Automatically generating question-answer pairs for assessing basic reading comprehension in Swedish

Dmytro Kalpakchi
Division of Speech, Music and Hearing
KTH Royal Institute of Technology
Stockholm, Sweden
`dmytroka@kth.se`

Johan Boye
Division of Speech, Music and Hearing
KTH Royal Institute of Technology
Stockholm, Sweden
`jboye@kth.se`

Abstract

This paper presents an evaluation of the quality of automatically generated reading comprehension questions from Swedish text, using the Quinductor method. This method is a light-weight, data-driven but non-neural method for automatic question generation (QG). The evaluation shows that Quinductor is a viable QG method that can provide a strong baseline for neural-network-based QG methods.

1 Introduction

In this article, we aim to establish a strong non-neural, but still data-driven, baseline for the automatic generation of reading comprehension questions (RC-QG) in Swedish. It is well-known that reading comprehension is a complex and multi-layered process, ranging from the simple decoding of individual words to advanced analytical tasks concerning the quality, veracity and purpose of whole texts (Alderson, 2000; Shaw and Weir, 2007). In this work, we aim at only generating RC questions on the *the information-locating level*, i.e. questions that facilitate the assessment of readers' ability to scan, locate and retrieve relevant information from a single text. This is the most basic level of RC, as identified by the PISA 2018 report (OECD, 2019, p.34), but still a crucial reading ability in need of assessment.

Our approach is data-driven, meaning that no handcrafted rules or linguistic expertise is required. Adapting to new kinds of questions is done via adding new text-question pairs to the training material. Furthermore, the generated questions and their respective answers (the *QA-pairs*) will quite faithfully reuse the wordings of the text, but can also use words and phrases that are not explicitly present in the text. Our system is completely open-source with source code available on GitHub (`https://github.com/dkalpakchi/swe_quinductor`).

2 Related work

To the best of our knowledge, the literature on reading comprehension question generation (RC-QG) particularly for Swedish is very limited. Wilhelmsson (2011, 2012) presented a system for generating questions using manually specified grammatical transformations (syntactic fronting of unbounded constituents and substitution of suitable question elements with question words). By design, their system is able to generate QA-pairs only using formulations that appear word-by-word in the text. The generated questions were limited to two categories, the first of which encompasses questions starting with "vem" (eng. "who/whom"), "vad" (eng. "what"), "vilken" (eng. "which") that would concern nominal constituents. The second category concerns questions to some adverbials. The author also presented a preliminary evaluation of the questions generated for ten random Wikipedia articles, but unfortunately neither specified the exact articles nor released the source code of his system, making a direct comparison impossible.

Lately, neural-network-based text generation methods have become very popular due their impressive results. Indeed, these methods have been applied to RC-QG as well, mostly for English (see e.g. Liao et al. (2020); Dong et al. (2019)). Our goal here is not to compete with these approaches, but rather to present a more light-weight but still strong baseline to which neural methods can be compared, but which is also interesting and useful in its own right.

3 Data

We have used the *SweQUAD-MC* dataset (Kalpakchi and Boye, 2021a) consisting of texts and multiple-choice reading comprehension questions (MCQs) for the given texts. It was created by three paid linguistics students. They were

instructed to formulate unambiguous questions, such that (1) the answer to the question appears verbatim in the text, (2) the question cannot be answered without reading the text (i.e., the text is *necessary*), and (3) the answer does not require extra knowledge not present in the text (i.e., the text is *sufficient*). These characteristics make the dataset suitable for assessing RC on the information-locating level.

The dataset is relatively small with the training set consisting of 962 MCQs, the development (dev) set – of 126 MCQs and the test set – of 102 MCQs. The distribution of the first words[1] in questions in the training set is shown in Figure 1.
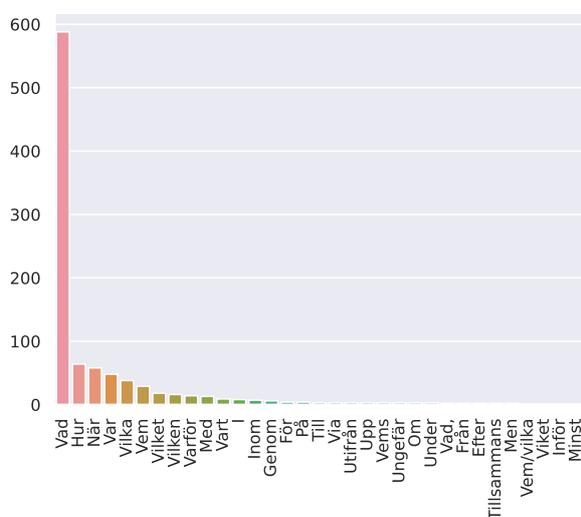
Figure 1: The distribution of the first words in questions of the training set of SweQUAD-MC

4   Method

We used Quinductor (Kalpakchi and Boye, 2021c), which is a mostly deterministic data-driven method applicable to any language having a dependency parser based on Universal Dependencies (Nivre et al., 2020). In particular, it is applicable to Swedish. Quinductor is only capable of inducing QA-pairs based on single declarative sentences, which is suitable for assessing RC on the information-locating level. The method is data-driven, requiring a corpus of texts with the associated QA-pairs as training material, as well as some additional files, detailed in Appendix A.

Quinductor works in two stages: first inducing the templates and then using them to generate questions from unseen sentences. In short, the first stage involves learning from data to express each given
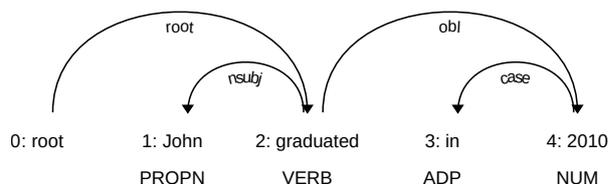

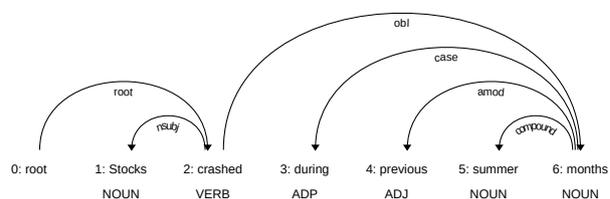Figure 2: The dependency tree for the sentence "John graduated in 2010"

Figure 3: The dependency tree for the sentence "Stocks crased during previous summer months"

QA-pair in terms of the dependency structures of the source sentence, which is a basis for both the question and the answer. To exemplify, consider the sentence "John graduated in 2010" (with its dependency tree in Figure 2) and the question "When did John graduate?" with the answer being in "2010". Quinductor will then learn to induce the template (1) for the question and the template (2) for the answer.

(1) `When did [r.nsubj#1] [r.lemma] ?`
                (John)   (graduate)

(2) `<r.obl#4>`
        (in 2010)

At test time, Quinductor applies an overgenerate-and-rank strategy, and attempts to apply all of the induced templates to each given sentence. Clearly, the more sentences with similar dependency structures will be present in unseen data, the more successful the method will be. Also, the more template expressions with angled brackets (matching a whole phrase, as in (2) above) are present in a template, the higher the generalization chance is. To exemplify, if we get a new sentence "Stocks crashed during previous summer months" (with its dependency tree in Figure 3), our previously induced template will be able to fire and produce the QA-pair "When did stocks crash?" – "during previous summer months", although the trees are clearly not identical. For further details on the method and its generalization capabilities we refer to the original article.

In this work we have induced templates based on the training set of SweQUAD-MC (see more

Statistics	Value
Number of induced templates	248
Support per template	
Mean ± STD	1.04 ± 0.23
Median (Min - Max)	1 (1 - 3)

Table 1: Descriptive statistics of the templates produced using the training set of the SweQUAD-MC dataset. Support per template is the number of sentences from the training set that yield the same template.

implementation details in Appendix A). As can be seen in Table 1, this resulted in 248 templates with most of them being unique, i.e. induced from (being supported by) only one sentence from the training set. Some templates had higher support with the maximum of 3 sentences per template.

5   Evaluation and discussion

The proportion of source sentences (the ones where the correct answer is found) from the dev. and test sets of SweQUAD-MC, for which Quinductor could generate something is reported in Table 2.

For evaluation, we took all 29 QA-pairs generated for the development set and all 24 QA-pairs

	dev	test
# of questions in the set	126	102
# of generated questions	207	213
(1) SS with ≥ 1 applicable template	64	44
(2) SS with ≥ 1 generated question after basic filtering	49	36
(3) SS with ≥ 1 generated question after mean filtering	29	24
(3) as % of the respective set	23%	23.5%
Generated questions per SS		
Mean	3.2	4.8
Standard deviation	4.1	7.2
Median	2	3
Minimum	0	0
Maximum	23	46

Table 2: Descriptive statistics of the questions induced on the dev. and test sets of the SweQUAD-MC using the templates, mentioned in Table 1. "SS" stands for "source sentence(s)", i.e., the sentence in which the correct answer is found. "≥ 1 applicable template" means that at least 1 question was induced from the given SS.

generated for the test set of SweQUAD-MC. These 53 QA-pairs were combined with 53 original QA-pairs corresponding to the same source sentences from the respective corpora (later referred to as *gold* QA-pairs). The resulting 106 QA-pairs and the corresponding source sentences formed 106 evaluation triples, and were presented to 2 human judges (one native Swedish speaker and one non-native, but with a high proficiency) in a random order (different for each judge). Following Kalpakchi and Boye (2021c), we required judges to evaluate each triple using a questionnaire consisting of 9 criteria. Each criterion required a judgement on a 4-point Likert-type scale (from "Disagree" to "Agree"). The evaluation itself was conducted online on an in-house instance of Textinator Surveys (Kalpakchi and Boye, 2022). The evaluation guidelines are reported in Appendix B.

Five criteria concerned questions and were formulated as statements asking whether a question:

C1 is grammatically correct ($\uparrow$)

C2 makes sense ($\uparrow$)

C3 would be clearer if more information were provided ($\downarrow$)

C4 would be clearer if less information were provided ($\downarrow$)

C5 is relevant to the given sentence ($\uparrow$)

The remaining 4 criteria concerned the answer and asked whether the suggested answer:

C6 correctly answers the question ($\uparrow$)

C7 would be clearer if phrased differently ($\downarrow$)

C8 would be clearer if more information were provided ($\downarrow$)

C9 would be clearer if less information were provided ($\downarrow$)

$\uparrow$ ($\downarrow$) indicates that the higher (lower) the judgements on the Likert scale, the better.

Following Kalpakchi and Boye (2021c), we have measured IAA using Randolph's κ (Randolph, 2005) and Goodman-Kruskall's γ (Goodman and Kruskal, 1979). The former, κ, ranging between -1 and 1, accounts for agreement in absolute rankings, i.e. being boosted if the annotators gave exactly same score to an evaluation triple. The value of 0 indicates the level of agreement that could

Criterion		dev		test	
		gold	**gen**	**gold**	**gen**
C1 ↑	κ	0.86	0.54	0.83	0.33
	γ	0.92	0.78	0.91	0.55
C2 ↑	κ	0.59	0.49	0.88	0.39
	γ	0.83	0.79	NA/4	0.75

Table 3: Inter-annotator agreement for criteria C1 and C2 on the development and test sets of SweQUAD-MC. $\gamma = $ NA/X means that at least one annotator gave the same score X to all evaluation triples, so it is impossible to count concordant and discordant pairs.

be expected by chance, the positive (negative) values indicate agreement better (worse) than chance. The latter, γ, also ranging between -1 and 1, accounts for agreement in relative ordering, i.e. being boosted if the annotators ordered a pair of two evaluation triples in the same way, no matter the actual scores. $\gamma = 0$ indicates no agreement, $\gamma = 1$ denotes a complete agreement and $\gamma = -1$ hints at a perfect disagreement.

Without a doubt, the two basic criteria, which must necessarily have high judgements in order to even consider looking at all other criteria, are C1 and C2. Table 3 shows that the evaluators had high agreement on both criteria, especially when it comes to relative ordering of triples ($\gamma > 0.5$).

As can be seen in Figure 4, nearly all *gold* questions of the dev and test sets were judged highly on C1 (median ≥ 3). At the same time, 17 ($\sim 58\%$) and 10 ($\sim 42\%$) of the *generated* questions of the dev and test sets respectively were rated highly on C1, resulting in a total of 50.9% between the sets.

Figure 4 also reveals that surprisingly 4 *gold* questions in total between the dev and test sets were judged as **not** making sense (median < 3). At the same time only 11 ($\sim 38\%$) and 8 ($\sim 33\%$) of the *generated* questions of the dev and test sets respectively were rated highly on C2 (median ≥ 3), resulting in a total of 35.8% between the sets. We also observed that only one generated question was judged highly on C2, but lower on C1.

We analyzed further only 19 *generated* QA-pairs between the sets that were judged highly on both C1 and C2. The analysis was carried out in terms of criteria C5 and C6, the agreement on both of which was reasonably high (see Appendix F). Out of these 19, 17 (11 on the dev set and 6 on the test set), were also rated highly (median ≥ 3) on
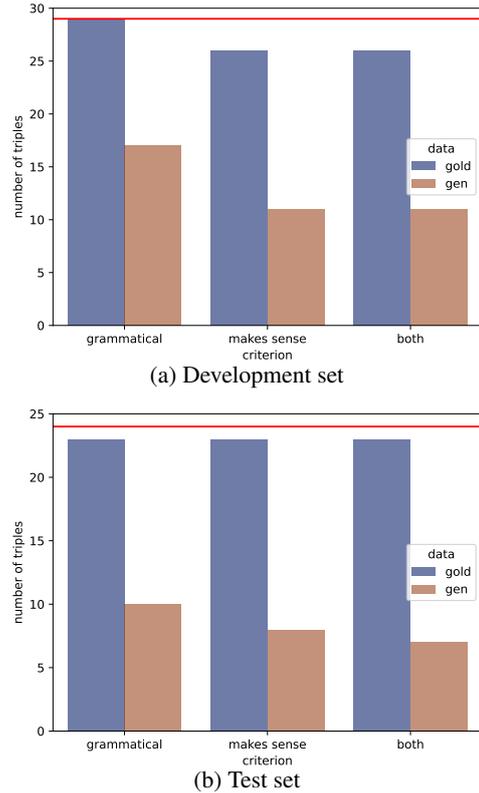

(a) Development set

(b) Test set

Figure 4: Barplots of the number of evaluation triples with rated highly (median ≥ 3) on C1 (grammatical), C2 (makes sense), and both C1 and C2. The horizontal red lines indicate a total number of evaluation triples that were considered (equal for gold and generated data)

C5 and 12 (63.2%), 7 on the dev set and 5 on the test set, were rated highly on C6. Crucially, there were **no** QA-pairs with the suggested answer being judged as correct (high median rating on C6), but with the question being deemed as irrelevant to the given sentence (low median judgement on C5). This means that out of all generated QA-pairs, 12 (22%) were judged as completely valid, 7 (24%) on the dev set and 5 (21%) on the test set, making Quinductor a resonably strong RC-QG baseline for SweQUAD-MC.

Some examples of successful generation are presented in Appendix C. Further analysis (in particular of errors) and related generation examples are presented in Appendix D. Although automatic evaluation metrics provide very limited insights, we also report them in Appendix E, following (Kalpakchi and Boye, 2021c).

Acknowledgments

References

Charles J Alderson. 2000. *Assessing reading*. Cambridge University Press.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.

Leo A Goodman and William H Kruskal. 1979. Measures of association for cross classifications. *Measures of association for cross classifications*, pages 2–34.

Dmytro Kalpakchi and Johan Boye. 2021a. BERT-based distractor generation for Swedish reading comprehension questions using a small-scale dataset. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 387–403, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Dmytro Kalpakchi and Johan Boye. 2021b. Minor changes make a difference: a case study on the consistency of UD-based dependency parsers. In *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, pages 96–108, Sofia, Bulgaria. Association for Computational Linguistics.

Dmytro Kalpakchi and Johan Boye. 2021c. Quinductor: a multilingual data-driven method for generating reading-comprehension questions using universal dependencies. *arXiv preprint arXiv:2103.10121*.

Dmytro Kalpakchi 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.

Yi Liao, Xin Jiang, and Qun Liu. 2020. Probabilistically masked language model capable of autoregressive generation in arbitrary word order. *arXiv preprint arXiv:2004.11579*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, Christopher D Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043.

OECD. 2019. *PISA 2018 Assessment and Analytical Framework*.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Stuart D Shaw and Cyril J Weir. 2007. *Examining writing: Research and practice in assessing second language writing*, volume 26. Cambridge University Press.

Kenneth Wilhelmsson. 2011. Automatic question generation from swedish documents as a tool for information extraction. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 323–326.

Kenneth Wilhelmsson. 2012. Automatic question generation for swedish: The current state. In *Proceedings of the SLTC 2012 workshop on NLP for CALL; Lund; 25th October; 2012*, 080, pages 71–79. Linköping University Electronic Press.

A Implementation details

We have used dependency parsing models from the package called Stanza[2] (Qi et al., 2020), version 1.4.2 and used quinductor[3] package (Kalpakchi and Boye, 2021c), version 0.2.2. This is important, because it is likely that the induced templates will differ if different models are used. The supplementary files necessary for either generating templates or during ranking the generated questions were obtained as follows:

- the IDFs were calculated based on the SweQUAD-MC training set (necessary for inducing templates);

- the morphological n-gram model was calculated based on training, dev and test sets of the UD's Talbanken treebank[4] (necessary for ranking);

---

[2] https://github.com/stanfordnlp/stanza
[3] https://github.com/dkalpakchi/quinductor
[4] https://universaldependencies.org/treebanks/sv_talbanken/index.html

- the question-word model was calculated based on the SweQUAD-MC training set (necessary for ranking).

Note that the question-word model was induced using one of the earlier versions of Stanza, which is why the distributions differ slightly, compared to the 1.4.2. All these suplementary files are released in the GitHub repository, associated with this paper.

B Human evaluation guidelines

Here we report the exact wording of the guidelines presented to the human evaluators.

> Tack att du deltar i vår utvärdering av läsförståelsefrågor! Du kommer att se ett antal meningar (en i taget) tillsammans med fråga och det rätta svaret (QA-par). Du kommer också att se ett antal påstående för varje QA-par. Din uppgift är att bestämma i vilken utsträckning du håller med varje påstående.
> Om frågan är obegriplig ska du välja "1" för alla påståenden relaterade till det föreslagna svaret.
> Vänligen ignorera alla möjliga formatteringsfel, t.ex. skiljetecken (.,!?:;) eller versaler som saknas.

C Generation examples

Sentence 1: inom hälso- och sjukvården arbetar dietisten med nutritionsbehandling och kostrådgivning, både med enskilda patienter och i grupp.
Question 1: var arbetar dietisten med nutritionsbehandling och kostrådgivning?
Suggested answer 1: inom hälso- och sjukvården

Sentence 2: om du är borta längre än ett år eller om du planerar att bosätta dig i ett annat land kan migrationsverket återkalla ditt uppehållstillstånd.
Question 2: vad kan migrationsverket göra om du är borta längre än ett år eller om du planerar att bosätta dig i ett annat land?
Suggested answer 2: återkalla uppehållstillstånd

Sentence 3: hovslagaren behandlar även skador i hoven till exempel hovsprickor eller hovbölder.
Question 3: vilka behandlar hovslagaren?
Suggested answer 3: skador i hoven

Sentence 4: pulverfärg appliceras ofta på metallkomponenter.
Question 4: vad appliceras ofta på metallkomponenter?
Suggested answer 4: pulverfärg

Sentence 5: miljö- och hälsoskyddsinspektörer samarbetar till exempel med andra tjänstemän inom kommuner, företag och myndigheter.
Question 5: vad gör miljö- och hälsoskyddsinspektörer?
Suggested answer 5: samarbetar till med andra tjänstemän inom kommuner , företag och myndigheter

D Error analysis

The QA-pairs that are rated low on C2 (question makes sense) represent the worst kind of errors, which are impossible to correct automatically (like, say, some grammatical errors). Hence, let's examine closer some examples of evaluation triples with generated QA-pairs containing questions that were rated low (median < 3) on C2. Please ignore formatting errors (like absense of punctuation or capital letters).

The questions from the following triples got a median score of 1 on C2:

Sentence 1: arbetsuppgifterna varierar beroende på var man är anställd men den främsta uppgiften är att se till så att hyresgästerna känner sig trygga och nöjda i sin arbets- och boendemiljö.
Question 1: vad gör arbetsuppgifterna beroende på var man är anställd men den främsta uppgiften är att se till så att hyresgästerna känner sig trygga och nöjda i sin arbets- och boendemiljö?
Suggested answer 1: varierar

Sentence 2: du kan då behöva vänta på ditt nya beslut utanför sverige.
Question 2: vad kan du göra på ditt nya beslut utanför sverige?
Suggested answer 2: vänta

The questions from the following triples got a median score of 1.5 on C2:

Sentence 3: den största delen består av barrskog, men i söder breder lummig lövskog ut sig.
Question 3: vad består den största delen från?
Suggested answer 3: barrskog

Sentence 4: en stuntman ersätter filmskådespelare i riskfyllda situationer.
Question 4: var ersätter stuntman?
Suggested answer 4: i riskfyllda situationer

The question from the following triple got a median score of 2 on C2:

Sentence 5: till exempel tillbringar kirurgen en stor del av sin tid i operationssalen, medan en psykiatriker arbetar mycket med samtal med sina patienter.
Question 5: vem tillbringar till en stor del av sin tid i operationssalen medan en psykiatriker arbetar mycket med samtal med sina patienter?
Suggested answer 5: kirurgen

The questions from the following triples got a median score of 2.5 on C2:

Sentence 6: en sjukskriven medarbetare kostar lågt räknat 2 740 kronor per dag*.
Question 6: vad gör en sjukskriven medarbetare?
Suggested answer 6: kostar kronor

Sentence 7: man bör ha god fysik då arbetet kan slita på nacke och axlar.
Question 7: vad bör man göra då arbetet kan slita på nacke och axlar?
Suggested answer 7: ha fysik

Question 1 was generated using the template `vad gör [r.nsubj#1] <r.advcl#2>?`. The problematic one in the case of sentence 1 turned out to be the last template expression `<r.advcl#2>`, which takes the whole subtree, whose root can be found by following the arc labeled `advcl` from the root `r` of the dependency tree. In this case the subtree was too large and thus the generated question contains a lot of unnecessary information. This is one typical type of errors that could be referred to as *overgeneralization*, i.e. the case when the induced templates become too general. However, a balance between letting

Quinductor to induce too generic or too specific templates is difficult to strike, so such kinds of errors are inevitable. Question 5 suffers from the same problem.

Question 2 is based on the template `vad [r.aux#1] [r.nsubj#2] göra <r.obl#3>?`, which in turn was generated from the sentence "vad behöver ambulanssjuksköterskan göra vid större olyckor med många skadade?". The structure is clearly very similar, but the preposition of the oblique nominal happened to be different, so the template expression `<r.obl#3>` did not generalize correctly. In fact questions 3 suffers from similar problems.

Question 4 represents the case when the question word from the template is wrong. All question words are guaranteed to be recorded verbatim as strings and not deduced from the dependency tree of the sentence, so such errors are also inevitable in the current version of Quinductor.

QA-pairs 6 and 7 represent more successful applications of templates for the questions, although the questions are still not perfectly intelligble. The major problem with these QA-pairs are the answers, which in fact suffer an inverse problem compared to that of question 1. For instance, the template for the suggested answer for question 7 is `[r] [r.obj#4]`. The problem with this template is that it just picks up only specific nodes of the dependency tree and is prone to errors if the object, referred by `r.obj#4` will turn out to have some dependents, which will also be relevant to include in the answer. This problem could be referred to as *undergeneralization*. As previously mentioned, the balance between over- and undergeneralization is hard to strike, especially given that Quinductor is a data-driven method and all templates do depend on the data at hand.

We noted that sometimes these errors arise, because of inconsistency of dependency parsers themselves, which is not a new observation. For instance, Kalpakchi and Boye (2021b) showed that even changes as minor as replacing one 4-digit numeral by another can cause surprisingly large inconsistencies in the resulting trees using the state-of-the-art dependency parsers, in particular for Swedish.

E   Automatic evaluation metrics

Following (Kalpakchi and Boye, 2021c), we have calculated BLEU-N, ROUGE-L and CIDEr using

nlg-eval (Sharma et al., 2017) and METEOR using METEOR-1.5 (Denkowski and Lavie, 2014) and specifying the language to be Swedish.

Metric	dev	test
BLEU-1	0.39	0.22
BLEU-2	0.29	0.15
BLEU-3	0.22	0.09
BLEU-4	0.18	0.06
METEOR	0.33	0.21
ROUGE-L	0.38	0.27
CIDEr	0.76	0.84

Table 4: Automatic evaluation on the development and test sets of SweQUAD-MC only for generated questions ranked first.

F Detailed IAA analysis

Inter-annotator agreement (IAA) for criteria C3 - C9 is presented in Table 5. Let's start by analyzing the agreement on the development set. We observed that agreement on C5 - C8 is quite strong for both gold and generated QA-pairs. On the other hand, γ is strongly negative for C3 (the question would be learer if more information were provided), indicating that the annotators relative ranking are almost exactly opposite. At the same time κ on C3 is quite weak as well, especially on the generated QA-pairs. For the generated questions both $\kappa = 0.08$, so $\gamma = -0.7$ indicate almost no agreement in absolute scores and a strong disagreement in relative orderings.

The scores for the gold questions on C3 are much more peculiar, namely $\gamma = -1.0$, while $\kappa = 0.45$, indicating complete opposite relative ordering between the annotators, while also having a moderate agreement in absolute ranks. If we look closer at the data, it turns out that annotator A gave almost all gold triples the score of 1 on C3, except one triple (let's call it T_x) that got the score of 3. Hence, this is the only triple that can be used to establish relative ordering for the annotator A. Now that very same T_x got the score of 1 on C3 from the annotator B. This means that T_x will always be scored higher than all other triples for the annotator A, whereas it will be scored lower than other triples for the annotator B. This, in turn, means that all relative orderings are opposite between the annotators, resulting in $\gamma = -1$. This example is a good illustrataion of why only κ or γ is not enough for

Criterion		dev		test	
		gold	gen	gold	gen
C3 ↓	κ	0.45	0.08	0.83	0.17
	γ	-1.0	-0.7	NA/1	-0.09
C4 ↓	κ	0.95	0.91	1.0	0.83
	γ	NA/1	-1.0	NA/1	1.0
C5 ↑	κ	0.77	0.72	0.67	0.44
	γ	0.82	0.93	NA/4	0.81
C6 ↑	κ	0.72	0.63	0.67	0.56
	γ	0.79	0.88	-1.0	0.78
C7 ↓	κ	0.63	0.49	0.94	0.83
	γ	0.62	1.0	NA/1	1.0
C8 ↓	κ	0.54	0.63	0.94	0.83
	γ	0.4	0.73	NA/1	0.93
C9 ↓	κ	1.0	1.0	1.0	1.0
	γ	NA/1	NA/1	NA/1	NA/1

Table 5: Inter-annotator agreement for criteria C3 - C9 on the development and test sets of SweQUAD-MC. γ = NA/X means that at least one annotator gave the same score X to all evaluation triples, so it is impossible to count concordant and discordant pairs.

assessing the IAA, but indeed both are needed and should be interpreted with caution. A similar situation is observed for the generated questions from the development set on the criterion C4, and for the gold questions from the test set on the criterion C6.