Language Resources and Evaluation
https://doi.org/10.1007/s10579-023-09702-y

AC-IQuAD: Automatically Constructed Indonesian Question Answering Dataset by Leveraging Wikidata

Kerenza Doxolodeo[1] · Adila Alfa Krisnadhi[1]

Accepted: 27 October 2023

Abstract

Constructing a question-answering dataset can be prohibitively expensive, making it difficult for researchers to make one for an under-resourced language, such as Indonesian. We create a novel Indonesian Question Answering dataset that is produced automatically end-to-end. The process uses Context Free Grammar, the Wikipedia Indonesian Corpus, and the concept of the proxy model. The dataset consists of 134 thousand simple questions and 60 thousand complex questions. It achieved competitive grammatical and model accuracy compared to the translated dataset but suffers from some issues due to resource constraints.

Keywords Question answering dataset · Under-resourced Language · Automatic dataset construction

1 Introduction

Question answering (QA) is a natural language processing (NLP) task where one is given a question in a particular natural language, such as English. It must return a correct answer based on some textual reference corpus. A high-quality QA dataset is needed to train a model that solves this task. Each element of such a dataset consists of a natural language (NL) question, a piece of text, and location information in the text where an answer to the question can be found.

Traditionally, constructing a QA dataset requires access to human annotators to perform tasks such as writing the candidate questions and sourcing the context text. For instance, constructing the English SQuAD dataset leveraged Stanford's Daemo crowdsourcing platform, paying each annotator $10.50 per hour (Rajpurkar et al., 2018). However, this cannot be attainable for researchers who study under-resourced

✉ Adila Alfa Krisnadhi
adila@cs.ui.ac.id

Kerenza Doxolodeo
kerenza.doxolodeo@ui.ac.id

[1] Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia

Published online: 03 January 2024

languages for various reasons, such as the lack of a crowdsourcing platform. The Amazon Mechanical Turk platform is unavailable to continental Africa and most Asian countries, including Indonesia (Turk, 2017). Funding issues can also cause problems. This issue causes even the most extensive Indonesian dataset, TydiQA, to have a limited dataset size of six thousand (Clark et al., 2020).

Previous researchers have tried to create datasets without human help. For instance, translation algorithms have been used to translate the English QA dataset to their native language. The alignment algorithm determines which English word is translated into each word in the new language. By having the words lined up, we can trace and work out which substring in the new language is the substring of the correct answer (Carrino et al., 2020). However, this strategy can be inaccessible for exotic languages that do not have a well-trained translation model due to a lack of translation dataset between two languages or the reliance on Translation API such as Google Translate, which can be expensive.

Lewis et al. (2019) used a different strategy where they used a web dump. They trained an unsupervised translation algorithm where one turns a context paragraph into a question. Their idea is that if a translation algor is supposed to convert a text from language A to language B, one can pretend that the context sentences are written in language A. The questions are written in language B. While this scheme eliminates the need for annotators, our attempt to reproduce the paper suggests that more effort is needed due to the scarcity of corpus. For instance, the source code from Lewis et al. (2019) indicates that it requires twenty million example questions. However, Wilie et al. (2020), the most prominent web dump for the Indonesian language, contains only 2 million unique questions. This is problematic when one considers that Indonesian is the tenth most spoken language in the world, spoken by 200 million speakers (Ghosh, 2020). If one of the top 10 most-spoken languages has a dataset size issue, this begs concern for other less widely used languages.

Furthermore, one needs access to the correct answer to construct a QA dataset. This causes a chicken-and-egg problem where one needs to read the context text to get the truth. However, to train a model, one needs to know the truth. One can use a publicly available knowledge graph (KG) to circumvent this issue.

A KG contains set of facts like “Paris is the capital of France.” Typically, such a fact is represented as a triple, say (Paris, capital of, France). These facts can be used to automatically build an QA dataset by systematically generating NL questions about those facts. Some publicly accessible KGs can be used for this purpose, for example, DBPedia, which leveraged info box from Wikipedia articles (Lehmann et al., 2015), and Wikidata, whose data are crowd-sourced (Vrandecic & Krötzsch, 2014).

We follow the aforementioned idea to solve the challenge of creating an QA dataset for an under-resourced language. Specifically, we propose a novel Indonesian QA dataset created by leveraging Wikidata as a source of facts to generate potential questions with the help of a set of grammar rules that conform to Indonesian grammatical patterns. Note that Wikidata is a multilingual KG that stores the proper nouns of each entity in many languages, including Indonesian. Since Wikidata data items are typically connected to a corresponding Wikipedia page, we use the Indonesian Wikipedia Corpus to attach suitable context sentences to the candidate questions. A proxy model verifies these pairs as grammatically accurate and attached to the correct text. Finally, our approach ensures dataset diversity by conducting deduplication as a post-processing step.

The generated dataset is called AC-IQuAD. Each row of AC-IQuAD consists of an question, a context paragraph, and the location of the substring that contains the correct answer to the question. Both the question and the context text are in Indonesian. Furthermore, as answers are obtained from Wikidata facts using SPARQL queries, each Indonesian question in the dataset has an equivalent SPARQL query. This allows AC-IQuAD to have a secondary purpose as a dataset for the knowledge graph question answering (KGQA) task, where the model converts a question in natural language to a SPARQL query which can be run against the KG to obtain the answer. However, the evaluation of the dataset presented in this paper is focused only on the natural language QA task. The evaluation concerning the KGQA task is left as future work.

The evaluation of our dataset comprises both manual and automated evaluation. We argue that it is essential that a dataset is evaluated manually by native speakers, but having it evaluated automatically can address the scalability issue. We present our human annotator with a sample of 100 entries from six types of questions and have them either approve the question or disapprove it with a pre-determined explanation. As for the automated evaluation, we fine-tune M-BERT with AC-IQuad and other benchmark QA datasets and compare their accuracy. A good dataset will have a high approval rate from human annotators and result in state-of-the-art models to achieve competitive accuracy compared to other datasets.

The paper is organized as follows. Section 2 discusses the related work relevant to our study. Section 4 outlines the four steps to generate the dataset. We then discuss the evaluation method in more detail with Sect. 5. Section 6 covers the evaluation results, and finally, Sect. 7 concludes.

2 Related work

2.1 Natural language question answering datasets

The largest English QA dataset is SQuAD 2.0 (Rajpurkar et al., 2018), comprising 150 thousand QA items. One-third of these items are *impossible items* where the context does not provide any substring that can be the correct answer. These impossible items are necessary to ensure that models do not overfit simple semantic text patterns but can demonstrate that they understand the text in a deeper meaning and show robustness against distracting sentences (Weissenborn et al., 2017).

There are two Indonesian QA datasets. The most prominent native is TyDI QA (Clark et al., 2020), which covers ten languages, including Indonesian. Based on our count, it has 6 thousand entries for training and 2 thousand entries for development & testing. The dataset is constructed by providing the first group of annotators snippets to a Wikipedia article and coming up with a genuine question unanswered

Question: Which city stands on the Vistula River?

Query:

```
SELECT ?item
WHERE
{
   ?item wdt:P206 wd:Q546.
}
```

Fig. 1 An example of an entry of a KGQA dataset where the SPARQL query conveys the intent question: an answer to the query coincides with an answer to the question. The namespace prefix `wdt:` and `wd:` refer to the URI http://www.wikidata.org/prop/direct/ and http://www.wikidata.org/entity/, respectively

by the article. The second group of annotators searched for a relevant text that responded to the question.

There is also an Indonesian SQuAD dataset (Muis & Purwarianti, 2020) obtained by translating the English SQuAD to Indonesian using Google Translate API. They employed the token alignment algorithm from Carrino et al. (2020) to track which token is translated to which token. This allowed them to work out where is the location of the correct substring in the context text. However, Clark et al. (2020) noted that this is not ideal as the generated text is too “translationese”[1] and not native enough.

2.2 Knowledge graph question answering datasets

The KGQA task refers to answering a given question with an answer obtained from entities in some knowledge graph. To solve this task, one usually has to construct a KG query, e.g., SPARQL, that captures the intention of the question as demonstrated by Fig. 1. The most comprehensive English KGQA dataset is LC-QUAD 2.0 (Dubey et al., 2019). It is notable for the variety of its question styles. Besides the straightforward one, it offered True-False questions, such as “Is Juan José Ibarretxe a chairperson of FC Barcelona?” that requires a SPARQL ASK-query, transitive questions (“The movie Hellboy is produced by which man who directed Shape of Water?”), questions that require access to more than one triple to answer (“Who are the writers of The Second Coming, whose death place is Menton?”). These questions are constructed by manually taking certain entities and relations that have been pre-determined. The construction proceeds through the KG to find as many triples as possible matching the pre-determined entity and relation list. These triples are then converted to text using several templates. These templates are not designed to be grammatically accurate. Merely, they exist to be used to create draft sentence,

[1] This is the actual term used by Clark et al. (2020).

which then can be rewritten by human annotators. Such a dataset for Indonesian has yet to exist.

2.3 Automated QA dataset construction

To the authors' best knowledge, there has been no attempt to build a QA dataset automatically in Indonesian. The following papers are efforts in the English language. We consider two approaches for automated question generation: deriving one from a knowledge graph triple or deriving one from a context text.

To the authors' best knowledge, there has yet to be an attempt to build an QA dataset automatically in Indonesian. The following papers are efforts in the English language. We consider two approaches for automated question generation: from a knowledge graph triple or a context text.

Serban et al. (2016) scraped Yahoo Answer to seek out the typical pattern of questions. They run their program to find a typical n-gram sequence in the corpus. The proper noun that appears in the question will change. Therefore, every similar n-gram sequence is grouped into one template. However, the proper noun is blanked with $. For instance, one template is "Who is the wife of #." To form a question from the context, a model trained with a dataset from Yahoo Answer accepts a context text and finds which template is the most suitable.

Lewis et al. (2019) leveraged Lample et al. (2018)'s unsupervised translation algorithm. They realized that instead of using the algorithm to translate English to another language, they could translate a context text to a question. The relevant noun is clozed and replaced with the proper NER tag to guide the algorithm in answering the desired question. For instance, for the context "[PLACE] is the capital city of Poland, whose currency is Zloty.", then the expected question is "What is the capital of Poland?" and not "What is the currency of Poland?" because it is Warsaw that is being clozed.

Heilman and Smith (2010) performed an exhaustive analysis and came up with heuristics to manipulate the grammar of a context sentence into a question sentence. They achieved this by representing the sentence as a constituency tree. Tregex rules are applied to the grammar of the sentence with Tsurgeon.

Like Heilman and Smith, we choose this explicit grammatical manipulation strategy. More precisely, we employ context-free grammar to exploit the grammatical patterns of the Indonesian language. This is necessary as we do not have a large corpus, so having something that can operate with minimal input helps.

2.4 Performance evaluation via proxy models

When one aims for a golden standard dataset, the standard practice is to evaluate the dataset's fitness to the problem with the help of human annotators. In the case of a QA problem, the dataset may contain examples, each of which is a question-answer pair. To evaluate the dataset, the annotators manually check every single example in

Table 1 Complex question definition

Type	Description	Example triples
?shr	Querying for the subject of 2 triples with the same predicate	?a wdt:P50 wd:Q208460. ?a wdt:P50 wd:Q1396889
?unq	Querying for the subject of 2 triples with different predicate	?a wdt:P50 wd:Q208460. ?a wdt:P108 wd:Q9531
shr?	Querying for the object of 2 triples with the same predicate	wd:Q26698156 wdt:P57 ?a. wd:Q461540 wdt:P57 ?a
unq?	Querying the object of 2 triples with different predicate	wd:Q26698156 wdt:P57 ?a. wd:Q461540 wdt:P162 ?a

the dataset and must ensure that the answer part of the example is indeed an answer to the question part.

However, this technique is tricky to scale due to the time and resources needed. A possible alternative is to automate it with a proxy model. In this case, we first pick a model that is known to perform well on the QA task based on past evaluation of some benchmark datasets. We then evaluate it to our newly created dataset. Suppose the latter evaluation achieves at least a comparable level of performance on the benchmark dataset. In that case, our newly created dataset is at least as good as the benchmark dataset.

Eyal et al. (2019) already explored this possibility with a text summarization problem. They suggested that a good text summarization model should retain all the necessary information. From this observation, a QA model is trained as a proxy model. The proxy model is then quizzed on the summarized text. A high accuracy indicated that the dataset was grammatically sound and did not lose the necessary info; otherwise, the model would be "confused" or not have the necessary substring to answer the question.

3 AC-IQuAD: dataset overview

Here and henceforth, we use the following IRI namespace prefixes:

- `wd:` for http://www.wikidata.org/entity/
- `wdt:` for http://www.wikidata.org/prop/direct/.

Our dataset AC-IQuAD contains two types of question: simple and complex question. A simple question expresses a SPARQL query consisting of one triple pattern, possibly, with an additional triple pattern defining the entity type of the queried subject or object. For example, **"Apa diproduseri oleh Guillermo Del Toro?"** or "What is produced by Guillermo Del Toro?" is a simple question since it can be expressed by a single triple pattern: `?A wdt:producer wd:Guillermo Del Toro`. Here, `?A` is a variable as indicated by a question mark prefix.

Question: **“Finlandia dan swedia bahasa resmi kota apa?”**

Query:

```
SELECT ?ax
WHERE
{
   ?ax wdt:P37 wd:Q1412.
   ?ax wdt:P37 wd:Q9027.
   ?ax wdt:P31 wd:Q515.
}
```

Answer: {r: ’Q1757’, n: [’Helsinki’]}

Context: **“Angka harapan hidup adalah 75,1 tahun untuk laki-laki dan 81,7 tahun untuk perempuan.BahasaBahasa Finlandia dan Bahasa Swedia adalah bahasa resmi munisipalitas Helsinki.”**

Answerline: [{answer: ’Helsinki’, start: 160, end: 168}]

Type: ?shr

Fig. 2 An example of a complex question entry from AC-IQuAD. The question (given in Indonesian) asks which city does have Finnish and Swedish as its official languages. The query part gives a SPARQL query formed by triple patterns that represent the question. The answer field denotes the entity that is an answer to the question, with the key ’r’ denoting its Wikidata entity ID. The context field provides the context text in which the answer of the question appears. Like in the SQuaD dataset, the field ‘answerline’ stores the substring that is the correct answer, with ‘start’ and ‘end’ indicating the location of the answer substring within the context text

Table 2 The WH-word breakdown of simple question

WH-type	Count	Example
Di mana—Where	10,289	**Di mana stadium Truist Park?** Where is Truist Park Stadium?
Siapa—Who	7545	**Siapa sutradara Shape of Water?** Who directed Shape of Water?
{TYPE} apa—{TYPE} what	92,318	**Di kota apakah stadium Truist Park?** In what city is Truist Park Stadium?
Apa—What	34,194	**Apa disutradarai oleh Guillermo Del Toro?** What was directed by Guillermo Del Toro?

Complex questions correspond to a SPARQL query that consists of two triples plus an optional typing triple of the queried subject or object entity. Generally, we divide complex questions into four types as shown in Table 1.

Table 3 The top 10 most widely used specifier for simple what-questions

Types	Frequency
Kecamatan—district	21,249
desa (Q532)—village	17,999
desa/kampung/kelurahan (Q2225692)—sub-district	15,781
film—movie	6,550
kelurahan (Q965568)—village	4669
komune di Italia Italian commune	3675
desa (Q26211545) village	1424
Munisipalitas di Filipina (Q24764) Phillipines municiplity	1321
Kota City	1029
Negara Country	1019

Table 4 The breakdown of complex question types by their fold

Type	Training examples	Test examples
?shr	3727	2106
?unq	1268	394
shr?	46,943	3423
unq?	2143	383

The dataset contains 134,645 simple questions and 60,387 complex questions, as shown by Fig. 2. Table 2 shows that the majority of the simple questions is a what-question with a type specifier, e.g., "**film apa ...**" ("what movie ...") instead of "**apa ...**" ("What ..."). The dataset has a total of 1002 unique specifiers for the what-questions and the 10 most widely used are listed in Table 3. The majority of these type of specifiers is about Indonesian districts, sub-districts, and villages.[2] Upon further investigation, we found that this happens because Indonesian Wikipedia has a low-content article for every district and sub-district in Indonesia, most likely created by a bot. Our procedure picked up these articles to populate the AC-IQuAD and skewed the topic distribution.

As Table 4 shows, the majority of the complex questions are of type shr?. This phenomenon is an artifact of our pipeline design where merging knowledge from two different articles often yields a number of sentences with the same subject, hence it tends to generate shr? question. Section 5.2 discussed why the train test ratio is not uniform by question type.

Figure 3 shows the cumulative distribution of the relative location of the answer span, with 0 as the context text's first character and 1 as the last character of the context text. Most of the answer line resides in the first couple of

[2] Both **kelurahan** (sub-district) and **desa** (village) refer to the fourth level of Indonesia's administrative division.

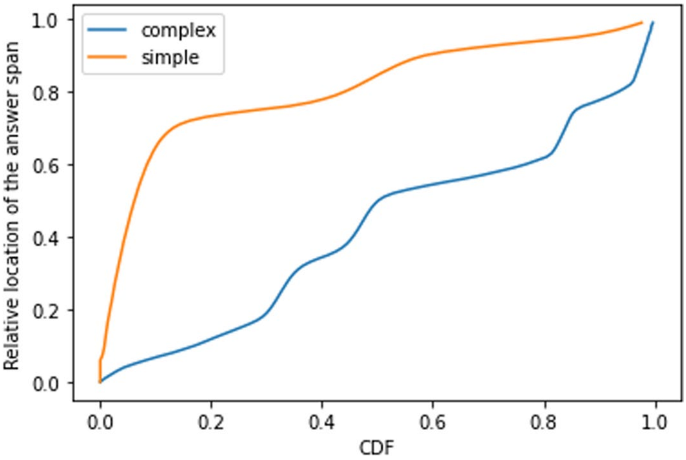

Fig. 3 Cumulative distribution of the relative location of the answer span. The x-axis shows the relative location of the answer span inside the context text

sentences for the simple questions. Among all entries, 20% of the answer span is within the first percent of the text. This makes much sense as a good portion of the context is taken from the first couple of introductory sentences from Wikipedia articles, and they put the essential proper nouns as early as possible. However, the answer line location is more spread around for the complex questions. This happens because complex questions are created by merging two simple question entries. This happens because complex questions are created by merging two simple question entries. Since each entry has its context text, the concatenated context text contains answer spans that are more spread around.

4 Method

As outlined by Fig. 4, with some variation in the detail, the method to create both simple and complex questions consists of the following four steps.

1. *Candidate Question Generation* We convert a triple from Wikidata into a simple question by using grammar. For complex questions, we convert two triples instead. Heuristics is used to decide the proper WH word for the question. In both simple and complex question cases, the type instance of the queried subject or object is considered when deciding the proper WH-word.
2. *Discovery of relevant context sentence* We extract a context text for the question from the subject entity's Wikipedia Indonesia page. If we do not find one, the question is dropped. This step also doubles as a way to reject grammatically unsound questions.

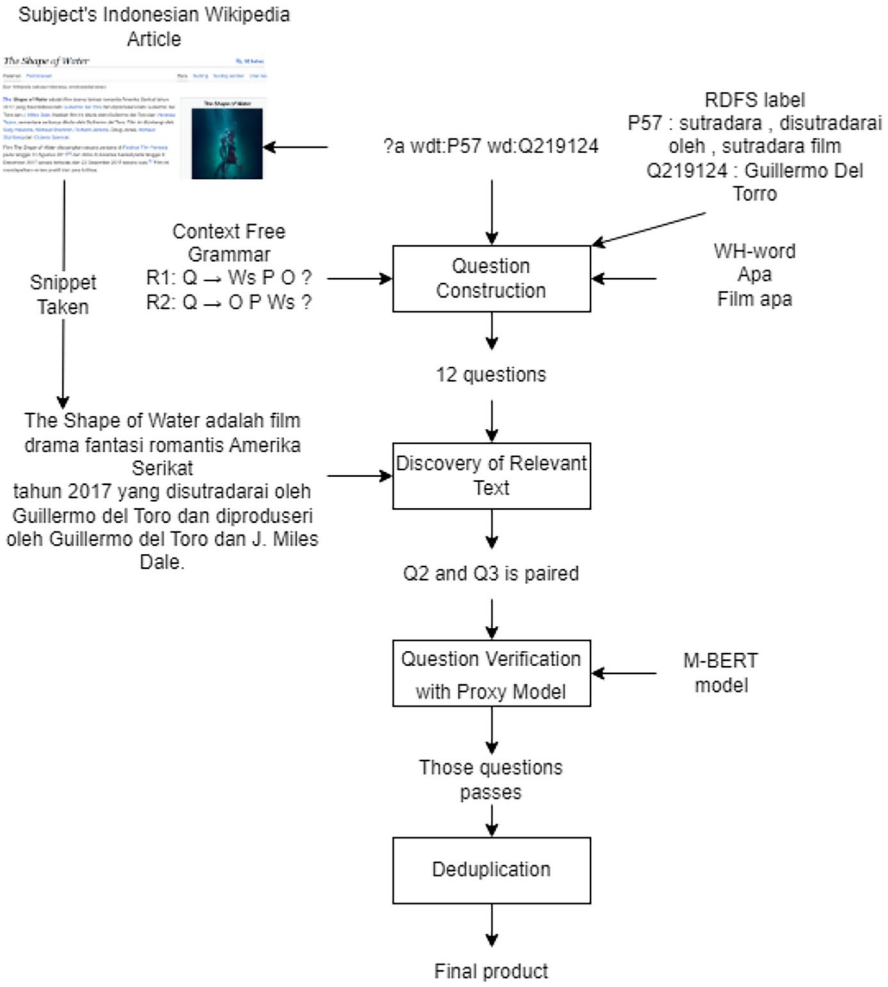
Subject's Indonesian Wikipedia Article
?a wdt:P57 wd:Q219124
RDFS label
P57 : sutradara , disutradarai oleh , sutradara film
Q219124 : Guillermo Del Torro
Context Free Grammar
R1: Q → Ws P O ?
R2: Q → O P Ws ?
Snippet Taken
Question Construction
WH-word
Apa
Film apa
12 questions
The Shape of Water adalah film drama fantasi romantis Amerika Serikat
tahun 2017 yang disutradarai oleh Guillermo del Toro dan diproduseri oleh Guillermo del Toro dan J. Miles Dale.
Discovery of Relevant Text
Q2 and Q3 is paired
Question Verification with Proxy Model
M-BERT model
Those questions passes
Deduplication
Final product

Fig. 4 Workflow diagram with an example where we process

3. *Question verification with proxy model* We use a trained QA model to verify that the questions are proper by having it answer them. If the proxy model comes up with a different answer, this gives us a reasonable cause that the question is bad and we should discard it.
4. *Deduplication* As this process may create more than one question, deduplication is done to reduce potential noise from bad questions that still pass through the previous step by keeping only the best ones.

To simplify the discussion, we describe the workflow only for the simple questions. Section 4.5 explains how the steps are modified for the complex question case.

4.1 Question construction via grammar

This step takes a triple and converts it into a question. This step aims to create as many valid questions as possible. Some of the questions generated can be dropped in subsequent steps if it is grammatically incorrect. This step consists of two parts.

1. Selection of an appropriate WH-word for the input triple.
2. Conversion of the triple into a natural language question with the selected WH-word.

4.1.1 WH-word selection

Let $t =$ `(S P O)` be the input triple. From this triple, one could consider two query patterns `(?x P O)` and `(S P ?y)`, asking for the subject and object entity, respectively. Suppose we focuse on the former, i.e., the asked entity is the subject of *t*. Then, we select an appropriate WH-word as follows:

1. If the triple `(S wdt:P31 wd:Q5)` holds in Wikidata, then `?x` represents an instance of human, and thus, the WH-word **"siapa"** or "who" is selected.
2. If the query `(S wdt:P625 ?coord)` has an answer for `?coord` in Wikidata, then `?x` is a geospatial place because the property `wdt:P625` represents coordinate location. In this case, **"di mana"** or "where" is a WH-word. In addition, if the query `S wdt:P31 ?Type` has an answer for `?Type`, then we have typed what-question. For example, if `?Type` is `wd:Q515`, which corresponds to 'city', then WH-phrase is **"Kota apa"** or "What city".
3. If neither `(S wdt:P31 wd:Q5)` nor `(S wdt:P625 ?coord)` is satisfied in Wikidata, then **"apa"** or "what" is an appropriate WH-word, leading to an untyped what-question. In addition, a typed what question word is also generated like above.

The above rules are also applied to *t*, but with the focus on `O`, the object of *t* as the asked entity. Note that the above rules allow one triple to result in the selection of multiple WH-words. As a more concrete example, consider the following set of triples:

$$\mathtt{wd}:\mathtt{Q26698156wdt}:\mathtt{P57wd}:\mathtt{Q219124}.$$

$$\mathtt{wd}:\mathtt{Q26698156wdt}:\mathtt{P31wd}:\mathtt{Q26698156}.$$

which expresses the fact that the movie Shape of Water (`wd:Q26698156`) has a director named Guillermo del Torro (`wd:Q219124`). Here, the first triple generates questions, while the second triple is an instance-of triple used to provide additional context. From these triples, we choose all the of following WH-words/phrases:

- **"siapa"** or "who" if the question is asking about `wd:Q219124`.

- **"film apa"** or "what movie" and **"apa"** or **"what"** if the question is asking about `wd:Q26698156`. Shape of Water. One should note that if `wd:Q26698156` has more than one Indonesian label, all possible alternatives become an acceptable specifier.

4.1.2 Candidate question generation

For this step, let $t =$ `(S P O)` be the triple being considered. Also, suppose `Ws` represents the appropriate WH-word/phrase for `S` and `Wo` represents WH-word/phrase for `O`. Then, we generate candidate questions Q for t according to the following four grammar rules where the first two are used when `S` is being asked, while the remaining two are used when `O` is being asked instead:

R1: `Q` → `Ws P O ?`
R2: `Q` → `O P Ws ?`
R3: `Q` → `S P Wo ?`
R4: `Q` → `Wo P S ?`

The concrete candidate questions generated from the above grammar are obtained by replacing `S`, `P`, and `O` by the corresponding human-readable labels in Wikidata.

For instance, we have the triple (`wd:Q26698156 wdt:P57 wd:Q219124`) and we are querying the subject. According to Wikidata, we know of the following Indonesian labels.

1. The label for `wd:Q219124` is "Guillermo Del Toro"
2. The label for `wdt:P57` is (a) **"sutradara"** or "director", (b) **"disutradarai oleh"** or "directed by", and (c) **"sutradara film"** or "movie director".
3. We also know that the acceptable WH-word for `wd:Q26698156`, i.e., "Shape of Water", is (a) **"film apa"** or "what movie" and (b) **"apa"** or "what"

From this observation, we have 12 possible combinations.

Q1. (Rule R1, 2a, 3a) **Film apa sutradara Guillermo Del Toro?**
Q2. (Rule R1, 2a, 3b) **Apa sutradara Guillermo Del Toro?**
Q3. (Rule R1, 2b, 3a) **Film apa disutradarai oleh Guillermo Del Toro?**
Q4. (Rule R1, 2b, 3b) **Apa disutradarai oleh Guillermo Del Toro?**
Q5. (Rule R1, 2c, 3a) **Film apa sutradara film Guillermo Del Toro?**
Q6. (Rule R1, 2c, 3b) **Apa sutradara film Guillermo Del Toro?**
Q7. (Rule R2, 2a, 3a) **Guillermo Del Toro sutradara film apa?**
Q8. (Rule R2, 2a, 3b) **Guillermo Del Toro sutradara apa?**
Q9. (Rule R2, 2b, 3a) **Guillermo Del Toro disutradarai oleh film apa?**
Q10. (Rule R2, 2b, 3b) **Guillermo Del Toro disutradarai oleh apa?**
Q11. (Rule R2, 2c, 3a) **Guillermo Del Toro sutradara film film apa?**
Q12. (Rule R2, 2c, 3b) **Guillermo Del Toro sutradara film apa?**

Q2 “What is directed by Guillero Del Toro?”, Q3 “What movie is directed by Guillermo Del Toro?”, Q4 “What is directed by Guillermo Del Toro?, Q8 “Guillermo Del Toro directs what?” as well as Q7 and Q12 “Guillermo Del Toro directs which movie?” are sound semantically and grammatically. All twelve questions goes to the next step.

It is possible for two combinations to come up with the same sentence. For example, Q7 and Q12 are identical, despite the fact that they arrive from a different rule application. Q7 has “sutradara film” as `P` and “apa” as `Wo`. Meanwhile, Q12 has “sutradara” as `P` and “film apa” as `Wo`.

Note that this example is merely for querying for the subject. To write a question for the object, we have to use a similar process using rule 3 and rule 4 instead.

4.2 Discovery of relevant context sentence

We search the relevant sentence for the question from Indonesian Wikipedia articles. We achieve this by going to the triple’s subject, even if the triple is querying the object. From our empirical experience, going through the object’s article provides a higher hit rate, even if we are querying the object.

For instance, if the triple is `(wd:Q26698156 wdt:P57 wd:Q21912)`, then we go to the Wikipedia article of Shape of Water (i.e., the entity `wd:Q26698156`), regardless of whether the question is about the movie or the director. The sentences from the article are tokenized at a token level and searched for the ideal context sentence.

A sentence is considered an *ideal context sentence* if:

- it contains (an exact match) of the subject, predicate, and object of the question;
- the order of the subject, predicate, and object in it is consistent with the question.

As an example, consider the following actual snippet from the Indonesian Wikipedia article of Shape of Water:

> **The Shape of Water adalah film drama fantasi romantis Amerika Serikat tahun 2017 yang disutradarai oleh Guillermo del Toro dan diproduseri oleh Guillermo del Toro dan J. Miles Dale.**
>
> The Shape of Water is a 2017 American fantasy romantic drama directed by Guillermo de Toro and produced by Guillermo del Toro and J Miles Dale.

For this particular snippet, Q2 and Q3 from Sect. 4.1.2 are the consistent questions.

- It uses both “Guillermo del Toro” and the predicate “disutradarai oleh”, instead of the alternative “sutradara” and “sutradara film”.
- The questions are consistent with their text’s subject-object order: “disutradarai oleh” comes before “Guillermo del Toro”.

In this step, we look for an ideal context sentence from the chosen Wikipedia article for each candidate question generated in the previous step. Hence, different candidate questions may be associated with different text snippets. Only questions to which we successfully attach a snippet go to the next step; the remaining are discarded. In addition, more than one text snippet may be attached to a single question. In that case, it will result in separate entries in the dataset.

One may notice that grammatically invalid questions such as Question 1 “What movie was produced by Guillermo Del Toro?’ that suffered from subject-object order confusion, would not obtain an associated text snippet pass to this question. This is because, unless the author of the Wikipedia article also committed the same mistake, the confused question will never pass the second condition of an ideal context sentence.

4.3 Question verification with proxy model

This step aims to verify that the questions generated are grammatically and semantically good. While we are confident that our grammar rules cover a good proportion of scenarios, Wikidata entity and property labels used in the second step may not be suitable. In this step, a proxy model takes a question and its context produced by the previous step. A question passes the verification if the proxy model returns an exact match as the intended answer (i.e., the label of the asked entity) with at least 70% confidence. The entire entry is discarded if the returned answer is not an exact match or associated with a smaller confidence.

The proxy model we use is an M-BERT fine-tuned with the English dataset SQuAD and the Indonesian part of TydiQA. M-BERT is a multilingual BERT trained with Wikipedia corpus across dozens of languages, including Indonesian (Devlin et al., 2019). The model worked because languages usually share the same word for the same meaning. Thus, it can infer the meaning of a word that exists only in one language by seeing how this isolated word interacts with shared words. Since it is a multilingual model, it can perform in a bilingual dataset, including training in one language and predicting in another language (Siblini et al., 2019). However, we decided that for this study, they will be combined. Our intuition is that the big size of SQuAD provides the volume needed to ensure the model fits, while the native TydiQA provides the refinement.

4.4 Deduplication

More than one question may be associated with the same snippet. In this case, we only keep the questions with a good chance of being grammatically and semantically valid. We use the following rule:

- Among typed what-questions with the same context text, keep the one with the highest probability by the proxy model.
- Among untyped what-questions with the same context text, keep the one with the highest probability by the proxy model.

Table 5 Grammar rules for complex question type ?shr. Input triples and their labels are given below. The token `dan` is the Indonesian word for ‘and’

`wd:Q266981 wdt:P50 wd:Q208460 .`	(George Orwell, author, 1984)
`wd:Q266981 wdt:P50 wd:Q1396889 .`	(George Orwell, author, Animal Farm)

?shr rule	?shr example
`Q → Ws P O1 dan O2 ?`	**Siapa menulis 1984 dan Animal Farm?** Who wrote 1984 and Animal Farm?
`Q → O1 dan O2 P Ws ?`	**1984 dan Animal Farm ditulis oleh siapa?** 1984 and Animal Farm were written by whom?

Table 6 Grammar rules for complex question type shr?. Input triples and their labels are given below. The token `dan` is the Indonesian word for ‘and’

`wd:Q26698156 wdt:P57 wd:Q219124 .`	(Shape of Water, director, Guillermo Del Toro)
`wd:Q461540 wdt:P57 wd:Q219124 .`	(Hellboy, author, Guillermo Del Toro)

shr? Rule	shr? example
`Q → Wo P S1 dan S2 ?`	**Siapa sutradara Hellboy dan Shape of Water?** Who directed Hellboy and Shape of Water?
`Q → S1 dan S2 P Wo?`	**Hellboy and Shape of Water disutradarai oleh siapa?** Hellboy and Shape of Water were directed by whom?

Table 7 Grammar rules for question type ?unq. Input triples and their labels are given below. The token `dan` is the Indonesian word for ‘and’, while the phrase `yang juga` roughly translates to ‘that is also’

`wd:Q266981 wdt:P50 wd:Q208460 .`	(George Orwell, author, 1984)
`wd:Q266981 wdt:P108 wd:9531 .`	(George Orwell, employer, BBC)

?unq Rule	?unq example
`Q → Ws P1 O2 dan P2 O2 ?`	**Siapa penulis 1984 dan bekerja di BBC?** Who wrote 1984 and worked in BBC?
`Q → Ws P2 O2 dan P1 O1 ?`	**Siapa bekerja di BBC dan menulis 1984?** Who worked in BBC and wrote 1984?
`Q → P1 O1 Ws yang juga P2 O2?`	**1984 ditulis oleh siapa yang juga kerja di BBC** 1984 was written by whom that also worked in BBC?
`Q → P2 O2 Ws yang juga P1 O1?`	**BBC lokasi kerja siapa yang juga penulis 1984?** BBC was the workplace of whom that also wrote 1984?

Assuming both questions Q2 and Q3 survive the previous step, they are not deduplicated as the former is an untyped what-question, while the latter is a typed what-question.

Table 8 Grammar rules for unq?. Input triples and their labels are given below. The token `dan` is the Indonesian word for 'and', while the phrase `yang juga` roughly translates to 'that is also'

`wd:Q26698156 wdt:P57 wd:Q219124 .`	(Shape of Water, director, Guillermo Del Toro)
`wd:Q461540 wdt:P58 wd:Q219124 .`	(Hellboy, scriptwriter, Guillermo Del Toro)

unq? rule	?unq example
`Q → Wo P1 S1 dan P2 S2 ?`	**Siapa sutradara Shape of Water dan penulis skrip Hellboy** Who was the director of Shape of Water and the scriptwriter of Hellboy?
`Q → Wo P2 S2 dan P1 S1 ?`	**Siapa penulis skrip Hellboy dan sutradara Shape of Water?** Who was the scriptwriter of Hellboy and the director of Shape of Water?
`Q → P1 S1 Wo yang juga P2 S2 ?`	**Shape of Water disutradarai oleh siapa yang juga penulis skrip Hellboy?** Shape of Water was directed by whom that also wrote the script of Hellboy ?
`Q → P2 S2 Wo yang juga P1 S1?`	**Hellboy ditulis oleh siapa yang juga disutradarai oleh Shape of Water** Hellboy was writen by whom that also directed Shape of Writer?

Table 9 Two different entries

	Entry 1	Entry 2
Triple	`wd:Q26698156 wdt: P57 ?a`	`?a wdt:P162 wd:Q2191`
Question	**Siapa sutradara film Hellboy?** Who directed Hellboy?	**Guillermo del Toro produser film apa?** Guillermo del Toro produced what movie?
Context	**Hellboy adalah film aksi supranatural yang dibuat tahun 2004 dibintangi Ron Perlman, John Hurt dan Selma Blair, serta disutradarai Guillermo del Toro.** Hellboy is a supernatural action movie made 2004 starred by Ron Perlman, Jon Hunt, and Selma Blair, and directed by Guillermo del Toro	**The Shape of Water adalah film drama fantasi romantis Amerika Serikat tahun 2017 yang disutradarai oleh Guillermo del Toro dan diproduseri oleh Guillermo del Toro dan J. Miles Dale.** The shape of Water is a 2017 American fantasy rom-com drama directed Guillermo del Toro and produced by Guillermo del Toro and J. Miles Dal
Answer	`wd:Q219124` (Guillermo Del Toro)	`wd:Q461540` (Shape of Water)

4.5 Refinement for the complex question

Complex questions describe two triples that share the same subject or object. We update the question generation and text discovery steps to generate these questions as they require more detailed grammar and text discovery techniques.

Let t_1 = `(S1 P1 O1)` be the first triple being considered and t_2 = `(S2 P2 O2)` be the second one. The two triples either share the same subject or the same object, but not both. We use c and `Wo` to denote an appropriate WH-word/phrase for the shared subject and object, respectively. Then, we propose two grammar rules for ?shr and shr? types, as shown in Tables 5 and 6 respectively.

Table 10 Newly created complex entries

Label	Complex entry
Triples	`wd:Q26698156 wdt: P57 ?a. wd:Q461540 wdt:P162 ?a.`
Question	**Shape of Water disutradarai oleh siapa yang juga produser Hellboy?** Shape of Water is direcred by whom that also producered Hellboy?
Context	Hellboy adalah film aksi supranatural yang dibuat tahun 2004 dibintangi Ron Perlman, John Hurt dan Selma Blair, serta disutradarai *Guillermo del Toro*. The Shape of Water adalah film drama fantasi romantis Amerika Serikat tahun 2017 yang disutradarai oleh *Guillermo del Toro* dan diproduseri oleh *Guillermo del Toro* dan J. Miles Dale
Answer	`wd:Q219124` (Guillermo Del Torro)

Furthermore, we and four pairs of CFG rules for ?unq and unq?, as shown in Tables 7 and 8.

We do not go back to the Wikipedia corpus for the second step. We go through the triples and context created from the simple question dataset. If the triple overlaps in one of the four fashions described in the list, they are combined to make a complex question. If the pair has the same context test, we use it. However, if the pair has a different text, we can concat the text into one. Multiple appearances of the answer line from the concatenated text will be accepted as part of the solution.

Take the example of Table 9 which shows two different simple question entries. The triple `wd:Q26698156 wdt:P57 wd:Q219124` and `wd:Q461540 wdt:P162 wdt:219124` shares the same object but a different predicate. This makes a candidate for an unq? question. They are then combined as shown in Table 10.

5 Evaluation methods

We argue that to measure the fitness of a dataset, it needs to be evaluated by both human and a QA model. Human evaluators who are native speaker are naturally the ideal form of judges, but they cannot cover the massive size of the dataset. QA model evaluation where we see the accuracy with different training covers the volume issue.

5.1 Human evaluation

We hired three annotators to evaluate six groups of data. The first five groups are part of the AC-IQuAD dataset, which consists of a group of simple questions and four groups of each complex question type. We also have a sample of the English SQuAD dataset translated to Indonesian as a comparison.

For each group, each annotator annotates fifty entries. The first twenty-five questions are shared among annotators. This is to have a sample that we can measure for annotator agreement. However, the second half is different for each annotator. This means there are a total of one hundred entries. Once we can show a high annotator agreement from the sample of 25 × 6 = 150 shared questions, we can feel confident about the quality of the remaining (25 × 3) × 6 = 450 unique questions.

Each annotator is given the question, the answer, and the context text, and they must give it a label of either:

- *Correct.*
- *Flawed evidence.* The question is grammatically and semantically valid, but the context does not answer the question.
- *Problematic grammar.* A question is given this label if it is grammatically or semantically invalid. The annotators are given the following rule of thumb: if the question does sound right when it is read out loud, then it is not grammatically problematic. This guidance is provided to ensure that questions are flagged for a genuine problem and not for some minuscule concern such as slight imprecision word choice.
- *Ambiguous question.* The annotator could not work out what the question is without looking at the answer or the context text.
- *Invalid for other reasons.*

We are interested in the percentage of questions approved by annotators and given the correct label. A detailed explanation of the wrong answer to analyze any systemic identification.

5.2 Computer evaluation

For this evaluation, an M-BERT model is trained with different variations of the dataset. The predicted answer and the actual answer are compared using two metrics: Exact Match (EM) and F1. Exact Match only credits the model when it gives an identical substring. In case the question has more than one correct substring, to get the credit, the model only has to return one of them. F1 will give partial credit if the returned substring has an extra token or is missing some necessary tokens. If the model gets partial credit from multiple correct substrings, the model keeps the biggest partial credit.

For the simple questions, two test dataset are used: TydiQA's test fold and AC-IQuAD's simple question test fold. Six types of training dataset are tested:

- TydiQA's train fold
- AC-IQUAD's simple question training fold
- English SQuAD's train fold
- Translated Indonesian train fold
- TydiQA combined with the English SQuAD.
- AC-IQuAD's simple question training fold combine with the English SQuAD.

To divide the AC-IQuAD's simple questions into training and test folds, with attention given to avoid leaking knowledge in the training fold to the test fold due to too similar questions, the following steps are done:

Table 11 Kappa agreement between annotators

Annotator pair	Binary Kappa	Specific Kappa
1/2	0.53	0.66
1/3	0.45	0.56
2/3	0.55	0.56

1. From the set of all unique context text, we randomly sampled 50% of them.
2. Any entry whose context text is in the 50%-list goes into the training fold. Since some entries may share the same context text, the training fold would contain more entries than the test fold.
3. As one triple may generate two or more questions, this step aims to keep all of them either in the training fold or testing fold, but not both. We move any entry from the test fold to the training fold if it shares the same (non instance-of) triple with another entry that is already in the training fold.. For example, both "Washington DC is the capital of which country?" and "What is the capital of US?" are derived from (`wd:Q30 wdt:P36 wd:Q61`), i.e., (United States, capital, Washington DC). Therefore, if one of them is the training fold, the second one will get transferred to the training fold as well. This specific step is called the "absorption step".
4. Any remaining entry goes to the test fold.

For the complex question, the test dataset used is AC-IQuAD's test fold of the complex question. Three training datasets are tried:

- AC-IQuAD's simple questions,
- AC-IQuAD's combined simple question and complex question training fold, and
- the translated SQuAD dataset.

We point out that as each complex question "merges" two simple questions, there are three potential combinations:

- It merges two simple questions which are both in the training fold.
- It merges two simple questions, one from the training fold and the second from the test fold.
- It merges two simple questions which are both in the test fold.

We intend only questions from the third group from the AC-IQuAD complex question to go to the test fold, while the rest go to the training fold. To do this, we compare the complex questions triples, and an entry only goes to the complex question's test fold if neither triple appears in the simple question training fold.

Table 12 Percents of questions approved by annotator

Data type	Correct (%)
Translated question	67
Simple question	72
?shr	66
?unq	75
shr?	74
unq?	23

Table 13 Breakdown why questions are disaproved by the annotator

Issue	Translated	Simple	?shr	?unq	sbr?	unq?
Flawed evidence	5	2	3	3	0	2
Problematic grammar	10	12	27	23	23	62
Ambigious question	16	14	3	2	3	13
Other reasons	2	0	1	0	0	0
Total rows	33	28	34	25	26	77

6 Results

6.1 Human evaluation

Table 11 shows two kappas. The first kappa shows the annotator's agreement on whether the sentence is good or not. The second kappa shows the annotator's agreement on the specific reasoning. The interpretation of how good the kappas are context-dependent. For the medical purpose, given the human lives at stake, these are too low for one to make a decision from it. However, for us, this is a perfectly reasonable inter-annotator agreement (McHugh, 2012). Table 12 showed with the exception of the question querying the subject with a different predicate, our produced dataset has the quality to compete with a translated dataset.

As Table 13 showed, a significant proportion suffered from problematic grammar. Our exploration shows that the most prevalent issue of flagged unq? is the proxy model is struggling at choosing the best WH-word when the question uses "yang juga" rule. We attach some of the examples to illustrate the issue:

- **Manila ibu kota dari di mana yang juga negara Angkatan Udara Filipina?** or "Manila is a capital of where that is the also the country of The Phillipines Air Force?" The triples used by the question are:

Table 14 Model accuracy on simple questions

Train dataset	Test TydiQA		Our test	
	F1	EM	F1	EM
Indonesian TydiQA	9.80	1.90	12.48	3.35
Our dataset	20.63	15.93	99.68	99.54
TydiQA + English squad	22.63	11.53	42.29	33.13
Our dataset + English squad	57.03	47.96	99.78	99.65
English squad	64.28	50.97	96.78	96.27
Translated	46.68	37.52	94.90	90.00

`wd:Q252wdt:P36wd:1461`(Manila, capital of, Philippines)
`wd:Q2327398wdt:P17wd:1461`(Philippine Air Force, country, Manila)

  - Besides having the word WH-word, the problem of these combination is that wd:negara does not quite have the right rdfs label to express this question properly.

- **Joshua Suherman lahir di kota sejuta apa dan provinsi Terminal Purabaya?** Joshua Suherman was born in which million city and province Terminal Purabaya? The triple used for this question:

`wd:Q1393191wdt:P19wd:Q11462`(Joshua Suherman, place of birth, Surabaya)
`wd:Q7260794wdt:P131wd:Q11462`(Purabaya bus station, located in, Surabaya)

  - Surabaya's entity type is city and million city. Million city is Wikidata's entity type for a city whose poulation is at least one million people.For this question, "what city" would have been a better WH-word selection than "what million city".

6.2 Computer evaluation

Table 14 showed the accuracy of the model when they are trained with various training dataset and either the test fold of TydiQa or the test fold of our produced simple question dataset, including introducing the Absorption Step in the train & test fold division to prevent test leak.

The tables showed mixed results. The small size of the Indonesian TydiQA dataset explains why their respective model underperforms. However, it does not make sense why TydiQA combined with the English SQuAD is outperformed by our dataset combined with the English SQuAD or the translated dataset. One will assume that both latter datasets will have noise due to an imperfect translation algorithm, and this noise causes the model to underfit. Both datasets underperforming the English Squad confirm this expectation. However, one will expect that the nativeness of

Table 15 Model accuracy on complex questions

Training dataset	?shr		?unq		shr?		unq?	
	F1	EM	F1	EM	F1	EM	F1	EM
Simple question only	99.92	99.57	99.36	98.22	99.33	99.12	97.91	97.91
Combined with complex question	99.89	99.57	99.22	98.22	98.75	98.48	98.24	97.91
Translated SQuAD	93.15	92.07	89.15	85.53	94.25	92.87	87.02	84.58

the TydiQA will complement the volume of English Squad to create the best model. We blame this on domain shift. Domain shift happens when a model is tested on a dataset sampled from a different distribution than the training dataset. In this case, the nature of the label remains the same. However, the feature changed by translation and context source Jia and Liang (2017) showed that something as innocent as adding one similar sentence to the model could cause a significant accuracy drop. Subtle changes in writing style can also be a disturbance. This is confirmed by our dataset having the best accuracy with our test dataset. The result suggests that our dataset has a competitive accuracy. However, future users of this dataset must exercise caution regarding domain shift, especially for cross-domain applications (Table 14).

Our training dataset has the best accuracy for the complex questions. The performance between the simple-question-only dataset or the dataset combined with the complex dataset are approximately similar (Table 15).

6.3 Domain shift hypothesis

To demonstrate our hypothesis of domain shift, we employ SentenceBert (Reimers & Gurevych, 2019). SentenceBert pools token embedding into a unified sentence embedding. However, further training the BERT embedding with a task-specific loss function refines it. In our case, we are concerned about whether the two questions are semantically similar. For this purpose, the paper used the natural entailment dataset of the SNLI and MNLI. These tasks give a model a pair of sentence, and they must tell whether the sentence contradicts each other, entails each other, or neither. For our pre-training, two BERT model, twinned in a Siamese network, embeds each sentence. The element-wise difference of each embedding cell is concatenated with the embedding of both sentences. This concatenated feature is used for classification by the model. The idea of the model is, especially with the element-wise difference feature in mind, that we want to pull the embedding of similar sentences as close as possible and push the embedding of different sentences as far as possible.

For this hypothesis, a pre-trained M-BERT model is used. 5700 Indonesian questions from the TydiQA's training fold and a sample of 5700 simple questions from our training fold form the corpus, 565 Indonesian questions from the TydiQA's test fold, and a sample of 565 simple questions from our training fold form the corpus, 565 Indonesian questions from the TydiQA's test fold. We retrieved their embedding and passed them to an auto-encoder. This autoencoder is trained with

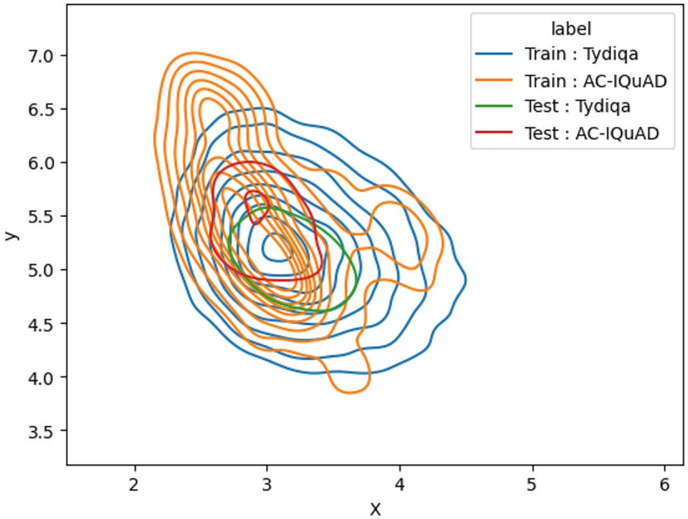

Fig. 5 KDE plot of the distribution of the autoencoded SentenceBert embedding of dataset entry

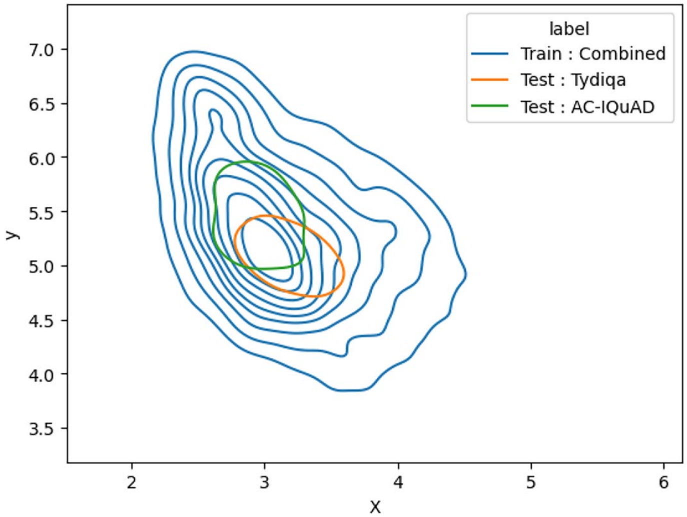

Fig. 6 KDE plot when the two training dataset is combined

M-SentenceBert embedding of 250,000 Indonesian sentences, taken from Wilie et al. (2020)'s Indonesian web dump. The auto-encoder produces an embedding of 2 elements. These two elements are plotted on the graph. To better visualize the contour, the distribution is plotted with KDE.

Figure 5 showed how different the distribution of the dataset is. The central part of the train TydiQA dataset almost coincides withe the test dataset. Meanwhile, our train AC-IQuAD distribution is more haphazard. When the training dataset is merged, as shown on Fig. 6, the distribution between the train and test no longer matches "quite nicely," demonstrating the domain shift in action. To have a numerical feel of this difference, we performed Max Mean Discrepancy, with the following result:

- The MMD distance between the train and the test of TydiQA is 0.005.
- The MMD distance between the train of TydiQA and our train is 0.074.
- When the train is combined, the MMD distance between it and TydiQA is 0.034.

As the MMD showed, the distribution is shifted when the dataset is combined.

7 Conclusion

We produced an Indonesian QA dataset by leveraging Wikidata knowledge graph. The dataset is available online in JSON format.[3] We showed that it has the potential, as indicated by manual evaluation from humans and automatic evaluation with QA model training. However, we highlighted that there are issues such as the possibility of domain shift and topic issues due to the corpus. Future research should focus on addressing these limitations.

Acknowledgements The funding to cover the publishing fee is covered by the Faculty of Computer Science, Universitas Indonesia. The fund to perform the human evaluation is from personal funding.

Author Contributions KD: Conceptualisation, Investigation, Project administration, Methodology, Resources, Supervision, Validation, Writing-original paper. AAK: Funding acquisition, Validation Refinement, Validation Result Interpretation, Writing-original paper.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

[3] https://www.kaggle.com/datasets/realdeo/indonesian-qa-generated-by-kg.

References

Carrino, C. P., Costa-jussà, M. R., & Fonollosa, J. A. R. (2020). Automatic spanish translation of squad dataset for multi-lingual question answering. In N. Calzolari, F. Béchet, & P. Blache, et al (Eds.), *Proceedings of the 12th language resources and evaluation conference*, LREC 2020, May 11–16, 2020 (pp. 5515–5523). European Language Resources Association. https://aclanthology.org/2020.lrec-1.677/

Clark, J. H., Palomaki, J., Nikolaev, V., et al. (2020). Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics, 8*, 454–470. https://doi.org/10.1162/tacl_a_00317

Devlin, J., Chang, M., & Lee, K. et al. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019*, June 2–7, 2019 (Vol. 1, (Long and Short Papers), pp. 4171–4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1423

Dubey, M., Banerjee, D., & Abdelkawi, A. et al (2019). Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In: Ghidini C, Hartig O, Maleshkova M, et al. (Eds.), *The semantic web—ISWC 2019—18th international semantic web conference*, October 26–30, 2019, Proceedings, Part II, Lecture Notes in Computer Science (Vol. 11779, pp. 69–78). Springer. https://doi.org/10.1007/978-3-030-30796-7_5

Eyal, M., Baumel, T., & Elhadad, M. (2019). Question answering as an automatic evaluation metric for news article summarization. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019* June 2–7, 2019 (Vol. 1 (Long and Short Papers), pp. 3938–3948). Association for Computational Linguistics. https://doi.org/10.18653/v1/n19-1395

Ghosh, I. (2020). Ranked: The 100 most spoken languages around the world. https://www.visualcapitalist.com/100-most-spoken-languages/

Heilman, M., & Smith, N. A. (2010). Good question! statistical ranking for question generation. In *Human language technologies: Conference of the North American chapter of the association of computational linguistics, proceedings*, June 2–4, 2010 (pp. 609–617). The Association for Computational Linguistics. https://aclanthology.org/N10-1086/

Jia, R., & Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In M. Palmer, R. Hwa, & S. Riedel (Eds.), *Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017*, September 9–11, 2017 (pp. 2021–2031). Association for Computational Linguistics, https://doi.org/10.18653/v1/d17-1215

Lample, G., Ott, M., & Conneau, A. et al (2018). Phrase-based & neural unsupervised machine translation. In E. Riloff, D. Chiang, & J. Hockenmaier, et al (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing*, October 31–November 4, 2018 (pp. 5039–5049). Association for Computational Linguistics. https://aclanthology.org/D18-1549/

Lehmann, J., Isele, R., Jakob, M., et al. (2015). Dbpedia—A large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web, 6*(2), 167–195. https://doi.org/10.3233/SW-140134

Lewis, PSH., Denoyer, L., & Riedel, S. (2019). Unsupervised question answering by cloze translation. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, ACL 2019*, July 28–August 2, 2019 (Vol. 1: Long Papers, pp. 4896–4910). Association for Computational Linguistics. https://doi.org/10.18653/v1/p19-1484

McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia Medica, 22*(3), 276–282.

Muis, F. J., & Purwarianti, A. (2020). Sequence-to-sequence learning for Indonesian automatic question generator. In *2020 7th international conference on advance informatics: Concepts, theory and applications (ICAICTA)* (pp. 1–6). https://doi.org/10.1109/ICAICTA49861.2020.9429032

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. In I. Gurevych, Y. Miyao (Eds.), *Proceedings of the 56th annual meeting of the association for computational linguistics, ACL 2018*, July 15–20, 2018 (Vol. 2: Short Papers, pp. 784–789). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-2124. https://aclanthology.org/P18-2124/

Reimers, N., Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084

Serban, I. V., García-Durán, A., Gulcehre, C. et al. (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* (Vol. 1: Long Papers, pp. 588–598). Association for Computational Linguistics. https://doi.org/10.18653/v1/P16-1056, https://aclanthology.org/P16-1056

Siblini, W., Pasqual, C., & Lavielle, A. et al. (2019). Multilingual question answering from formatted text applied to conversational agents. CoRR abs/1910.04659. http://arxiv.org/abs/1910.04659

Turk, A. M. (2017). Mturk is now available to requesters from 43 countries. https://blog.mturk.com/mturk-is-now-available-to-requesters-from-43-countries-77d16e6a164e

Vrandecic, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Communication of the ACM, 57*(10), 78–85. https://doi.org/10.1145/2629489

Weissenborn, D., Wiese, G., & Seiffe, L. (2017). Making neural QA as simple as possible but not simpler. In R. Levy, & L. Specia (Eds.), *Proceedings of the 21st conference on computational natural language learning (CoNLL 2017)*, August 3–4, 2017 (pp. 271–280). Association for Computational Linguistics. https://doi.org/10.18653/v1/K17-1028

Wilie, B., Vincentio, K., & Winata, G. I. et al. (2020). Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. In K. Wong, K. Knight, & H. Wu (Eds.), *Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing, AACL/IJCNLP 2020*, December 4–7, 2020 (pp. 843–857). Association for Computational Linguistics. https://aclanthology.org/2020.aacl-main.85/