QUA-RC: the semi-synthetic dataset of multiple choice questions for assessing reading comprehension in Ukrainian

Mariia Zyrianova, KTH Royal Institute of Technology, Stockholm, Sweden mariiaz@kth.se
Dmytro Kalpakchi, KTH Royal Institute of Technology, Stockholm, Sweden dmytroka@kth.se

Abstract In this article we present the first dataset of multiple choice questions (MCQs) for assessing reading comprehension in Ukrainian. The dataset is based on the texts from the Ukrainian national tests for reading comprehension, and the MCQs themselves are created semi-automatically in three stages. The first stage was to use GPT-3 to generate the MCQs zero-shot, the second stage was to select MCQs of sufficient quality and revise the ones with minor errors, whereas the final stage was to expand the dataset with the MCQs written manually. The dataset is created by the Ukrainian language native speakers, one of whom is also a language teacher. The resulting corpus has slightly more than 900 MCQs, of which only 43 MCQs could be kept as they were generated by GPT-3.

1 Introduction

Assessing reading comprehension is of interest both for the native speakers of any language (for instance, through PISA (OECD, 2019) assessments), and for the foreigners learning the language (e.g., through IELTS[1] for English, DELE[2] for Spanish, or DELF[3] for French). In both cases the skills are frequently assessed on the same scale, namely the one proposed by the Common European Frame of Reference (CEFR; Council of Europe (2001)). One of the assessment formats recommended on any CEFR-level is multiple choice questions (MCQs), which consist of the following components:

- *stem*, typically a question inquiring about some information from the text;
- *key*, the correct answer for the stem;
- *distractors*, wrong but plausible options.

The key and the distractors together are called *alternatives*. Note that reading comprehension MCQs require carefully selected texts, which are absolutely crucial, since reading comprehension MCQs are not designed to stand on their own.

In practice, the assessment with MCQs is rather popular because it enables fast, automatic, and thus objective grading. On the other hand, creating MCQs is comparatively slow and requires a lot of manual efforts, which motivated the research on NLP methods for generating MCQs automatically. As Ch and Saha (2018) report, researchers have tried different techniques for MCQ generation, ranging from the manually created pipelines to more recent methods based on learning from data. Indeed, the introduction of large language models (LLMs), such as BERT (Devlin et al., 2019), or GPT-3 (Brown et al., 2020), resulted in new approaches being tested for many NLP tasks, not least for MCQ generation, especially for English (Vachev et al., 2022; Raina and Gales, 2022; Dijkstra et al., 2022). By comparison, MCQ generation problem (particularly for reading comprehension) received much less attention in other languages, and especially in Ukrainian. In this work we aim to bridge the gap for Ukrainian by making the following contributions:

- We present the first (to the best of our knowledge) dataset of Ukrainian MCQs for reading comprehension called QUA-RC. The dataset contains more than 900 MCQs (for example, the English translation of one such MCQ is provided in Figure 1), and is designed with the Ukrainian-first mindset (instead of being a translation of another dataset). The texts are taken from the real-world Ukrainian reading comprehension tests, and the MCQs themselves are created semi-automatically using GPT-3 (zero-shot), followed by manual curation and then manual expansion of the dataset.

[1] https://www.ielts.org/
[2] https://www.dele.org/
[3] https://fiaf.org/exams/delf-dalf/

- At the same time, we evaluate GPT-3 on the task of generating MCQs for reading comprehension in Ukrainian in a zero-shot manner. Our evaluation reveals extensive shortcomings of this approach with less than 10% of MCQs judged to be of sufficient quality.

Both the dataset, and the accompanying source code are available on GitHub: https://github.com/dkalpakchi/QUA-RC.

Text:
[...] Is there at least one city in Ukraine that can be viewed as an example in these terms? "It is Lviv, which is a pioneer city and a role model for the whole country in the attitude towards animals. There is an excellent communal enterprise that registers pets, keeps a clear electronic account of homeless four-legged friends and tracks their number," says Oleksandra Mezinova, head of the Kyiv animal shelter.

Stem:
Which Ukrainian city is seen as exemplary in its attitude to animals?

Alternatives:
(A) Kyiv
(B) Lviv
(C) Kharkiv
(D) Zaporizhzhia

Figure 1: An example MCQ with an accompanying text from the collected QUA-RC dataset (translated from Ukrainian into English). The alternative in **bold** denotes the key, whereas all the other alternatives (in this case (A), (C), and (D)) denote the distractors.

2 Related work

To the best of our knowledge there has been no prior work on creating datasets of MCQs specifically for Ukrainian first, let alone semi-automatically.

In parallel with this work Bandarkar et al. (2023) have developed Belebele benchmark where they have created a parallel reading comprehension dataset in 122 languages, with Ukrainian being among them. The texts and MCQs in the dataset have been manually translated from English with reportedly rigorous curation process. The texts for this dataset were taken from three sources: WikiNews, WikiVoyage and WikiBooks. By their nature, such texts contain mostly facts, lacking, for instance, literary devices or dialogues, and often appear additionally structured (compared to narrative texts) for ease of reading. Moreover, the translations for the dataset were produced to maximize the alignment between 122 languages, which could lead to the increased use of Translationese (Gellerstam, 1986), as the authors themselves note. By contrast, the texts used in our dataset are taken directly from the Ukrainian national tests for reading comprehension, meaning they are guaranteed to not contain Translationese, and are considered to be of suitable quality by the experts.

The translated datasets in Ukrainian are scarce even when looking at the broader field of Question Answering. The only work that we are aware of is an attempt at translating the SQuAD dataset (Rajpurkar et al., 2016) to Ukrainian[4]. However, it is unclear to what extent the translations have been curated, and the dataset contains no distractors (similar to the original SQuAD).

In general, the idea of creating synthetic QA datasets is not new, and has been rejuvenated by the advent of Large Language Models (LLMs). For instance, Alberti et al. (2019) produced synthetic question-answer pairs by using three different BERT (Devlin et al., 2019) models fine-tuned on SQuAD2 (Rajpurkar et al., 2018) to perform three different tasks: (1) extract the potential answer, (2) generate the question for that answer, and (3) answer this new question to check for the roundtrip consistency and filter-out the inconsistent questions.

The idea of creating synthetic MCQ datasets is not new either. For instance, Kalpakchi and Boye (2023) generated MCQs using OpenAI's GPT-3 (Brown et al., 2020) in a zero-shot manner. After curating the output, 44% of MCQs turned out to be of acceptable quality. In this work we build on the work of Kalpakchi and Boye (2023) and expand it in the following ways:

- we perform our experiment in Ukrainian, which differs from English much more than Swedish, in multiple ways: (1) it uses a different script, (2) it is characterised by a relaxed word order, and (3) it is more morphologically complex;
- our prompt attempts for a fine-grained control by requesting MCQs with a different number of alternatives (e.g., one MCQ with two alternatives, three MCQs with three alternatives, and two MCQs with four alternatives) to get an indication of the extent to which such format control is possible;
- we removed the request for MCQs of varying complexity since GPT-3 could not arrange the MCQs in the order of increasing complexity, as reported by Kalpakchi and Boye (2023).

[4] https://huggingface.co/datasets/FIdo-AI/ua-squad

Additionally, in contrast to Kalpakchi and Boye (2023), we also attempt to revise the generated MCQs that did not meet the quality standards. Furthermore, we expand the dataset with manually written MCQs, instead of relying entirely on the synthetically generated MCQs, thus taking a semi-automatic approach. We also conduct a pilot investigation and check to what extent the synthesised MCQs could inspire the creation of the new ones.

3 Data

Any MCQ dataset for reading comprehension consists of the texts and MCQs based on these texts. The choice of texts is crucial in this endeavour as it partly defines what kinds of MCQs would appear in the dataset (e.g., those testing simple text scanning skills, or more advanced, asking the reader to compare or contrast). In this paper we took the texts from the Ukrainian national tests in the Ukrainian language and literature, which are part of the university admission exams in Ukraine, called External independent evaluation, EIE (*Ukr.* "Зовнішнє незалежне оцінювання, ЗНО"). Specifically, we took the texts from the "Reading" section of the tests administered between 2007 and 2021 (the last year before the radical change of format). We have cleaned the texts by removing titles and/or subtitles of the original texts, numeration of the text parts, and other notes (e.g., names of the authors, number of the words included to the text). Additionally, we have filtered out texts that included non-continuous elements (following the definition of OECD (2019), e.g., lists) or relied on images for the narration.

Furthermore, we were forced to split the vast majority of the texts into parts, which resulted in 62 excerpts from the 32 original texts. The reason behind the aforementioned splitting is illustrated by Figure 3, which shows that one word in Ukrainian corresponded to between slightly less than 7 and 8.5 GPT-3 tokens, (in stark contrast to roughly 1.33[5] tokens per word for English).

While the aforementioned problem is often solved using the sliding window approach, we would like to argue that it is not sufficient for this particular problem. The reason behind this is that the generated MCQs need to go beyond "local" factual questions about the information that is presented in a couple of sentences. Indeed, we are also interested in the MCQs that test higher-order reading skills (e.g. making high-level inferences or drawing conclusions from a text), for instance, MCQs with such stems as "What is the main idea of the text?", "Why did X do Y and not Z?", or "What is the relationship between X and Y, according to the text?", which are prevalent in real-world reading comprehension tests. If we require a model to generate such stems in a reliable way, the whole text must be provided to a generation model.

Bearing in mind that we asked GPT-3 to generate N_q MCQs per text, we have empirically identified that the excerpts should be at most 250 words long to allow enough space for the MCQs themselves. Additionally, we took only those excerpts which discuss a particular topic and/or convey a certain idea, so that each of them can be perceived as a standalone text. The extracted 62 excerpts are divided into the following three types:

- *Narrative* texts mainly convey facts or tell a story informing the reader about something or somebody. The texts can be of an *encyclopedic* nature (providing summarised knowledge on a certain object or phenomenon), or *biographical* (narrating life of famous people). Contrary to the Wikipedia-style factual texts, narrative texts in our dataset include literary devices (e.g., metaphors).
- *Descriptive* texts portray something or somebody by giving detailed characteristics of their appearance or features. These texts can include elements of narrative texts.
- *Argumentative* texts convey a certain opinion or a set of opinions (of one or several people) aiming to persuade the reader and/or encourage them to take a certain action. These texts can include elements of narrative and/or descriptive texts.

Later in the article we will refer to these 62 excerpts as simply *texts*.

4 Method

In this work we have investigated the three-stage semi-automatic approach to creating the MCQ dataset. **At the first stage**, we have seeded GPT-3 with the following prompt **in Ukrainian** in an attempt to synthesise N_q^T MCQs:

> Напиши N_q^T різних завдань до даного тексту для перевірки розуміння прочитаного. У кожному завданні має бути одне запитання, пронумероване арабськими цифрами (1, 2, 3 ...). З цих N_q^T завдань S_2^T містити два варіанти відповіді, S_3^T містити три варіанти відповіді, S_4^T містити чотири варіанти відповіді. Варіанти відповіді повинні мати вигляд переліку, позначеного буквами (а, б, в, г). З усіх варіантів другий варіант (б) завжди має бути правильною відповіддю.

[5] Based on the information here: https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

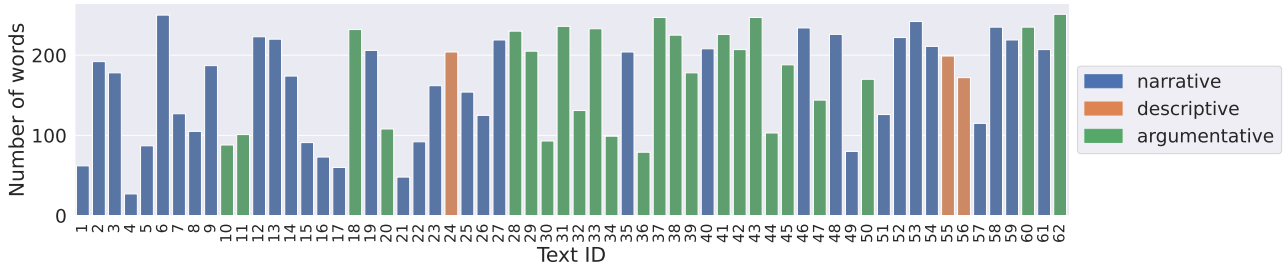

Figure 2: The distribution of the text length in words (defined as space-separated tokens).

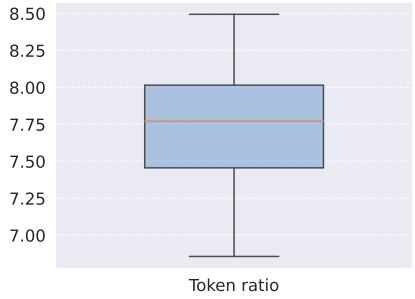

Figure 3: The boxplot showing the distribution of the token ratio $\frac{N_{GPT}}{W_T}$ for the 62 texts from Figure 2, whiskers denote the minimum and maximum values.

> У кожному завданні правильною має бути лише одна відповідь.

To aid the reader, we supply the English translation of the prompt, although we stress again that the prompt was fed to GPT-3 **in Ukrainian**.

> Write N_q^T different reading comprehension tasks for this text. In each task there should be one question, enumerated with arabic numbers (1, 2, 3 ...). From these N_q^T tasks, S_2^T contain two answer alternatives, S_3^T contain three answer alternatives, S_4^T contain four answer alternatives. Answer alternatives should be in the form of a list, marked by letters (а, б, в, г). From all these alternatives, the second alternative (б) must always be the correct answer. In each task there must be only one correct answer.

The number N_q^T was calculated as follows:

$$N_q^T = \max\left(3, \left\lceil \frac{W_T}{\overline{W}} \right\rceil\right) \tag{1}$$

In Equation 1 W_T denotes the number of space-separated tokens in the text T, and $\overline{W}$ denotes the average number of space-separated tokens per text in the corpus. In this article we have empirically calculated $\overline{W} = 14$ based on the collected 62 texts.

Each S_x^T is a string of the form "N_x^T <should>", where N_x^T is the requested number of MCQs with x alternatives, and <should> is the correct form of the Ukrainian verb "мати" (equivalent to the Eng. *should* in this context) grammatically aligned with the number N_x^T, which is calculated as follows:

$$N_x^T = \left\lfloor \frac{N_q^T}{3} \right\rfloor + \mathbb{1}_{N_q^T \% 3 > 4-x} \tag{2}$$

In Equation 2, $\mathbb{1}_{N_q^T \% 3 > 4-x}$ is an indicator function taking the value of 1 if $N_q^T \% 3 > 4 - x$ holds, and 0 otherwise. Since $N_q^T \% 3 \leq 2$, the aforementioned condition enables distributing the remainder $N_q^T \% 3$ roughly equally between N_x^T, by first incrementing N_4^T, and then N_3^T.

At the second stage we went through all synthesised MCQs and divided them into the following three types:

- *Kept* denote MCQs of sufficient quality that did not require any corrections. We use N_k^T to denote the number of such MCQs for the text T.
- *Revised* denote MCQs that were manually corrected keeping the stem, the key, and at least one distractor semantically equivalent to (or even the same with) the original ones. Such correction is possible if the original MCQ meets the following three conditions: (1) it is possible to understand the meaning of the original stem and correct its deficiencies, (2) the key answers the new stem correctly, and (3) at least one distractor is still plausible but wrong for the new stem. If the key was not present in the original MCQ, the condition (2) is ignored, and introducing the key counts as correction. We use N_r^T to denote the number of such MCQs for the text T.
- *Discarded* denote MCQs failing to meet at least one condition for being revised. We use N_d^T to denote the number of such MCQs for the text T.

For the sake of simplicity, we will refer to the discarded MCQs and the original MCQs behind the revised ones

Figure 4: Histogram showing the number of MCQs per text generated by GPT-3. The MCQs are divided into types defined for the second stage. The black dashed lines indicate the *requested* number of MCQs for each text, whereas the height of each bar indicates the *actual* number of generated MCQs. The bars with diagonal hatching indicate the texts for which GPT-3 stopped generating due to reaching the maximum size of its context window (4096 tokens).

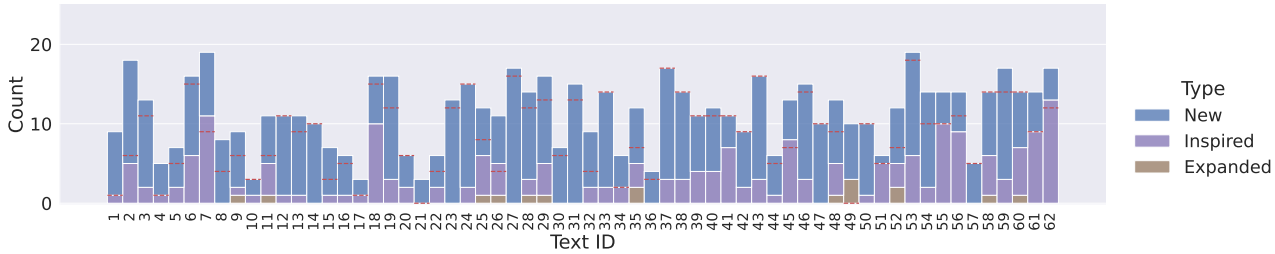

Figure 5: Histogram showing the number of manually added MCQs per text after the third stage. The red dashed lines indicate the minimum required number of MCQs for each text to reach the black dashed lines in Figure 4.

as *MCQs of insufficient quality*. For these MCQs we have identified and categorised the problems causing their poor quality, as described in Section 4.1.

To re-iterate, the introduced revisions are meant to keep the original meaning of the stem and alternatives if it can be derived. If such revisions are impossible, we proceed to the next stage and create a new MCQ.

At the third stage we attempted to complete the dataset with manually written MCQs so that there are *at least* N_q^T MCQs of sufficient quality for *each* text. To be more specific, this means that we needed to write at least $\max(0, N_q^T - N_k^T - N_r^T)$ for each text T. Here we differentiate between three types of MCQs:

- *Expanded* denote MCQs that keep both the original stem and *all* alternatives (consisting of at least the key and one distractor) but introduce more distractors. We use N_e^T to denote the number of such MCQs for the text T.
- *Inspired* denote MCQs which meet at least one of the two conditions: (1) the stem is changed by adding/removing parts compared to the original stem, and (2) none of the distractors are semantically equivalent to any of the original ones. We use N_i^T to denote the number of such MCQs for the text T.
- *New* denote MCQs with entirely new stems, although the alternatives could be taken from the original MCQ(s). We use N_n^T to denote the number of such MCQs for the text T.

Formally, the goal of this stage is that for every text T the following inequality holds:

$$N_n^T + N_i^T + N_e^T + N_r^T + N_k^T \geq N_q^T \tag{3}$$

4.1 Problem categorisation

We have categorised the problems found in MCQs of insufficient quality based on the impact of these problems on the further revision. Some problems required simple fixes of grammatical errors, whereas others forced us to re-write the entire stem. The rule of thumb is that the larger the re-written part is, the more severe the problem is considered. More specifically, we have grouped the problems into the following four categories:

1. *Formatting errors* – the stem or the alternatives do not follow the formatting requested in the prompt. Such errors can be easily edited.
2. *Language errors* – problems related to inaccurate use of language in terms of its syntax, punctuation, grammatical or lexical norms, while the meaning of the stem and the alternatives is clear. Such problems can be fixed by referring to and following a particular language rule or dictionary.

3. *Semantic errors* – problems which (might) lead to misinterpretation of the stem or the alternatives, or completely prevent a reader from understanding the meaning of these. Depending on the type of fault such errors may be fixed by editing of the stem or the alternative(s). Stems which are *not in the interrogative form* are included to this category since it is not always possible to keep the original (often clear) meaning of the stem after transforming it into a question.

4. *Content-related errors* – problems which keep MCQs incomplete (e.g., abruptly cut stem, lack of alternatives) or affect the stem or the alternative(s) so that their meaning does not correspond to that conveyed by the related text. Such problems usually cannot be fixed by editing, so the MCQ is to be completely re-written (though certain elements of it can still be used as a source of inspiration for a new MCQ).

For explanations and examples of individual errors belonging to each category we refer to Appendix A.

5 Evaluation

To categorise the MCQs as outlined in Section 4, we have manually annotated all MCQs generated by GPT-3. The annotations of the generated MCQs were performed by the first author of this paper who has background in teaching. However, we followed an iterative annotation process (annotating – discussing issues – re-annotating) with both authors (native speakers of Ukrainian) contributing to the discussion and re-annotation. The manually added MCQs, created by the first author, were mostly annotated by the second author (although even here we followed the very same iterative annotation process). Both kinds of annotations were performed using the Textinator annotation tool (Kalpakchi and Boye, 2022).

The results of the annotations for the second stage described in Section 4 are presented in Figure 4. As can be seen from the figure, GPT-3 has produced the required N_q^T MCQs only for slightly less than half of the texts (30 out of 62 texts). Interestingly, although in line with findings of Kalpakchi and Boye (2023), for slightly more than half of cases where GPT-3 did not reach N_q^T (18 out of 32) the generation was stopped because of reaching the stop token (hatched bars in Figure 4), and **not** the maximum number of tokens, meaning more MCQs could potentially be generated for 18 texts.

The number of MCQs that could be kept as they are (green in Figure 4) is very low, only 43 of 525 MCQs, and is distributed unequally among the texts. The number of MCQs that could be revised (orange in Figure 4) also differs substantially between the texts. In total for 36 texts (58% of texts) the number of discarded MCQs is larger or equal to the number of kept and revised ones *together*. This observation reveals a substantial problem with using GPT-3 for generating MCQs in Ukrainian, since discarded MCQs are those that could not be revised without re-writing the major parts of the MCQ.

Recall that we have also requested different number of MCQs with two, three, and four alternatives, attempting to keep each number roughly equal to one third of N_q^T. Figure 6 shows the distribution of the number of alternatives for the generated MCQs per text. As can be seen, GPT-3 failed to meet the aforementioned request for *all* texts. For some texts GPT-3 has also generated MCQs with only one alternative, or even no alternatives at all. Most frequently, GPT-3 generated MCQs with either two or four alternatives, with three alternatives being very rare. This suggests that GPT-3 might have an inductive bias towards generating two or four alternatives (as such cases might have been much more frequent in its training data). Additionally, we note that most of the kept MCQs (green in Figure 6) had only two alternatives (which often were of *yes/no* type), of which only *one* was a distractor.

In an attempt to reach N_q^T MCQs per text we have proceeded to the third stage described in Section 4, which is summarised in Figure 5. Note that for all texts where GPT-3 stopped generating MCQs of its own accord we could manually add the required number of MCQs (and beyond that). For 9 texts most of the newly added MCQs were in fact inspired by the deficient ones produced by the GPT-3. This indicates that MCQs produced by the GPT-3 could potentially be used as an inspiration for the MCQs rather than blindly relied upon. At the same time, we note that for 10 texts none of the MCQs produced by GPT-3 provided the inspiration for the new MCQs (entirely blue bars in Figure 5).

In total, our efforts on correcting the generated MCQs and adding the new ones resulted in expanding the dataset from 43 automatically generated MCQs that could be kept as they are, to 926 MCQs. Observe that MCQs with the same stem but with the alternatives of different types are counted as *different* MCQs. To exemplify, consider the stem "Who wrote the stories about Hercule Poirot?", and the following three sets of alternatives: (1) Agatha Christie, Arthur Conan Doyle; (2) An English, A French; (3) A woman, A man. While the stem is the same, the first set of alternatives inquires about full names, the second – about nationalities, and the third – about gender. Depending on the text, some of these things might be stated verbatim, while others would need to be inferred, resulting in MCQs of various difficulty. This is why we count the aforementioned example as three different MCQs with two alternatives each, rather than one MCQ with six alternatives.

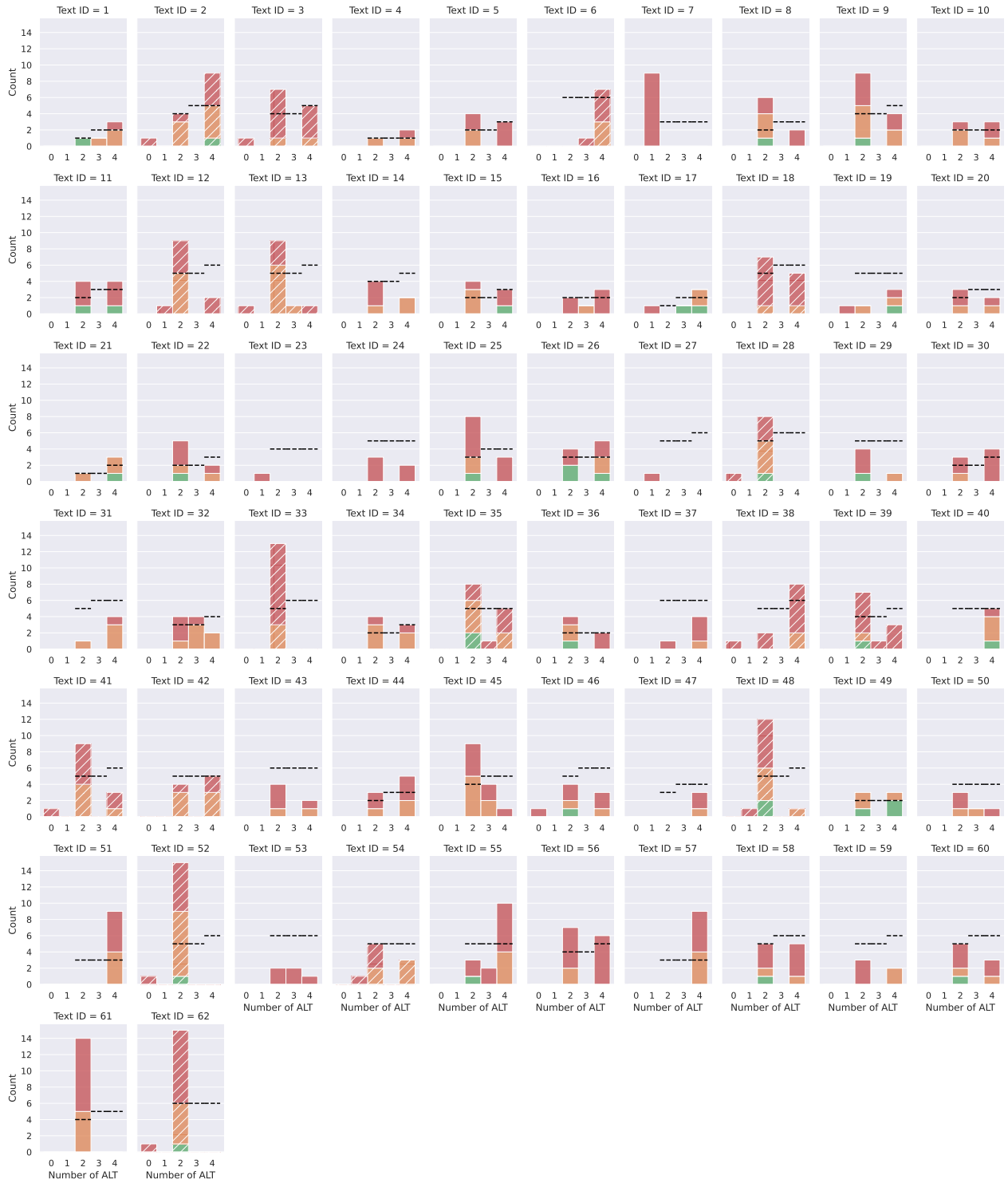

Figure 6: Histogram showing the number of alternatives for MCQs per text generated by GPT-3. The MCQs are divided into types defined for the second stage (the color legend is the same as in Figure 4). The black dashed lines indicate the *requested* number of MCQs with the specified number of alternatives for each text, whereas the height of each bar indicates the *actual* number of generated MCQs with this number of alternatives. Similarly to Figure 4, the bars with diagonal hatching indicate the texts for which GPT-3 stopped generating due to reaching the maximum size of its context window (4096 tokens).

For further analysis of the discarded MCQs and the original MCQs behind the revised ones, the distribution of the identified errors is presented in Figure 7. As can be seen, two the most frequent kinds of errors are associated with the formatting errors (yellow bars in Figure 7), **the least severe category of errors** from Section 4.1. These errors signify MCQs that did not follow the formatting requested in the prompt. For instance, consider the following two MCQs:

> 1. Скільки мільйонів статей має Вікіпедія? а) 280 б) 2 в) Більше двох г) Менше двох Відповідь: Більше двох.
>
> 2. В якому році з'явилася книга «Скаутинг для хлопців»? Відповідь: (а) 1906 (б) 1908.

For this example the exact translations do not matter but note the word "Відповідь" (Eng. *Answer*) that is present in both MCQs. In the first MCQ it comes *after* the four alternatives and provides the correct answer (not following the request in the prompt of simply making the correct answer the second one). In the second MCQ, this word comes *before* the alternatives and is absolutely redundant. While such formatting inaccuracies might seem minor, they impede fully automatic processing of MCQs, i.e. getting the stem, the key and each distractor as separate strings.

The second category of problems by severity is language errors (light orange bars in Figure 7). Observe that fixable grammatical errors in the stem and the alternatives belong to the top three most frequent errors, accompanied by the lexical errors in the stem. One interesting kind of grammatical errors made by GPT-3 is introduced by the use of anglicisms, where words or phrases are translated word-by-word from English, as in the stem below:

> У якій мові вона робить записи українських пісень?
> (*In what language does she record Ukrainian songs?*)

Here the beginning of the stem "У якій мові" is a word-by-word mapping of the English *In what language*, whereas the correct phrase in Ukrainian contains only two words, namely "Якою мовою".
Another example which is likely an anglicism concerns capitalisation of nationalities, as in the stem below:

> Що пропагував Французький літературознавець Жуль Ренар?
> (*What did the French literary critic Jules Renard promote?*)

Here the capitalisation of the nationality *French* is transferred to the stem in Ukrainian as "Французький", although nationalities must not be capitalised in Ukrainian. These two small examples suggest that GPT-3 might be prone to Translationese (Gellerstam, 1986) and use the direct translations of phrases from English. This hypothesis seems plausible given that texts in English constituted 92.64% of the training data of GPT-3, whereas texts in Ukrainian constituted only *0.00763%*[6]. However, further investigations on the matter are required.

An interesting example of lexical errors are russianisms, for instance, as in the stem below:

> Як ван Гог відносився до своєї праці?
> (*How did van Gogh relate to his work?*)

Here the Ukrainian word "відносився" ('vidnosyvsja', Eng. *related*) is likely taken from the Russian "относился" ('otnosilsja', Eng. *treated*), whereas the correct verb in Ukrainian is "ставився" ('stavyvsja', Eng. *treated*). This suggests that the Ukrainian texts that were included in the training data of GPT-3 have not necessarily been lexically correct to the fullest extent, something that should be investigated further.

The third category of problems by severity is semantic errors (dark orange bars in Figure 7). Here the three most frequent errors are all stem-related, namely ambiguous formulation, misleading grammatical errors and too literal text interpretation. The last one is especially interesting, since one of the motivations behind the use of large language models is exactly to avoid such cases. To exemplify, consider the following MCQ generated by GPT-3:

> Text: Галина Бабій, радіожурналіст: Буваючи на відпочинку чи у відрядженнях за кордоном, зауважила, що вільний обмін книжками там дуже поширений. [...]
> (*Halyna Babiy, radio journalist: While on vacation or on a business trip abroad, I noticed that the free exchange of books is very common there. [...]*)
>
> В якому місці Галина Бабій зауважила поширення обміну книжками?
> (*In which place did Halyna Babiy notice the spread of book exchange?*)
> a) у відпочинку (*in vacation*)
> b) за кордоном (*abroad*)

Observe that the MCQ is based on the single provided sentence which itself does not point to a specific place but rather to a situation (on vacation or on a business trip). Hence asking "in what place" is inappropriate in these circumstances, let alone the fact that this detail is very minor and is unlikely to be asked in a real-world reading comprehension test.

[6] As reported here: https://github.com/openai/gpt-3/blob/master/dataset_statistics/languages_by_word_count.csv

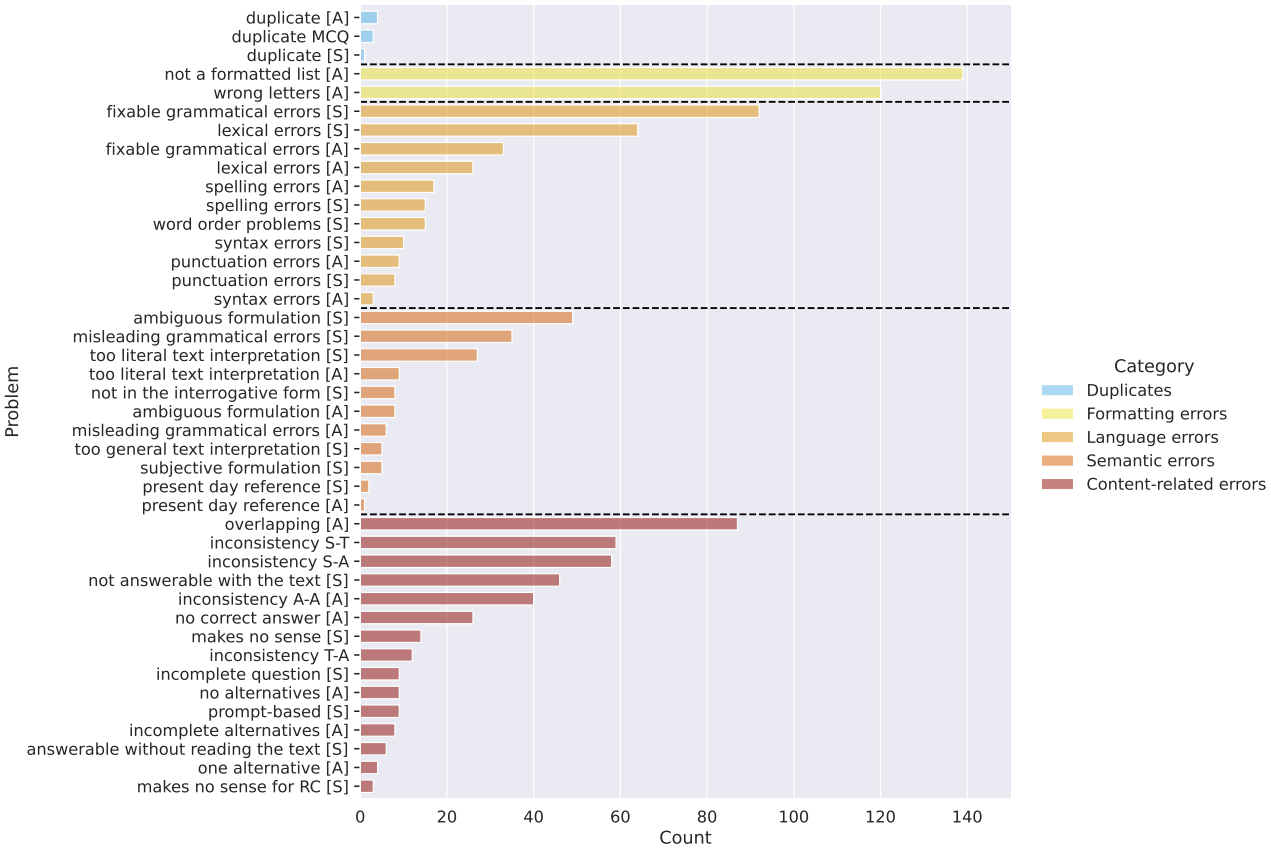

Figure 7: Histogram showing the distribution of problems in the discarded MCQs and the original MCQs behind the revised ones. The problems are categorised as described in Section 4.1, the categories in the legend are ordered from the least severe (at the top) to the most severe (at the bottom). The problems ending with [S] are stem-related, whereas problems ending with [A] are related to at least one of the alternatives within. RC stands for "reading comprehension".

The final and the most severe problem category contains complex problems with the three most frequent being overlapping alternatives, inconsistency between the stem and the text or the stem and the alternatives. Note that the fourth problem which is very close to the TOP-3 signifies stems that are unanswerable by the text. One particularly interesting problem in this category concerns prompt-based MCQs, such as the one below:

> Чи має перелік варіантів відповіді бути позначений буквами (а, б, в, г)?
> (*Should the list of the answer alternatives be marked by the letters (a, b, c, d)?*)
> а) Так (*Yes*)
> б) Ні (*No*)

Clearly, the MCQ above asks about the prompt (provided in Section 4), and not about the content of an actual text. This phenomenon was not observed by Kalpakchi and Boye (2023) when applying GPT-3 for generating MCQs in Swedish. Such discrepancy between our and their findings calls for an empirical investigation across languages and across LLMs aimed at defining the cases when the prompt and the text are not separated by LLMs.

One category that we have not discussed previously are duplicate MCQs (blue in Figure 7). These MCQs constitute cases when the whole MCQ or its part (a stem or a set of alternatives) completely repeats another one or has semantically the same meaning with it. We have not encountered any instances of fully duplicated MCQs, word by word. While some MCQs were semantically equivalent to each other, we have kept them in the dataset.

6 Discussion

The presented dataset of MCQs is created semi-automatically and has its own limitations regarding both the automatic and the manual parts. One limitation that concerns both parts is that the dataset follows no particular principles for ordering either the MCQs for each text, or the alternatives within one

MCQ. While any or both of these could play a role in a real-life testing scenario, we are unaware of any systematic investigation on this matter. Furthermore, any such investigation would be constrained to the particular groups of students, something that is beyond our control in this work. Hence, all alternatives in our dataset are presented in a random order.

Regarding the automatic part, as we have previously discussed, GPT-3 seems to have a number of problems related to Translationese, i.e. applying the phrases or grammar rules of other languages (most notably, English and Russian) to Ukrainian. Bearing in mind that Ukrainian texts constituted only a tiny part of the training data of GPT-3 (0.00763%), such finding is to be expected. Avoiding such problems is one of the strongest arguments either for language models trained specifically for Ukrainian language, or on multilingual language models, where texts in all languages are represented equally.

That said, we do not believe that *fine-tuning* GPT-3 on Ukrainian texts is a feasible way forward due to multiple reasons. First and foremost, to the best of our knowledge, it is currently impossible to estimate what quantity of texts would be enough to reach increase in the model's performance. Secondly, we believe that fine-tuning for this task would require high quality texts in Ukrainian, which are not readily available copyright-free. Lastly, it is very likely that the amount of texts in English in the training data of GPT-3 is higher than all (copyright-free) texts in Ukrainian we will be able to find. All of these arguments together with the cost of fine-tuning GPT-3, and potentially maintaining access to the fine-tuned version for the general public, make such approach practically infeasible for this research.

Another discovery worth further investigation concerns the cases where GPT-3 failed to identify the boundary between the prompt and the supplied text. In our investigation, this manifested itself as MCQs asking about the details of the prompt rather than about the content of the supplied text. Our suggestion is to conduct a systematic investigation on whether such problem occurs across languages and language models (and, ideally, also across NLP tasks).

We have also noticed that GPT-3 did not succeed in "decoding" literary devices (e.g., metaphors, rhetorical figures) and phraseological units, as in the MCQ below:

> Text: Але ретельні рентгенівські дослідження засвідчили, що всі роботи митця написані зі «швидкістю виконання й без вагань», «на одному подиху». [...]
> (*But careful x-ray studies proved that all the works of the artist were written with "speed of execution and without hesitation", "in one breath". [...]*)
>
> На якому подиху були написані всі роботи В. ван Гога?
> (*In what breath were all the works of V. van Gogh written?*)
> а) Довгому (*Long*)
> б) Одному (*One*)
> в) Короткому (*Short*)
> г) Завеликому (*Too large*)

Here *in one breath* is a phraseological unit with a stable meaning of "very quickly, without difficulties", and its component parts cannot be separated from each other. Below there is an example of the MCQ which contains a metaphor:

> Text: Квітка вступила до нью-йоркської консерваторії. Оперне майбутнє не склалося, а її американською «дійсністю» стає... рекламний конвеєр, і ось вона — цей янгол — співає дивним тембром сто мільйонів разів якісь «трелі»-заставки для кока-коли.
> Було в її кар'єрі й залучення до «великого» кіно. Але це так – мимохідь – так і не розквітла для «Оскара». Але родичі чітко усвідомили: призначення цього херувима не кока-кола, а щось неземне. [...]
> (*Kvitka entered the New York Conservatory. The future in the Opera did not materialise, and her American "reality" becomes... an advertising conveyor belt, and here she - this angel - is for a hundred million times singing some "trills" - screensavers for Coca-Cola - in a strange timbre. Her career also involved "big" movies. It was, though, very circumstantial, and she never blossomed for "Oscar". However, her relatives clearly understood: the destination of this cherub is not Coca-Cola ads, but something otherworldly. [...]*)
>
> Що було призначенням херувима Квітки?
> (*What was the destination of the Kvitka's cherub?* or *What was the destination of Kvitka, the cherub?*)
> а) Кока-кола (*Coca-Cola*)
> б) Щось неземне (*Something otherwordly*)

Here *the cherub* is a metaphor to describe Kvitka herself, and not her property; neither was Kvitka a real cherub - something that GPT-3 did not manage to catch.

We note that the aforementioned performance problems were documented when we tested GPT-3 in a zero-shot way (e.g., just a prompt with a task specification, without any examples). It is theoretically possible that giving some examples of texts and MCQs for these texts (i.e. formulating the problem as few-shot) could bring more MCQs of sufficient quality. However, in practice, given that one word in Ukrainian amounts

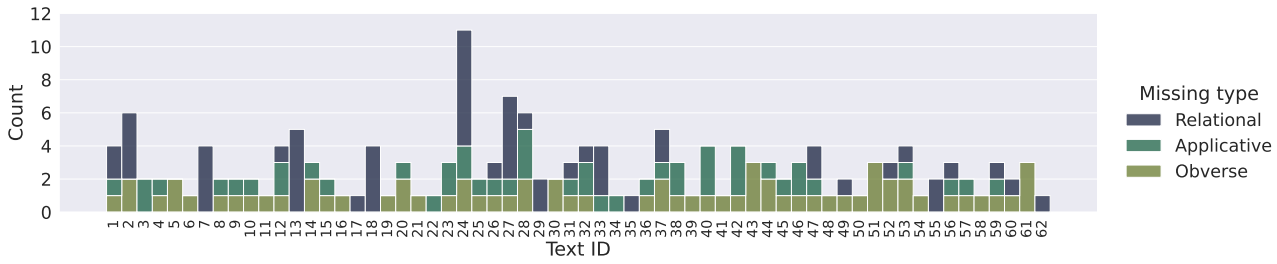

Figure 8: Histogram showing the distribution of new manually written MCQs of three missing MCQ types per text.

on average to about eight tokens, the example texts and their MCQs would take a considerable chunk of the tokens available for GPT-3, leaving little to no space for the actual text and its generated MCQs.

Regarding the manual part, both new and inspired MCQs were written aiming to diversify the MCQs structurally as well as content-wise. For instance, we noticed that certain types of MCQs frequently used in the real EIE tests were absolutely absent from the MCQs generated by GPT-3. Hence, aiming to both increase the diversity and create MCQs resembling the EIE examinations, we attempted to add the missing types of MCQs for each text. More specifically, we limited ourselves to the following three *missing MCQ types*:

- *Relational MCQs* are those asking to establish the relations between two or more elements. Such MCQs are often based on comparison of objects, defining similarities/differences between them or their advantages/disadvantages. Note that the aforementioned relationships should NOT be stated verbatim in the given text for an MCQ to count as relational. For instance, the stem "Яку перевагу бачить автор у театрі перед соціальними мережами?" ("*What advantage of the theater compared to social networks does the author see?*") would give a rise to a relational MCQ, if such advantage is not written verbatim.
- *Obverse MCQs* are those requiring to detect the opposite from that directly stated in the text. In such MCQs, the stem often includes a clause with a negation which is absolutely necessary to find the key correctly. The negation is often expressed by the particle "не" (*not*), or words such as "відсутній" (*absent*), or "заперечувати" (*to deny*). Together with that, if the text itself focuses on describing what is not happening (e.g., factors which do not cause a certain disease) and the stem requires to name the opposite (e.g., what can cause the named disease), such MCQ is also considered obverse. However, if a stem retains the negation which is already stated in the text (e.g., still asking which factors cannot cause the named disease, while the factors are mentioned in the text as those not leading to the disease), such MCQ is not considered obverse.
- *Applicative MCQs* are those asking to apply the knowledge from the text to a hypothetical real-life situation introduced in the stem. Typically, a reader is required to first locate the relevant piece(s) of information in the text, extract the knowledge from there, and then correctly apply this knowledge to the given situation. Note that these MCQs require to extract the established knowledge, and NOT someone's opinion. For instance, "Нобеліантом якої країни стане громадянин України вірменського походження, який на момент присудження премії мешкає у Франції?" ("*A citizen of Ukraine of Armenian origin who lives in France at the time of awarding the prize will become a Nobel laureate of which country?*") is the stem of an applicative MCQ requiring the reader to understand the formal rules for awarding the Nobel Prize.

Our goal was to investigate whether it was possible to manually create at least one MCQ for each of the aforementioned missing MCQ types.

The results of the aforementioned endeavour are presented in Figure 8. As can be seen, we could create an MCQ of at least one missing type for *each text*. At the same time, only 13 out of 62 texts received at least one MCQ of *all* missing types. This shows that not every kind of text could provide grounds for every missing MCQ type. For instance, the obverse MCQs could be created for the vast majority of the texts, since it is usually enough to have a single fact (which are usually abundant in the texts of various genres) for such MCQ. On the contrary, applicative MCQs require the text to include some kind of knowledge that can be applied, which, for instance, immediately excludes vast majority of the biographies and descriptive texts. Similarly, relational MCQs can not be written for each and every text, since they require at least two objects or concepts that could be compared/contrasted. Additionally, shorter texts (especially those consisting of only a couple of sentences) tend to give less opportunities for these kinds of MCQs.

In addition to the MCQ types mentioned above, there are also so-called *tabular* questions. These MCQs are associated with the texts that contain information that could have been arranged in a table. For instance, one of the texts in our dataset describes the history of comic books and includes information about the names of the comic books in different countries. Such information could be represented as a table with the name of the country in one column and the corresponding name of the comic book in another. A tabular MCQ from our dataset for this text is:

> У якому рядку правильно визначено відповідність між країною походження та терміном, що використовується?
> (*In which line the correspondence between the country and the used term is correctly specified?*)
> США - «комікс», Франція - «мальовані історії», Японія - «манга», Україна - «стрічка малюнків»
> (*The US – "comics", France – "drawn stories", Japan – "manga", Ukraine – "picture tape"*)
> США - «комікс», Франція - «стрічка малюнків», Японія - «манга», Україна - «мальовані історії»
> (*The US – "comics", France – "picture tape", Japan – "manga", Ukraine – "drawn stories"*)
> США - «стрічка малюнків», Франція - «комікс», Японія - «манга», Україна - «мальовані історії»
> (*The US – "picture tape", France – "comics", Japan – "manga", Ukraine – "drawn stories"*)
> США - «мальовані історії», Франція - «комікс», Японія - «манга», Україна - «стрічка малюнків»
> (*The US – "drawn stories", France – "comics", Japan – "manga", Ukraine – "picture tape"*)

The notable feature of tabular MCQs is that one could create many MCQs by simply varying the number of items in each alternative (the number of countries in this example), and matching the values of different columns in various ways (grouping the country name with its name of the comic books in this exmaple). In our dataset, we tend to keep only a few examples of tabular MCQs, without providing its all possible variations.

Similarly, MCQs which include names and digits are mainly represented in one variation only, while the names can often be altered from the full to the shortened ones (or vice versa), including those with only first letters of the name left, and the numbers can be represented in digits or in words.

An opposite kind of MCQs with no variation in alternatives, are the yes/no MCQs, which most frequently contain only two alternatives ("Yes" and "No"), sometimes more (e.g., including "Maybe"). Among MCQs generated by GPT-3, only 21 were of such type, of which only one was kept as it was and nine were revised. For such MCQs, the set of alternatives is always fixed. The same concerns the stems like "What is the theme of the text?" where the stem is fixed, while the alternatives change with each new text. Writing such MCQs equates to constructing only a stem or only a set of alternatives which makes the process faster but of the lower priority for automation. Taking that into account, we did not add such MCQs manually.

Another transformation that could expand the dataset is the use of synonymous reformulations or paraphrases of the given stems, which might potentially allow to manipulate difficulty of the MCQs but where an extensive coverage is hardly reachable. Mainly focused on covering content-related aspects, we leave vocabulary alterations and difficulty evaluation for the future work.

However, already from the MCQs included into this dataset (from those created both automatically and manually), we have noticed that their difficulty might vary depending on a personality-based factor (OECD, 2019) of previous knowledge – the knowledge a reader already had before beginning to read the given text. Since such factor can hardly be controlled, it can be tricky to judge whether an MCQ is suitable for testing the reader's reading skills rather than their previous knowledge. For instance, MCQs which are to some extent based on the so-called "common knowledge" may require making complex inferences with respect to the text but become absolutely trivial for people with the relevant previous knowledge. We noticed that the particular kinds of previous knowledge for which it is true are stable facts (those that could be verified from multiple credible sources and are not based on opinions), meanings of idioms, and definitions of terms. To exemplify further, consider the following text and an MCQ:

> Text: Микола Леонтович збирав народні пісні й адаптував їх для хорового співу. [...] Композитор, як різьбяр, зробив навколо основної поспівочки витончену оправу. Поєднавши прийоми народного багатоголосся з досягненням класичної поліфонії, він домігся того, що кожен голос почав відігравати самостійну роль, відтворюючи найтонші зміни настрою. Леонтович кілька разів переробляв твір, аж поки 1916 року не створив досконалий хорал.
> (*Mykola Leontovych collected folk songs and adapted them for choral singing. [...] The composer, like a carver, made an elegant frame around the main song. By combining the techniques of folk polyphony with the achievements of classical polyphony, he made each voice play an independent role, reproducing the most subtle mood*

changes. Leontovych revised the work several times until the year 1916, when he managed to create a perfect chorale.)

Ким був Микола Леонтович за фахом?
(*Who was Mykola Leontovych by profession?*)
а) Композитором (*A composer*)
б) Різьбярем (*A caver*)
в) Хористом (*A chorister*)
г) Етнологом (*An ethonologist*)

Here M.Leonotych's profession might be a completely unknown fact for some people while they are able to infer the information from the text. However, students of a music school or students with broad knowledge in arts and/or Ukrainian culture are likely to answer this MCQ without even reading the text.

Another example of an MCQ which is potentially answerable without reading the text is presented below:

Text: Практика надання допомоги безпритульним тваринам сягає XVII ст. Саме 1695 р. в Японії, у місті Едо (нині Токіо), з'явився перший (з відомих нам) притулок для собак. [...]
(*The practice of helping homeless animals dates back to the 17th century. It was in 1695 in Japan, in the city of Edo (now Tokyo), that the first (we know about) shelter for dogs appeared. [...]*)

Яке місто мало назву Едо?
(*Which city used to be named Edo?*)
а) Токіо (*Tokyo*)
б) Львів (*Lviv*)
в) Київ (*Kyiv*)
г) Кіото (*Kyoto*)

Here students might be completely unaware of the first name of the city of Tokyo; however, it is a stable real-life fact which a student can know from school subjects (e.g., geography, arts) or other sources not related to the reading material used for a reading comprehension test.

In practice it means that no pre-generated set of MCQs can be blindly taken as it is for real-life learning and is still to be verified by a person (likely, a teacher) who knows their target audience, peculiarities of the learning process of this audience, exact objectives of a test, and so on.

7 Conclusions

Despite the format of MCQs being widely used in the Ukrainian educational system, specifically for reading comprehension tests as part of the university admission exams, automatic generation of these questions in Ukrainian has not been introduced yet. Inspired by what has been achieved in the NLP field for other languages, we created a semi-synthetic MCQ dataset for reading comprehension in Ukrainian which can be used as training or evaluation data for models specialising in MCQ generation and answering.

As expected, to achieve the sufficient quality of the dataset, manual editing was necessary to fix the errors made by GPT-3 and diversify the whole dataset by creating additional MCQs. However, the extent to which human assistance appeared necessary is surprisingly high - more than 90 per cent of the generated MCQs. The faults by the model are named and additionally categorised according to their impact on further revision.

We also note that prompted to generate MCQs with a different number of alternatives, GPT-3 failed to meet the request, which means that this aspect of generating tasks in the multiple-choice format appears to be hardly controllable in the zero-shot scenario, which is a likely way real-world teachers would interact with such models.

Additionally, we found that the effective context window size for Ukrainian is much smaller than $\frac{4096}{1.33}$ words, since one word in Ukrainian is roughly equal to 8 tokens of GPT-3. Such limitation prevented us from generating MCQs on real-length reading comprehension texts, and calls for development of models with tokenisers keeping the token-to-word ratio closer to 1.

That given, this particular model, GPT-3 (as of July 2023), does not seem to be appropriate for reading comprehension MCQs generation in Ukrainian. However, more tests with other models and languages are needed to determine the extent to which LLMs can be used as a helpful generative tool for the stated task.

Acknowledgements

We gratefully acknowledge the financial support for this research provided by the Knut and Alice Wallenberg Foundation to the first author.

References

Alberti, Chris, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Bandarkar, Lucas, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension

dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ch, Dhawaleswar Rao and Sujan Kumar Saha. 2018. Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25.

Council of Europe. 2001. *Common European framework of reference for languages: Learning, teaching, assessment*. Cambridge University Press.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dijkstra, Ramon, Zülküf Genç, Subhradeep Kayal, Jaap Kamps, et al. 2022. Reading comprehension quiz generation using generative pre-trained transformers.

Gellerstam, Martin. 1986. Translationese in Swedish novels translated from English. *Translation studies in Scandinavia*, 1:88–95.

Kalpakchi, Dmytro and Johan Boye. 2022. Textinator: an internationalized tool for annotation and human evaluation in natural language processing and generation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 856–866, Marseille, France. European Language Resources Association.

Kalpakchi, Dmytro and Johan Boye. 2023. Quasi: a synthetic question-answering dataset in Swedish using GPT-3 and zero-shot learning. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 477–491, Tórshavn, Faroe Islands. University of Tartu Library.

OECD. 2019. *PISA 2018 Assessment and Analytical Framework*. OECD Publishing, Paris.

Raina, Vatsal and Mark Gales. 2022. Multiple-choice question generation: Towards an automated assessment framework. *arXiv preprint arXiv:2209.11830*.

Rajpurkar, Pranav, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 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.

Vachev, Kristiyan, Momchil Hardalov, Georgi Karadzhov, Georgi Georgiev, Ivan Koychev, and Preslav Nakov. 2022. Leaf: Multiple-choice question generation. In *European Conference on Information Retrieval*, pages 321–328. Springer.

A Error typology

In this section we present a more detailed account of the problem categories in Section 4.1. The errors within each category are listed in the alphabetic order.

Formatting errors

1. *Not a formatted list* – the generated alternatives are arranged in a line, which does not comply with the prompt and thus is considered to be an error made by the model.
2. *Wrong letters* – the generated alternatives are marked with symbols other than prompted (а, б, в, г), which also includes upper case or a different script (e.g., latin) used by the model.

Language errors

1. *Fixable grammatical errors* – a phrase/sentence contains a faulty, uncommon, or controversial usage of the Ukrainian language (e.g., misuse of grammar cases, verb tenses or voices, breaking grammatical alternation rules, etc.), which can be clearly identified and fixed by applying corresponding rules.
2. *Lexical errors* – a word/phrase is used in an inappropriate meaning, repeats another root word or phrase from the same sentence, is missing or redundant to express the intended idea.
3. *Punctuation errors* – punctuation marks are missing, redundant, or incorrectly used in the sentence, according to the Ukrainian grammar.
4. *Spelling errors* – a word/phrase is formed incorrectly in terms of choice of letters, order of letters, capitalisation or usage of special characters/symbols according to the rules of Ukrainian.
5. *Syntax errors* – a phrase/sentence is incorrectly built in terms of agreement between its parts or choice of the parts of speech (mainly function words), which might partially or completely prevent the reader from understanding the meaning.
6. *Word order problems* – an incorrect or awkward placing of the words in a sentence which makes it more difficult to understand the meaning or breaks sentence structure rules fixed in the grammar.

Semantic errors

1. *Ambiguous formulation* – a phrase/sentence is formulated in an unclear way which allows several possible interpretations in the given context.
2. *Misleading grammatical errors* – a faulty, uncommon, or controversial usage of the Ukrainian language (e.g., misuse of grammar cases, verb tenses or voices, breaking grammatical alternation rules, etc.) or a combination of these which prevents from understanding the meaning. The only possible fix to the problem is re-writing the major part(s) of the stem or alternative(s).
3. *Too literal text interpretation* – a phrase or sentence is taken verbatim from the text to the stem or alternative so that a corresponding part of the MCQ sounds incomplete, unclear, or weird.
4. *Too general text interpretation* – a phrase or sentence in the stem is extracted from the text without all necessary details for the stem to be evident, context-related, and answerable with one of the given alternatives.
5. *Not in the interrogative form* – the stem is given in the form of a fill-in-the-gap or continue-the-sentence tasks which does not comply with the prompt and thus is considered as an error made by the model.
6. *Present-day reference* – the requested information is related to the period of time defined by the words “currently”, “recently”, “today” (or similar) in the text, while the same formulation in the stem tends to become irrelevant with the pass of time and might then confuse a reader. Moreover, sometimes the key changes as time passes by, for instance, if the stem inquires about the number of months between “today” and some event, which means that the key will require adjustment with the pass of time.
7. *Subjective formulation* – the stem or alternative requires evaluation of an object or phenomenon based on a reader’s personal opinion, feelings or experience, where the reader’s answer will likely lack grounds for support or disapproval, and consequently, for objective evaluation.

Content-related errors

1. *Answerable without reading the text* – it is possible to answer the question by analysing the stem and alternatives, without reading the given passage.
2. *Incomplete alternatives* – the process of generating alternatives was started but then stopped for some reason, so the alternatives are cut.
3. *Incomplete question* – the process of generating the stem was started but then stopped for some reason, so both the stem and the set of alternatives are cut.

4. *Inconsistency A-A* — alternatives within one and the same MCQ do not correspond in represented type of content. For instance, the stem asks about the kind of the objects, while the alternatives are "long and round" (naming the form), "white and blue" (naming the colour).
5. *Inconsistency S-A* – information requested by the stem does not match with the type of information provided by one or more of the alternatives within one and the same MCQ. For instance, the stem asks about the shape of the objects, whereas at least one of the alternatives provides colors.
6. *Inconsistency S-T* – information requested by the stem is not provided or cannot be inferred from the text.
7. *Inconsistency T-A* — information provided in the text does not correspond semantically to that in the alternative.
8. *Makes no sense* – the combination of words/phrases in the stem makes its meaning either incomprehensible or hardly plausible for a real life context-related situation.
9. *Makes no sense for RC* – the stem is grammatically and semantically correct (or can be easily edited to be correct) but focuses on the details from the given text which are not strictly important to make relevant inferences and understand the meaning.
10. *No alternatives* – not a single alternative was generated by the model; it is, though, possible, that the model still presented the correct answer for the corresponding stem.
11. *No correct answer* – among the generated alternatives, not a single one can be considered as a key to the stem.
12. *Not answerable with the text* – the given text provides no information for a reader to be able to answer the generated stem.
13. *Prompt-based* – the generated MCQ or its part is based on information from the prompt rather than that from the given text.
14. *One alternative* – from the requested number of alternatives (two, three, or four), only one alternative was generated by the model, which does not comply with the prompt and thus is considered as an error made by the model.
15. *Overlapping alternatives* – more than one alternative satisfy the conditions stated in the stem and thus result in more than one key for the MCQ, not complying with the prompt.