.

3.3. Converting DBpedia Predicates to Wikidata

Predicates from DBpedia appearing in SimpleDBpediaQA questions must be mapped to Wikidata ones before inclusion in ForwardQuestions. For instance, 'playerPosition' from DBpedia must be mapped to 'position played on team / speciality' (P413) in Wikidata. For some of the 6,236 Wikidata predicates, their equivalent in DBpedia is specified, but this happens only for 177 predicates out of the 365 ones appearing in SimpleDBpediaQA, leaving 188 predicates with no known DBpedia equivalents.

We mapped these 188 remaining predicates using two approaches. Firstly, we looked for partial matches of the DBpedia labels with those from an online list of 1,872 Wikidata predicates with labels.[8] For example, for DBPedia's 'populationTotal' predicate, we could easily find the equivalent Wikidata predicate 'population'. Secondly, for non-matched labels, we performed a manual word-based search on Wikidata and selected the closest matching predicate.

The final mapping of predicates is provided with the ForwardQuestions data set in the `mapping.json` file of the Github repository (footnote 2). Each entry includes the DBpedia name and the matching Wikidata code, e.g. ('primeMinister', 'p6'). The first 177 predicates are those with explicit DBpedia equivalents in Wikidata, while the following 188 ones are those we mapped. In fact, we mapped many more predicates than those actually appearing in the selected questions, in anticipation of future needs.

3.4. Results

To sum up, we merged the forward questions from SimpleDBpediaQA and SimpleQuestionsWikidata, discarded backwards ones, removed duplicates (32% of the SimpleDBpediaQA), converted DBpedia predicates to Wikidata ones, and generated question templates by replacing subjects with <placeholder>. The ForwardQuestions data set contains 38,480 questions, having in total 94 different predicates. The various filtering operations, especially backward question removal and subject matching, have led us to keep only about 35% of the original SimpleQuestionsWikidata entries.

[8] Found at `quarry.wmflabs.org/run/45013/output/1/json`. The gathering of all predicates in one list simply facilitated our search.

The most frequent predicate in ForwardQuestions is ‘genre’, which appears in more than 8,000 questions. Its meaning is quite general (akin to ‘type’ or ‘category’) and it can appear in triples concerning movies, books, music albums, artists, etc.[9] The next predicates by decreasing frequency are ‘place of birth’, ‘country of citizenship’, ‘sex or gender’, and ‘position played on team / speciality’. The full list with frequencies is provided with the data set.

4. Question Generation from Triples

The ForwardQuestions is intended to help with the generation of new questions, from knowledge triples not included in the set. We propose a method inspired from the rule-based baseline from Serban et al. (2016), with the following differences. Their data set used Freebase, but we use Wikidata as the underlying triple store: our observations show that these predicates are often more precise, and specify sufficiently the type of the expected answer. For this reason, we created for each item in ForwardQuestions a template with a generic placeholder for the subject, unlike Serban et al. (2016) who use type-specific placeholders such as <location placeholder>, which strongly reduces the number of questions available for generation. It is still an open question whether the size of ForwardQuestions allows the training of deep learning models; for the time being, we use the following template-based generation method.

We generate a sample set of questions using 20 randomly selected templates among all those having the same predicate as the given triple, by replacing the placeholder of the question with the subject of the triple.[10] We then rank the questions using semantic similarity (4.1) and a language model (4.2).

The main issue to address can be illustrated with the following example. If we use a template such as “What kind of music does <placeholder> play?”, derived from (‘John Duffey’, ‘genre’, ‘bluegrass music’), but we want to generate a question based on the new triple (‘Claude Monet’, ‘genre’, ‘portrait’), then we obtain the question “What kind of music does Claude Monet play?”, with the expected correct answer being ‘portrait’. The question is incorrect because the rendering of the predicate ‘genre’ in the initial question is too specific and incompatible with the sense of the new triple. Alternatively, the reference to the subject in the template can also be too specific, e.g. if we use the template “What genre is the tv program <placeholder>?” with the triple above, we obtain the incorrect question “What genre is the tv program Claude Monet?”

4.1. Ranking with semantic similarity

To avoid the issues exemplified above, we use semantic similarity between the word vectors provided by the word2vec library (Mikolov et al., 2013).[11] We compare the average of the word vectors from the opening Wikipedia paragraph of the subject with the average of word vectors of the question template using cosine similarity. We also compute the similarity between the words of the template and those from the opening Wikipedia paragraph of the object, and retain the maximum of the two similarities as the semantic compatibility score of the question and the triple.

4.2. Ranking with a language model

We observed that some ungrammatical questions obtained high semantic compatibility scores. Our second goal is thus to filter them out, using the KenLM language modeling software (Heafield et al., 2013)[12] with a language model for English provided by Zamia.[13] The perplexity score of the language model for the full question provides an estimate of the well-formedness of the question, i.e. a syntactic fluency score.

Therefore, for a given triple, we combine the semantic and fluency scores, giving more weight to the first one, and select the question which has the highest average score.

5. Use of Questions for a Quiz Chatbot

The method for generating questions from arbitrary triples, using ForwardQuestions, can be used to build a quiz chatbot which prompts the user to select a topic. This topic is matched to a Wikipedia page, from which we find a reasonable number of interesting triples, from which questions can be generated as explained above. The chatbot proposes the questions one by one to the user, and compares their answers to the expected ones.[14]

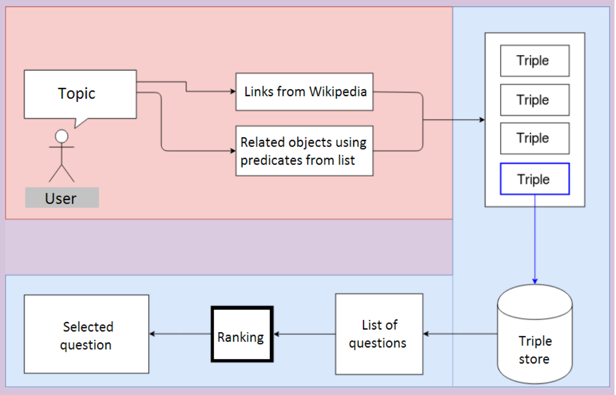

Figure 1: Overview of the quiz generation chatbot.

5.1. Selection of Triples for a Given Topic

The Wikidata entry of the topic indicated by the user may contain a large triple set, but all of them have the topic itself as the subject. This reduces the diversity of questions, as all of them will bear on some property of the subject.

[9] For instance, the triple (‘John Duffey’, ‘genre’, ‘bluegrass music’) has the associated question “What kind of music does John Duffey play?”

[10] We filter out any parentheses from the subject’s description, e.g. ‘The Danton (1983 film)’ is reduced to ‘The Danton’.

[11] As implemented in the Gensim package (radimrehurek.com/gensim/models/word2vec.html) with Google’s pre-trained model. As the templates are very short, and for simplicity reasons, we did not experiment with more elaborate paragraph representation models.

[12] github.com/kpu/kenlm

[13] goofy.zamia.org/zamia-speech/lm/, large model

[14] In other words, there are no follow-up questions, unlike Su et al.’s (2019) system, based on an ontology with RDF triples restricted to the dialogue domain.

For instance, if the topic is 'Queen', the British rock band (Q15862), then all questions from triples on their Wikidata page will have 'Queen' as the subject: "When was Queen founded?", "What country is Queen from?", "What is the music genre of Queen?", etc.

To increase diversity, we select additional triples from entities that are related to the main one, as follows. First, we select a list of Wikidata predicates that tend to connect the subject to meaningful objects, such as 'has part' (P527). Using the Wikidata entries of these objects, we select additional triples. For instance, from ('Queen', 'has part', 'Freddie Mercury') we infer that 'Freddie Mercury' is a related entity, and find the triple ('Freddie Mercury', 'religion', 'Zoroastrianism') which allows us to build the question: "What was Freddie Mercury's religion?" for the topic 'Queen'. We identified about 50 such predicates that allow the extension of the topic.

However, some notions such as 'Rock Music' or 'Cooking' have few properties on their Wikidata pages, which is why we also use a second strategy, relying on the Wikipedia page of the subject. We use the "See also" or "Related topics" section to retrieve related subjects, but these sections are not always present and contain only a small list of subjects, of variable relevance. Therefore, we consider all hyperlinks from the page to other Wikipedia entries, and select the 20 first random subjects whose Wikidata entries contain at least 25 triples (so that each related subject has sufficient substance).

In addition, we merge the objects of triples that have the same subject and predicate, to obtain the list of all possible correct answers. For instance, from ('Barack Obama', 'child', 'Malia Obama') and ('Barack Obama', 'child', 'Sasha Obama'), we obtain a single triple for question generation, with a list of two acceptable answers.

Finally, for any topic selected by the user, we randomly select 10 subjects found with the first method, and 20 found with the second one, with the aim of obtaining about 100 knowledge triples from which questions are generated.

5.2. Implementation of a Chatbot Prototype

We implemented a chatbot demo with the following components. Actions on Google[15] is the front-end proposed by Google to create apps for its Google Home smart speaker. Dialogflow[16], which is connected to Actions, enables the design of simple dialogue models. The backend, running on one of our servers, is coded in Python with the Flask web development framework. As we found that our question generation is too slow to run in real time (taking several minutes on a mid-range computer, especially due to querying Wikipedia pages), we generated questions offline for several subjects ("Olympic Games", "Politics of the United States", "Rock music", "Super Mario Bros.", "Switzerland", "The Legend of Zelda", and "World War II"). The chatbot proposes to the user three randomly selected topics, among which one must be chosen. Sample questions (Q) and their correct answers (A) for "World War II" are:

- Q: Which country was involved in the Eastern Front?
  A: Nazi Germany, Soviet Union, . . .
- Q: Who was one of the major figures in the Attack on Pearl Harbor?
  A: Husband Edward Kimmel, Mitsuo Fuchida, . . .
- Q: Who was the developer for A6M Zero?
  A: Mitsubishi Heavy Industries
- Q: What was the cause of death for Adolf Hitler?
  A: shot to the head, suicide by shooting

6. Evaluation of the Questions

The following evaluation protocol is targeted at the quality of the questions and their correct answer(s), and not at the usability of the chatbot, which depends also on the dialogue model and the speech recognition system.[17] For each triple and question, we asked human judges to rate the following quality aspects:

1. **Triple**
   - *Importance of predicate and object.* How interesting are the predicate and the object? For instance, for the triple ("North America", "located in time zone", {"Hawaii-Aleutian Time Zone", . . .}), the predicate and its value do not appear to be interesting.
2. **Question**
   (a) *Specification of the subject.* For instance, in the question "Who was responsible for the music in the film Super Mario Bros.?", the specification of the subject is incorrect given that Super Mario Bros. is a video game, not a film.
   (b) *Specification of the object.* For instance, in "Which city in Scotland did J. R. R. Tolkien come from?", the specification of the expect answer (the triple's object) is wrong because Tolkien is from Birmingham, which is not in Scotland.
   (c) *Formulation of the question.* Is the question understandable and well-formed in English? This includes spelling mistakes. For instance, "Who was the published the game Harry Potter?" is poorly formulated.
   (d) *Correctness of the expected answer..* For instance, for "Who was the film Harry Potter and the Deathly Hallows based on the story by?", the expected answer is 'Steve Kloves' (author of the screenplay), but one may estimate that 'J. K. Rowling' should be the correct answer, as she is the author of the original book.
3. **Overall:** is this a good item for a quiz?

We asked four persons not familiar with the project to perform the following comparison. Given a knowledge triple, we show them the best question found by our method and the worst one (also according to our method) from a random

[15] console.actions.google.com

[16] dialogflow.com

[17] The complete chatbot could be evaluated with the recent set of 17 metrics made available by Csáky et al. (2019).

subset of 20 questions generated for the triple. The human must rate each of the questions, without knowing their origin, on a five-point scale for each criterion. The goal is thus to measure the improvement brought by our method, with respect to a rather poor question, but still much stronger than the worst of all questions. For a set of 105 triples and 210 questions, we obtained 472 ratings.

	Score	
Criterion	**Best question**	**Poor question**
Predicate + object	4.12	
Subject specification	4.51	3.51
Object specification	4.02	3.55
Question formulation	4.19	4.25
Answer correctness	3.80	3.40
Overall quality	3.35	2.84

Table 3: Average scores of the best question and of a random poor question on a five-point scale.

The results are presented in Table 3. On all but one dimension, the best question shows clear improvement with respect to the poor one. The best questions score below poor ones regarding "formulation", but both scores are in fact rather high. With an overall quality of 3.35 out of 5, the questions are satisfactory, but there is also potential for progress.

The largest improvement brought by our method (1 point out of 5) is for the specification of the subject, which is excellent for the best questions. The specification of the object (expected answer) is also improved. This was indeed one of our main goals, given that user-generated questions often include specifiers which become incorrect when the subject is replaced with another one. The improvement of the expected answer is quite similar to the one for the specification of the object, as these two elements are closely related. Finally, the relevance of the predicate + object is quite high, showing that the triple selection method is effective.

7. Conclusion

In this paper, we presented the ForwardQuestion data set, which we make available under the same CC-BY licence. The data set results from the conversion and combination of the SimpleDBpediaQA and SimpleQuestionsWikidata datasets, in particular by mapping predicates from Freebase to Wikidata. The 38,480 questions of the data set are accompanied by templates where the subject is replaced by a placeholder, in preparation for question generation that can be used in a quiz chatbot. The difficulties of triple conversion and predicate mapping strongly point to the need for interoperable semantic annotation in the realm of knowledge-based question generation.

In future work on quiz generation, we aim to improve the relevance of the triples selected for a topic, as well as the diversity of the questions. While the size of the data set remains modest for use with deep learning generation methods, the triples could be used in conjunction with a pre-trained language model such as GPT-2 (Radford et al., 2019) or CTRL (Keskar et al., 2019), to serve as adaptation data for neural question generation conditioned on the triples.

Acknowledgments

We thank the University of Applied Sciences of Western Switzerland (HES-SO) for the PLACAT grant that partly supported this work (AGP n. 82681), as well as the evaluators for the time they devoted to rating the questions submitted to them.

8. Bibliographical References

Azmy, M., Shi, P., Lin, J., and Ilyas, I. (2018). Farewell Freebase: Migrating the SimpleQuestions dataset to DBpedia. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2093–2103, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Bast, H. and Haussmann, E. (2015). More accurate question answering on freebase. In *Proceedings of the Conference of Knowledge Management (CIKM'15)*, pages 1431–1440, Melbourne, Australia.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2787–2795.

Bordes, A., Usunier, N., Chopra, S., and Weston, J. (2015). Large-scale simple question answering with memory networks. *CoRR*, abs/1506.02075.

Chali, Y. and Hasan, S. A. (2015). Towards topic-to-question generation. *Computational Linguistics*, 41(1):1–20.

Csáky, R., Purgai, P., and Recski, G. (2019). Improving neural conversational models with entropy-based data filtering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5650–5669, Florence, Italy. Association for Computational Linguistics.

Diefenbach, D., Tanon, T. P., Singh, K. D., and Maret, P. (2017). Question answering benchmarks for Wikidata. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference*, Vienna, Austria.

Duan, N., Tang, D., Chen, P., and Zhou, M. (2017). Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (2013). Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 690–696, Sofia, Bulgaria.

Heilman, M. and Smith, N. A. (2010). Good question! Statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). CTRL: a conditional transformer language model for controllable generation. *arXiv cs.CL*, 1909.05858.

Khullar, P., Rachna, K., Hase, M., and Shrivastava, M. (2018). Automatic question generation using relative pronouns and adverbs. In *Proceedings of ACL 2018, Student Research Workshop*, pages 153–158, Melbourne, Australia. Association for Computational Linguistics.

Malyshev, S., Krötzsch, M., González, L., Gonsior, J., and Bielefeldt, A. (2018). Getting the most out of Wikidata: Semantic technology usage in Wikipedia's knowledge graph. In *Proceedings of the 17th International Semantic Web Conference (ISWC 2018), LNCS volume 11137*, pages 376–394, Monterey, CA, USA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Pham, X. L., Pham, T., Nguyen, Q. M., Nguyen, T. H., and Cao, T. T. H. (2018). Chatbot as an intelligent personal assistant for mobile language learning. In *Proceedings of the 2018 2nd International Conference on Education and E-Learning (ICEEL 2018)*, page 16–21, Bali, Indonesia. Association for Computing Machinery.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Serban, I. V., García-Durán, A., Gulcehre, C., Ahn, S., Chandar, S., Courville, A., and Bengio, Y. (2016). Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 588–598, Berlin, Germany. Association for Computational Linguistics.

Su, M.-H., Wu, C.-H., and Chang, Y. (2019). Follow-up question generation using neural tensor network-based domain ontology population in an interview coaching system. In *Proceedings of INTERSPEECH*, pages 4185–4189, Graz, Austria.

Sun, X., Liu, J., Lyu, Y., He, W., Ma, Y., and Wang, S. (2018). Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939, Brussels, Belgium. Association for Computational Linguistics.

Unger, C., Ngomo, A.-C. N., and Cabrio, E. (2016). Sixth open challenge on question answering over linked data (QALD-6). In *Semantic Web Challenges, CCIS 641*, pages 171–177, Berlin. Springer-Verlag.

Zhao, Y., Ni, X., Ding, Y., and Ke, Q. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Zhou, W., Zhang, M., and Wu, Y. (2019). Question-type driven question generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6032–6037, Hong Kong, China. Association for Computational Linguistics.