Scientific question generation: pattern-based and graph-based RoboCHAIR methods

Senja Pollak, Vid Podpečan, Janez Kranjc, Borut Lesjak, Nada Lavrač
Jožef Stefan Institute
Jadranska cesta 39, Ljubljana, Slovenia
senja.pollak,vid.podpecan,nada.lavrac@ijs.si

Abstract

RoboCHAIR is a system for generating scientific questions and is implemented as a web interface. We focus on generating questions which can trigger new scientific ideas or bring attention to the elements that need clarifications. This paper extends the initial version of the RoboCHAIR's question generation module by keeping only a selection of best evaluated templates from the initial pattern-based approach and proposes a novel method based on triplet graphs enriched with word embeddings to identify parts of text which require clarifications by the author(s). In this paper the two methods are compared showing that the pattern-based method achieves higher scores.

Introduction

Science begins by asking questions and then seeking answers; children understand this intuitively as they try to make sense of their surroundings (Vale, 2013), and the Socratic questioning is considered a powerful contemporary teaching method (Brill and Yarden, 2003; Vale, 2013). This work addresses the task of automated question generation from scientific papers. The task belongs to the field of scientific creativity (O'Donoghue et al., 2015), a subfield of computational creativity (Boden, 2004; Colton and Wiggins, 2012) that is concerned with developing software that exhibits behaviours reasonably deemed creative.

Automatic Question Generation (AQG) technologies can be used in question-answering (e.g. Kalady, Elikkottil, and Das, 2010), dialogue systems (e.g. Piwek and Stoyanchev, 2010), educational applications, or intelligent tutoring systems (e.g. Sullins et al., 2010; Khodeir et al., 2014). In one of the latest reviews of AQG for educational applications, Kurdi et al. (2020) clearly show that the majority of works focus on questions as assessment instruments. AQG systems predominantly focus on factual questions (e.g. Rus et al., 2011; Heilman and Smith, 2010; Becker, Basu, and Vanderwende, 2012; Wang et al., 2020). This is in line with the assessment perspective, where fact-based answers are then evaluated. On the other hand, some of the projects have focused on the design of web-based systems for student question generation (Yu, 2009; Wilson, 2004; Hazeyama and Hirai, 2007). From a constructivist perspective to learning that aims to engage students in meaningful and understandable tasks about which they can reflect abstractly, systems that support student question generation are useful (Steffe, 1991; Geelan, 1997; Yu, 2009; Yu and Liu, 2009). In one of the systems focusing on critical thinking in academic writing (Liu, Calvo, and Rus, 2014), the authors propose an automated system helping students in critical literature review writing by generating a contextualised feedback in the form of trigger questions. In our approach, the focus is also not on factual question-answering, but we aim at posing the questions related to expressing decisions and opinions and identifying under-specified elements in the paper. By moving away from the assessment oriented factual question generation framework, our system aims to mimic human intelligence and creativity of a scientific audience. Instead of understanding the AQG as the inverse task of question answering, our system supports creative and critical thinking by asking the author of the paper for argumentation of their decisions and makes them consider alternative solutions. In addition, we support questions needing clarifications, as often the authors forget to sufficiently present the background known to them but not to scientific audience they are addressing.

AQG can be seen as a two-phase process, where a sentence selection step is followed by a question formulation phase. Question generating methods can use syntax-based, semantic-based, and template-based approaches (Kurdi et al., 2020), and recently, as in the other fields of NLP, neural methods (Pan et al., 2019).

This paper develops upon our initial RoboCHAIR system, described in Pollak et al. (2015). The RoboCHAIR creative assistant was developed originally to assist conference session chairs by generating relevant scientific questions during a conference. In addition, the system can be used to support the students when preparing their papers or reviews. Student assesment is also performed in the experimental setting of this paper. The new version of RoboCHAIR, which is the topic of this paper, integrates two different methods. Both are based on finding relevant source sentences in the text. The first one utilises an improved pattern-based approach, keeping only the best performing question categories, following the evaluation described in Pollak et al. (2015). The second approach has been newly developed and is based on templates incorporating triplet graphs and embeddings. The motivation for this method is to overcome the drawback of

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5

using the pattern-based method alone, which limits question generation to a sentence level, and does not consider the context of the entire article. While the first method is more focused on creative questions, the second method focuses on clarification types of questions. Note that at this stage, the second method has an advanced sentence selection step, but only a basic question generation part implemented.

Our system comes with a web interface. In this paper, we use the interface as an evaluation platform, where students upload the papers, and evaluate the generated questions. In this setting, the students can get feedback on the papers they wrote, where they use the RoboCHAIR system as an author assistant. On the other hand, they can upload the papers that they read. In this setting, the system can be seen as supporting them in their first reviews.

The paper extends the system by Pollak et al. (2015), which focused on the conference session chairs assistance. We propose a novel question generation method and by a novel evaluation in which we compare the two question generation methods. The paper is organised as follows. We first present the platform functionalities, followed by the description of the RoboCHAIR question generation methods, focusing on the triplet embedding method, the evaluations and a presentation of conclusion and future work plans.

Platform Functionality

RoboCHAIR, available at `http://robochair.ijs.si` integrates two main modules:

Conference scheduling assistant integrates the system by Škvorc, Lavrač, and Robnik-Šikonja (2016) into the RoboCHAIR platform and is designed to help conference chairs identify groups of similar papers using clustering, and to assign papers to predefined time slots.

Question generation assistant integrates the pattern-based question generation to enhance creative process and graph-based question generation to identify sentences needing clarifications. The question generation module can be used for several purposes: *Session chair assistant mode* is designed for the conference use and assists conference session chairs. In this mode, the questions (both automatically generated and posed by the audience) are ranked based on the audience evaluation. *Author assistant mode* assists authors before submitting papers to a conference or when preparing a conference presentation: the author is thus exposed to questions that she/he could get from the peers. The generated questions are evaluated by individual researchers. In *Reviewer assistant mode* the aim is to assist young researchers in the reviewing phase, generating the questions for the papers that they did not author (same as for the Session chair assistant mode).

The question generation module (`http://kt-robochair.ijs.si`) consists of the following functionalities:

Uploading files in three different formats (.pdf, .tex, .txt) and preprocessing to improve PDF-to-text and TeX-to-text conversions.

Question generation selection allows user to choose the faster, pattern-based question generation, or graph-based clarification questions.

Question rating is used after the questions are generated, for rating and ranking (the user interface is shown in Figure 1). The questions are rated using two criteria: acceptability/understandability and relevance/meaningfulness.

Question editing allows users to correct the questions, which can serve for future improvements of the system.

Question commenting enables feedback for specific questions.

Suggesting new questions - the user can suggest their own questions or enter questions received from reviewers or from the conference audience. The question could be used as positive training examples in future.

Information about the paper The user is asked about paper authorship and for permission that the questions become public. Uploading the paper as "public" is obligatory in the *Session chair assistant* mode, in which the conference audience can rate the paper.

General comments about the system are invited.

In our paper, we focus only on the Question generation assistant, and not on the conference scheduling assistant. The evaluation is done by students in an offline setting, corresponding to the *Author assistant* and *Reviewer assistant* modes.

RoboCHAIR Question Generation

In this section, we present a module for question generation from scientific papers. It implements two methods:

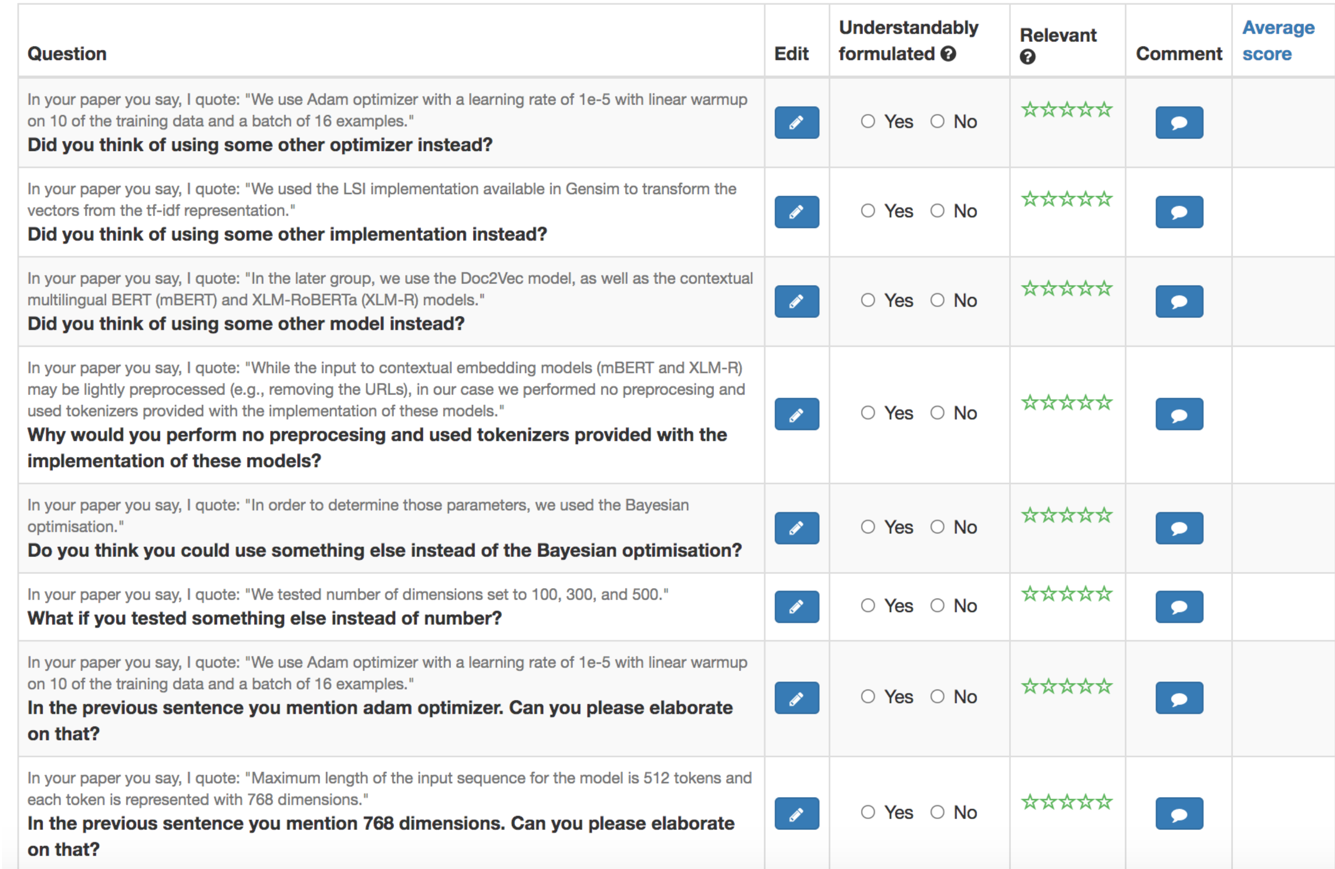
- A pattern-based approach to find relevant sentences, followed by a template-based natural language generation mechanism.
- A novel embedding-based method for sentence selection which identifies sentences containing candidate words that could be used to construct relevant questions.

A pattern-based method

This module implements a selection of the best templates from the system described in Pollak et al. (2015). Below, we briefly summarise the method.

Sentence selection This process first defines the list of verb forms (based on linguistic anchors), synonyms and conjugate catchword expansions. It proceeds with sentence matching and discarding incorrectly formed sentences. From the categories of the original system, we keep the categories that had high scores in the RoboCHAIR initial evaluation. The selected categories (with up to five example verbs functioning as linguistic anchors) are:

- Divide (differentiate, divide, exclude, isolate, limit)
- Focus (concentrate)
- Certainty (acknowledge, ascertain, certify, check, clarify)
- Usage (choose, investigate, try, use)



Question	Edit	Understandably formulated	Relevant	Comment	Average score
In your paper you say, I quote: "We use Adam optimizer with a learning rate of 1e-5 with linear warmup on 10 of the training data and a batch of 16 examples." **Did you think of using some other optimizer instead?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "We used the LSI implementation available in Gensim to transform the vectors from the tf-idf representation." **Did you think of using some other implementation instead?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "In the later group, we use the Doc2Vec model, as well as the contextual multilingual BERT (mBERT) and XLM-RoBERTa (XLM-R) models." **Did you think of using some other model instead?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "While the input to contextual embedding models (mBERT and XLM-R) may be lightly preprocessed (e.g., removing the URLs), in our case we performed no preprocesing and used tokenizers provided with the implementation of these models." **Why would you perform no preprocesing and used tokenizers provided with the implementation of these models?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "In order to determine those parameters, we used the Bayesian optimisation." **Do you think you could use something else instead of the Bayesian optimisation?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "We tested number of dimensions set to 100, 300, and 500." **What if you tested something else instead of number?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "We use Adam optimizer with a learning rate of 1e-5 with linear warmup on 10 of the training data and a batch of 16 examples." **In the previous sentence you mention adam optimizer. Can you please elaborate on that?**		○ Yes ○ No	☆☆☆☆☆		
In your paper you say, I quote: "Maximum length of the input sequence for the model is 512 tokens and each token is represented with 768 dimensions." **In the previous sentence you mention 768 dimensions. Can you please elaborate on that?**		○ Yes ○ No	☆☆☆☆☆		

Figure 1: A screenshot of the evaluation platform showing a subset of questions generated for the paper by Pranjić et al. (2020). Note that the first six questions on the picture are generated by the pattern-based methods, and the last two by the TEM method.

- Academic (accept, achieve, acquire, adopt, advance)
- Likelihood (assume, appear, occur, believe, consider)
- Improve (contribute, facilitate, improve, increase, reinforce)

In the new RoboCHAIR system we omit the following categories that had low scores in user evaluation of the original RoboCHAIR system: *Speech act*, *Attempt* and *Construct*.

With the category and the corresponding verb linguistic anchors, also the information about the verb WordNet synset is provided, by which all the WordNet synonyms are then automatically extracted (for example for verb *use*, the WordNet synset synonyms include `[apply, employ, use, utilise, utilize]`. Next, the code corresponding the Penn Treebank II POS tags (Bies et al., 1995) for verbs is associated to each anchor, where e.g., `D` is used for past tense in `VBD`, `G` for gerund of `VBG`, `N` for past participle of `VBN`, and `P` for `VBP` non-3rd person singular present. These tenses are then generated for each verb using Rita system (Howe, 2009).

In addition, one can determine if only active only passive or active and passive voice are considered for a specific template. Next, the approach automatically supplies pronoun catchwords to every verb catchword from the expanded list. Following simple heuristic rules, the pronouns `I` and `we` are used for active voice detection, and the pronoun `it` in passive voice forms (as well as some special cases of active voice and "dummy it", e.g., `it seems`). The quality of our candidate selection process is further stabilized by specifying additional white list or stop list elements in combination with specific anchors (e.g., for linguistic anchor `use`, the stop word filter `to` excludes the sentences of form " *we used to*" and differentiates them from e.g., "*we used [this method]*". In addition to these specific filters, also a general stop list is applied: as an example, sentences containing word `because` or starting with `why` are excluded, as we suppose that they already contain argumentation, or be questions themselves.

After defining these two- or three-word long catchphrases, a simple standard regular expression matcher is used and the sentences that were identified follow the syntactic analysis step,using the Stanford CoreNLP POS tagger and syntax parser (Manning et al., 2014). We then use the parse tree of a candidate sentence to find a pronoun from our list of catchphrases (e.g., `we` from `we show that`). For every match, we identify the enclosing noun phrase within the tree, and continue searching for the first verb phrase following. If it is found, we enter the phrase and search within it for the verb token from our catchphrase, e.g., `show`. If the verb is found, we test also for the optional third word of the catchphrase, in our example `that`. We mark everything within the subtree currently under consideration that follows the catchphrase as a so-called `object X` and use it later in the question generation phrase.

Let us have a look at the following sentence (from our initial RoboCHAIR paper (Pollak et al., 2015)): *"In a similar way, we used ConceptNet to find theme words by inspecting all the IsA relations in its database, from which it identified 11,000 themes."*. The catch phrase is formed by the pronoun *we* and the verb *use* in the VBD (Verb, past tense) form. The sentence also does not match the criterion for stopping: the catch phrase is followed by a noun phrase *ConceptNet* and not by a template-specific stop word *to* (to exclude the phrases of a type (*used to*). In this sentence, *ConceptNet* corresponds to *object X*, which is used in the question generation phase.

Question generation For each category, a corresponding template is activated. For the example above, the template is *"What if you $VBD something else instead of $X?"*

Variable `$VBD` is replaced by a past-tense form of the verb from the formula that was used when finding a pattern match (in our case *used*), and variable `$X` is replaced by the whole object X, as extracted from the candidate sentence parse tree in the selection phase (in our case *ConceptNet*). If object X exists in a domain specific ontology, its hypernym is used instead. Instead of a single template, several templates are proposed to improve the diversity for each pattern.

Below, we provide few examples from the evaluation presented in this study, together with the linguistic category of the anchor. We list the input sentence form the sentence selection phase, and the resulting generated question.

- **Divide**
  Sentence: *For sequences longer than 512 tokens after tokenization, we took 256 tokens from the beginning of the text and 256 tokens from the end of the text and concatenated them together.*
  Question: *Why did you decide to take 256 tokens from the beginning of the text and 256 tokens from the end of the text and concatenated them together?*[1]
- **Focus**
  Sentence: *We focus on the hate speech recognition task.*
  Question: *Why did you decide to focus on the hate speech recognition task?*
- **Usage**
  Sentence: We use sentence-level rewards to optimize the extractor while keeping our ML trained abstractor decoder [fi][2]xed, so as to achieve the best of both worlds.
  Question: *Do you think you could use something else instead of sentence-level rewards?*

[1]Note that the AUQ results can contain some grammtical errors. For example, in the coordinated sentence above, the result of the question generation step is not fully correct; the transformation of the verb past form is not performed on the second part of the coordinated sentence (... *and concatenated them together*). In future, additional rules for coordinated sentences should be introduced.

[2]fi was omitted in the pdf to text conversion

Other examples of sentences can be seen in Figure 1. The first eight sentences, generated based on the input paper by Pranjić et al. (2020), are presented (the file was uploaded as a .tex file). The first six questions on the picture are the result of the pattern-based method. We can see that the category *usage* was the source of the majority of the questions on the picture.

A triplet graph embedding method (TEM)

In order to overcome the limitations of the pattern-based sentence selection we have developed an advanced approach which tries to mimic human understanding and sentence selection for question generation. The process of question generation in humans is based on deep understanding of the text while adding all the available background knowledge of the reader. The content that cannot be sufficiently explained using this procedure requires additional information which can be obtained by formulating questions about the relevant parts of the input.

Our approach approximates this process by extracting the essence of the text in the form of a triplet graph which is followed by inserting additional edges between triplet parts using similarity queries based on word embeddings. Our assumption is that in the end the nodes with very low number of outgoing edges are the ones that require additional explanation because they are not sufficiently interlinked in the triplet knowledge graph. In comparison with human question generation, triplet extraction mimics extracting the content while word embeddings plays the role of the background knowledge database which connects entities into a coherent picture. In the following we provide a detailed description of the proposed method.

Triplet extraction is a common way for information extraction from unstructured text data. A triplet consists of *subject*, *predicate* and *object* and defines a binary relationship between the subject and the object. Given a chunk of text one can extract triplets and construct a triplet graph which is a visual summary of the text. On the other hand, word embedding enables the mapping from words to vectors of real numbers which allows for various computations such as distance (similarity). Using word embedding, "nearest neighbours" of words can be easily obtained.

The TEM method combines the two methods by taking the triplet graph and computing additional edges using word embedding. Taking into account that word embeddings can be trained on specific domains, this approach mimics the use of human background knowledge during the cognitive process when the recognised entities are grounded. Given some input text, a trained word embedding and a triplet extractor, the algorithm works as follows:

1. extract triplets from the document;
2. create a triplet graph (a directed multigraph) by adding subjects and objects as nodes and predicates as directed edges connecting subjects with objects;

Figure 2: A triplet graph with several additional word embedding based edges. Edges between triplet parts are in black while word embedding edges are in gray. Terminal nodes without outgoing edges are shown in red colour and represent good candidates for question generation.

3. for each node, find n nearest neighbours[3]
4. scan all nodes and insert directed edges if a nearest neighbour of some node is contained in another node;
5. the result is a set of nodes $\mathcal{A} = \{a \mid m \leq deg^{+}(a) \leq M\}$ for some user-defined thresholds m and M.

The algorithm return a set of nodes (triplet subjects and objects) which corresponding source sentences are candidates for question generation. There are three parameters which affect the size and quality of the result:

1. number of nearest neighbours: n
2. lower bound for node outdegree: m
3. upper bound for node outdegree: M

The number of nearest neighbours is related to the size of the triplet graph. When the graph is large, the number of neighbours should also be large. The key observation when adjusting n is that if it is too high, the graph will be overconnected and no node will have a low outdegree. On the other hand, if the number of neighbours is too low, unconnected components may appear and give a false impression of importance of the corresponding nodes. As a rule of thumb, the number of neighbours is proportional to the size of the input until some upper limit. For example, for a text consisting of one or two paragraphs, $n \in [1, \ldots, 10]$ is sufficient, while for a text consisting of several pages $n \in [100, \ldots, 1000]$ is an appropriate choice.

The lower and upper bound parameters m and M are only used during the selection of the result set and have no effect on graph construction. In general, when the number of neighbours is within reasonable limits, the nodes with zero or one outdegree are true outliers. For example, misspellings, names, abbreviation, foreign words etc. are often found in such nodes. Therefore, it is recommended to set m to some low number and increase it only if true outliers still appear in the result. The upper bound M is used to limit the number of results but can be determined automatically by gradually increasing it, starting with $m + 1$. This way, it is possible to return the desired number of results.

We illustrate the TEM method in the following example.

Example. Suppose we have the following text (an excerpt from our initial paper on conference management assistant (Pollak et al., 2015)) for which we want to identify parts that are considered relevant for asking questions:

[3] As a similarity function, cosine similarity was used. Note that for nodes containing more than one word, the node embedding is represented as a normalised sum of of the vectors of its constituents.

```
Question generation module
Our question generation system (depicted
in Figure 1) is composed of the source
sentence selection, i.e.  detection
of sentences in the article that will
be used for generating the questions,
and question formulation.  The system's
input are preprocessed text documents,
which are uploaded and converted into
raw text on the online platform (see
Section 3).
Pattern-Based Selection of Source
Sentences
We begin with a list of linguistic
anchors, i.e.  a database of
catchwords that enables us to select
candidate sentences as a source for
question generation.  The sentence
matching process has two phases:  the
coarse-grained and the fine-grained
sentence selection process.  We decided
to use relatively strong conditions for
selecting candidate sentences, since
we believe that - in order to achieve
higher quality - it is better to miss
some good candidates in the process
of selection than to generate too many
non-relevant questions.
```

Using the ReVerb triplet extractor (Fader, Soderland, and Etzioni, 2011) and GloVe word embeddings (Pennington, Socher, and Manning, 2014) we obtain a graph which is shown in Figure 2. The graph was constructed by representing each triplet `<subject, predicate, object>` as a set of nodes connected with a directed, labelled edge. In addition, 5 nearest neighbours of each node obtained by the GloVe embeddings trained on Wikipedia were considered for adding additional edges. The lower bound threshold m was set to 0 and the number of desired results was set to 3 (this turned on the auto tuning of the upper bound threshold M). The nodes in the result set were labelled as terminal nodes and coloured in red. We conclude that the phrases or terms in terminal nodes are not sufficiently explained in the text and should be considered while formulating questions. For example, the graph in Figure 2 can identify the following nodes, which could inspire the following questions:

- *two phase*: Please explain the two phases of the sentence matching process.
- *some good candidates*: Can you give some examples of good candidates?
- *many non-relevant questions*: How do the generated non-relevant questions look like in general?

The questions are highly relevant to the corresponding text and clearly demonstrate that the proposed method is able to identify weak points in its input. However, in the current implementation, the generation uses a simple generator template "Can you elaborate on X", where X is the identified triplet graph node.

In Figure 1, the last two sentences are the output of the TEM module.

Evaluation

The two methods were evaluated by six computer science students, who evaluated 11 papers in total. They were asked to select two papers, one that they authored and one that they read. The first setting corresponds to the *Author assistant mode* and the second one to the *Conference chair assistant* functionality. When uploading the papers, they were asked to use the default setting (both generation methods) and not only the faster pattern-based generation option. They were also asked to indicate whether they are evaluating the questions for the article they wrote (system used as *Author assistant*), or if they evaluate the paper that they did not author, (system used as *Reviewer assistant*).

The evaluation criteria were the same as in Pollak et al. (2015):

Understandability/Acceptability is a binary category verifying if the question was understandably formulated. The evaluators were asked not to penalize smaller mistakes (grammatical or PDF conversion errors), but to give negative answers if the question is not understandable.

Meaningfulness/Relevance is scored on a scale from 1 star (irrelevant) to 5 stars (very relevant), with the following description: 5=very relevant(meaningful, related to the topic, no semantic issues), 4=relevant (meaningful, minor semantic issues), 3=partly relevant (good but partly impertinent, some semantic issues), 2=not relevant (too trivial, big semantic issues), 1=completely irrelevant (not meaningful, wrong.

Regarding the selection of evaluation categories, our work was inspired by previous studies: the binary score evaluating if the question was *acceptable* (i.e. *understandably formulated*) can be related to the "acceptable vs. unacceptable" binary scores in Liu, Calvo, and Rus (2014); Chali and Hasan (2012). Next, our 5 star *meaningfulness/relevance* score can be aligned with the evaluation of (topic) relevance in (Chali and Hasan, 2012) but adapted, as in their study factual questions were generated.

The evaluation was focused on comparing the two question generation methods: the legacy pattern-based method which was already found to produce moderately relevant and understandably formulated questions and the new method based on embeddings and triplet graphs which employs a mechanism to identify under-explained parts but currently uses only one general template to construct questions.

In total, for the 11 scientific papers, the system generated 306 questions. 5.3% of the pattern-based approach[4] and 22.2% of TEM-based were not understandable. In total, 273 questions were rated as "understandable". Taking into account only those, the pattern-based system generated 196 questions where triplet embedding method (TEM) generated 77 questions.

The uploaded articles contained in average 624.18 sentences (the shortes article had 254 sentences and the longest

[4]Significantly better than in the initial RoboCHAIR system, where 13% of questions generated by the pattern-based approach were not understandable.

one 1572). The mean number of generated questions per paper was 24.8 and the standard deviation 14.6. The pattern-based method does not control the number of questions, as the number of resulting sentences depends on the sentence matching step. In contrast, TEM can return a desired number of questions. In order to balance the number of questions generated by both methods we configured TEM to match the number of questions produced by the pattern-based method if the number of pattern-based questions was between 5 and 10. Outside of this range the lower limit was set to 5 and the upper limit to 10.

The mean meaningfulness score for all understandable questions was 3.03, while if considering only the pattern-based method, the score was 3.17 and the mean score of the TEM method was 2.68. Compared to the initial pattern-based method from the first RoboCHAIR version, the score for the pattern-based approach increased from 2.99 to 3.17, and the overall score (including both methods) to 3.03. The TEM method average score (2.68) is also above the threshold of 2.5 which was selected for keeping the pattern-based question templates in the system. The distribution of the meaningfulness score for both methods is shown in Figure 3.

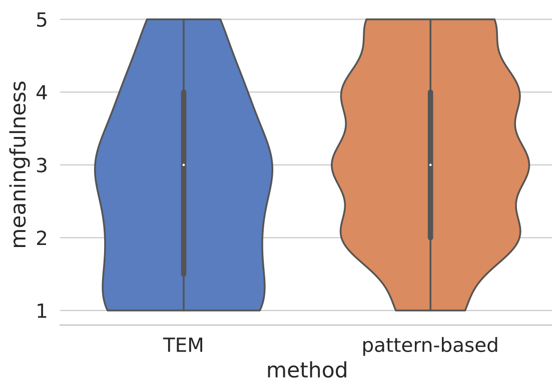


Figure 3: The distribution of the meaningfulness score of understandable questions for both question generation methods.

The TEM method received many lowest possible scores. An investigation revealed that the generated questions were ranked low because of the outliers that were identified as interesting nodes. This suggests that the default lower bound $m = 2$ is too low and the outlier nodes in the TEM graph are mistakenly identified as relevant. For example, names of people, methods, abbreviations, numbers, equation parts, etc. are targeted for question generation. This problem can be almost perfectly resolved by increasing the lower bound, improving text conversion, filtering triplets and a using domain-specific embedding model.

Taking into account the authorship of papers the meaningfulness score reveals a bias which is present in both methods but especially notable in the pattern-based method (see Figure 4). For evaluators who were not the authors the peak is close to 4 while for authors the peak is close to 3 which

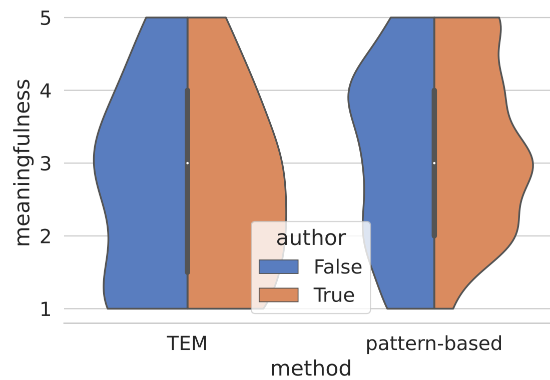


Figure 4: The distribution of the meaningfulness score for both question generation methods while taking into account whether the evaluator is also the author.

suggests that authors which have a detailed knowledge about their papers consider questions less relevant or possibly too simple.

While in the current implementation the TEM method is outperformed by the pattern-based one, we believe that the TEM method still has a lot of potential for improvement. Currently, the method focuses only on the sentence selection phase, but not on the question generation one, which is in our opinion one of the reasons of the lower results. This might be the source of a positive bias towards questions generated with the pattern-based method which uses more elaborate templates to generate questions. In few cases, both methods selected the same sentence as relevant and formulated a question. However, the question generated using the templates from the pattern-based approach was evaluated much higher than the question generated with the TEM method, which uses one general template. For example, based on the same input sentence, both methods generated the questions, which were ranked as 4 for the pattern-based method and 1 for TEM:

Sentence: We took 1,430 tweets labeled as the hate speech and randomly sampled 3,670 tweets from the remaining 23,353 tweets.

Pattern-based question (score: 4): Why did you decide to take 1,430 tweets labeled as the hate speech and randomly sampled 3,670 tweets from the remaining 23,353 tweets?

TEM question (score: 1): In the previous sentence you mention 1,430 tweets. Can you please elaborate on that?

This indicates that more elaborate generation part following the sentence selection by TEM could improve the results.

The number of overlapping sentences selected by the two methods is very low and mostly coincidental because the pattern-based method selects sentences according to predefined patterns which are suitable for question generation while TEM selects sentences according the connectivity of nodes in the triplet-embedding graph.

In one of the comments, the evaluator also explains that while in an application oriented paper, the system achieved quite relevant questions, in an overview paper that he uploaded, the questions were not that relevant.

Conclusions and further Work

This paper describes the updated RoboCHAIR system. The paper was evaluated by the students. In contrast to the main RoboCHAIR functionality, where the system was designed to help conference chairs, the students are also one of the core target groups, and the sytem aims to support them in the process of scientific writing of papers or reviews. The questions are designed to model human intelligence by triggering new scientific ideas or making the authors explain the decisions behind their approach, using two different methods, one based on patterns and one on triplet graphs. The first method was evaluated with higher scores, but we believe that as the generation module was more developed in the pattern-based approach, the triplet graph based method could be further improved in the future.

For example, in TEM method multi-word phrases during neighbourhood search are currently decomposed into words and the corresponding vectors are added to get the final vector. A possible improvement would be to consider bigger units, e.g., named entities instead of single words. Yet another interesting addition would be to use sense embedding (Camacho-Collados and Pilehvar, 2018) instead of word embedding to account for different meanings of words, or consider mapping contextual embeddings to static graph nodes. Finally, the discovered nodes with low outdegrees contain only parts of sentences and the problem of formulating the actual question about the under-explained part uses very simple templates. In future the question generation part of TEM should be further improved.

In terms of evaluation, in future work it would be interesting to compare automatically generated questions to question generated by humans. Currently, the evaluation criteria are relatively general, and in future it would be interesting to introduce scores explicitly focused on the novelty of the questions, as well as to get feaeback more specific to the actual use. Last but not least, using the evaluations in an active learning setting, where the scores would inform a machine-learning model to identify relevant would be a valuable addition.

Acknowledgements

We acknowledge the project the financial support from the Slovenian Research Agency for research core funding for the programme Knowledge Technologies (No. P2-0103).

References

Becker, L.; Basu, S.; and Vanderwende, L. 2012. Mind the gap: Learning to choose gaps for question generation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, 742–751. Stroudsburg, PA, USA: Association for Computational Linguistics.

Bies, A.; Ferguson, M.; Katz, K.; and MacIntyre, R. 1995. *Bracketing Guidelines for Treebank II Style Penn Treebank Project*.

Boden, M. 2004. *The Creative Mind: Myths and Mechanisms*. London: Weidenfeld and Nicholson.

Brill, G., and Yarden, A. 2003. Learning biology through research papers: a stimulus for question-asking by high-school students. *Cell biology education* 2:266–74.

Camacho-Collados, J., and Pilehvar, M. T. 2018. From word to sense embeddings: A survey on vector representations of meaning. *CoRR* abs/1805.04032.

Chali, Y., and Hasan, S. A. 2012. Towards automatic topical question generation. In *Proceedings of International Conference on Computational Linguistics (COLING 2012)*, 475–492.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of 20th European Conference on Artificial Intelligence (ECAI)*, 21–26. Amsterdam: IOS Press.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*.

Geelan, D. R. 1997. Epistemological anarchy and the many forms of constructivism. *Science & education* 6(1):15–28.

Hazeyama, A., and Hirai, Y. 2007. Concerto ii: A learning community support system based on question-posing. In *Seventh IEEE Int. Conf. on Advanced Learning Technologies (ICALT 2007)*, 338–339.

Heilman, M., and Smith, N. A. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, 609–617. Stroudsburg, PA, USA: Association for Computational Linguistics.

Howe, D. C. 2009. Rita: Creativity support for computational literature. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, 205–210.

Kalady, S.; Elikkottil, A.; and Das, R. 2010. Natural language question generation using syntax and keywords. In *Proc. Third Workshop on Question Generation, The Tenth International Conference on Intelligent Tutoring Systems (ITS'2010)*, 1–10.

Khodeir, N.; Wanas, N.; Darwish, N.; and Hegazy, N. 2014. Bayesian based adaptive question generation technique. *Journal of Electrical Systems and Information Technology* 1:10–16.

Kurdi, G.; Leo, J.; Parsia, B.; Sattler, U.; and Al-Emari, S. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education* 30(1):121–204.

Liu, M.; Calvo, R. A.; and Rus, V. 2014. Automatic generation and ranking of questions for critical review. *Journal of Educational Technology & Society* 17(2):333–346.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 55–60.

O'Donoghue, D. P.; Abgaz, Y.; Hurley, D.; Ronzano, F.; and Saggion, H. 2015. Stimulating and simulating creativity with dr inventor. In *Proceedings of the Sixth International conference in Computational Creativity*, 220–227.

Pan, L.; Lei, W.; Chua, T.-S.; and Kan, M.-Y. 2019. Recent advances in neural question generation.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Piwek, P., and Stoyanchev, S. 2010. Question generation in the coda project. In *Proc. Third Workshop on Question Generation, The Tenth International Conference on Intelligent Tutoring Systems (ITS'2010)*, 29—34.

Pollak, S.; Lesjak, B.; Kranjc, J.; Podpečan, V.; Lavrač, N.; et al. 2015. RoboChair: Creative assistant for question generation and ranking. In *2015 IEEE Symposium Series on Computational Intelligence*, 1468–1475. IEEE.

Pranjić, M.; Podpečan, V.; Robnik-Šikonja, M.; and Pollak, S. 2020. Evaluation of related news recommendations using document similarity methods. In *Proceedings of the Conference on Language Technologies & Digital Humanities 2020*, 81–86.

Rus, V.; Piwek, P.; Stoyanchev, S.; Wyse, B.; Lintean, M.; and Moldovan, C. 2011. Question generation shared task and evaluation challenge: Status report. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, 318–320. Stroudsburg, PA, USA: Association for Computational Linguistics.

Steffe, L. P. 1991. The constructivist teaching experiment: Illustrations and implications. In von Glasersfeld, E., ed., *Radical constructivism in mathematics education*. Kluwer, Dordrecht. 177–194.

Sullins, J.; Graesser, A.; Tran, K.; S.Ewing; and Velaga, N. 2010. The effects of cognitive disequilibrium on question generation. In *Proc. Third Workshop on Question Generation, The Tenth International Conference on Intelligent Tutoring Systems (ITS'2010)*, 21—-28.

Vale, R. D. 2013. The value of asking questions. *Molecular biology of the cell* 24:680–2.

Škvorc, T.; Lavrač, N.; and Robnik-Šikonja, M. 2016. Co-bidding graphs for constrained paper clustering. In Mernik, M.; Leal, J. P.; and Oliveira, H. G., eds., *5th Symposium on Languages, Applications and Technologies, SLATE 2016, June 20-21, 2016, Maribor, Slovenia*, volume 51 of *OASICS*, 1:1–1:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Wang, S.; Wei, Z.; Fan, Z.; Huang, Z.; Sun, W.; Zhang, Q.; and Huang, X. 2020. PathQG: Neural question generation from facts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 9066–9075. Online: Association for Computational Linguistics.

Wilson, E. V. 2004. Examnet asynchronous learning network: Augmenting face-to-face courses with student-developed exam questions. *Computers & Education* 42(1):87–107.

Yu, F.-Y., and Liu, Y.-H. 2009. Creating a psychologically safe online space for a student-generated questions learning activity via different identity revelation modes. *British Journal of Educational Technology* 40(6):1109–1123.

Yu, F.-Y. 2009. Scaffolding student-generated questions: Design and development of a customizable online learning system. *Computers in Human Behavior* 25(5):1129–1138.

Proceedings of the 12th International Conference on Computational Creativity (ICCC '21)
ISBN: 978-989-54160-3-5