2015 IEEE Symposium Series on Computational Intelligence

RoboCHAIR: Creative assistant for question generation and ranking

Senja Pollak, Borut Lesjak, Janez Kranjc, Vid Podpečan, Martin Žnidaršič, Nada Lavrač
Jožef Stefan Institute and Jožef Stefan Postgraduate School
Jamova 39, 1000 Ljubljana, Slovenia
Contact email: senja.pollak@ijs.si

***Abstract*—Computational Creativity is a subfield of Artificial Intelligence research, studying how to engineer software that exhibits behaviours which would reasonably be deemed creative. This paper addresses a creative task of question generation from scientific papers, using a pattern-based approach to finding relevant sentences from which questions should be generated, a natural language processing question construction mechanism, a crowdsourcing mechanism for question rating, and a robot interface for posing questions during a conference session, integrated in a creative RoboCHAIR solution. The system was trained on a set of 200 articles from past computer science conferences and evaluated on a set of articles of members of the local lab.**

I. Introduction

This paper addresses a creative task of question generation from scientific papers. On the one hand, creative question generation is a computational creativity task, given that Computational Creativity [1], [2]—as a subfield of Artificial Intelligence research—is concerned with engineering software that exhibits behaviours which would reasonably be deemed creative. On the other hand, from a technical perspective, the main underlying technology is automated question generation, belonging to fields of Artificial Intelligence and Computational Linguistics, addressing also disciplines such as Psychology and Formal Logic in a supporting role [3]. Despite the fact that automation of scientific research is an important trend in last decades, covering various fields from literature review aiding systems [4] to robotic hypothesis generation and experimentation [5], automated question generation systems are generally not conceived to be used by scientists or (post)graduate students. The exception is a system offering help to students in performing critical literature reviews, based on automated question generation [6]

Automated question generation (AQG) is a task of automatically generating questions from some form of input [7], [3], where the input can vary from information in a database to pure text (e.g. tasks defined at sentence or paragraph level), deep semantic representation or queries, etc. [8]. AQG technologies can be used in question-answering (e.g. [9]), dialogue systems (e.g. [10]), educational applications or intelligent tutoring systems (e.g. [11], [12]). Numerous projects have focused on design of web-based systems for student question generation (e.g. [13], [14], [15]). From a constructivist perspective, where learning is mainly considered as engagement of students in meaningful and understandable learning tasks about which they can reflect abstractly, systems that support student question generation are considered, since they support individuals' understanding and cognitive development, as well as direct experience and creative manipulation of information [16], [17], [13], [18].

This paper addresses a creative task of question generation from scientific papers, using an advanced pattern-based approach to finding relevant sentences from which questions should be generated, a natural language processing question construction mechanism, a crowdsourcing mechanism for question rating, and a robot interface for posing questions during a conference session, integrated in a creative RoboCHAIR solution. The resulting RoboCHAIR system in an online conference support system, whose main underlying functionality is automated question generation from scientific articles.

The motivation for this development is to assist a conference session chair in posing relevant questions to researchers that present their papers at conferences. Given that scientists submit their papers to conferences, and that accepted conference papers are organized in sessions, the session chair has to moderate the question-answering debate and frequently needs to pose interesting questions to the presenting researcher. Posing an interesting and relevant question is a proof of human intelligence. In this paper, human-like performance is intended to be achieved through automated question generation from input texts and basic domain knowledge. The developed RoboCHAIR system, which is available online, can be used for (a) generating questions, (b) gathering evaluations of generated questions for papers which are uploaded in the system and declared as "public" by a session chair, (c) gathering question evaluations for the papers uploaded by the audience and declared as "public" by the paper author, (d) gathering user generated questions and (e) editing and improving the automatically generated questions. The RoboCHAIR system has been designed to be used in two main modes:

AUTHOR ASSISTANT mode is designed to be used by the paper author before submitting the paper to a conference or when preparing a conference presentation. In this mode, the author is exposed to questions that he could get from a paper reviewer, a session chair or the conference audience. The generated questions are evaluated by the individual researcher. The evaluation of the RoboCHAIR system, presented in Section V, is done using this system modality.

SESSION CHAIR ASSISTANT mode is designed for conference use to assist the conference session chair. In this mode, the audience evaluates the questions generated by RoboCHAIR for the currently presented paper. During paper presentation,

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.208

the audience can upload also their own questions. Finally, the crowdsourcing interface is used for gathering audience evaluations and for question ranking based on audience ratings. When used in a conference session, the system might stimulate greater audience participation, since questions can be posed anonymously. In addition—when available—a robot interface can be used to present the top ranked questions (see Section IV for details).

The RoboCHAIR system could be extended for use in academic educational scenarios (or integrated in tutoring systems) aimed at developing critical thinking about what we write or read. Asking questions is a method by which assessment and enhancing learners' engagement [19] can be achieved. In that line, our system could be used to trigger critical thinking by generating questions that can be edited, inspiring students to generate their own questions or by asking them to provide answers to selected questions.

The paper is structured as follows. The sentence identification and question generation module is presented in Section II. The RoboCHAIR platform including the crowdsourcing interface for question evaluation is described in Section III, while the robot interface is covered in Section IV. Evaluation results are reported in Section V followed by data-mining experiments reported in Section VI. The related work is outlined in Section VII, followed by the conclusions and plans for future work.

II. 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 III).

A. 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.

1) Defining Linguistic Anchors: First, a set of verbs expressing decisions is defined (precisely the verbs *use*, *choose* and *decide*). This is motivated by searching for expressions by which researchers explain their decisions made in their research process and questioning them on other decision alternatives. Next, we added a list of verbs from Biber [20], namely the *certainty*, *likelihood*, and *speech act* categories. This choice is motivated by searching for presuppositions and claims that lack (sufficient) argumentation and could be questioned, further explained or referenced in the article. We further expanded the list of catchwords with additional verbs from two other sources: reporting verbs of three levels, neutral, tentative, and strong [21]; and verbs from Paquots Academic Keyword List [22], which is the most general. The verbs from various resources were classified into ten categories and serve as a basis for the coarse-grained sentence selection process. The ten categories are listed in Table I, ranked in terms of their usefulness for relevant question generation.

2) Coarse-Grained Sentence Selection: This process consists of defining the list of verb forms (based on linguistic anchors), synonym and conjugate catchword expansion and sentence matching, as well as discarding incorrectly formed sentences.

After splitting the input document into sentences by tokenizer and sentence splitter of the Stanford CoreNLP [1] module, we discard the incorrectly written sentences, i.e.: all sentences that do not start with a capital letter or end with a full stop or an exclamation mark, and all sentences that contain any ASCII control character.

The next step is matching the words in the candidate input sentence with our catchwords and automatically acquired synonyms. We have assembled a sizable list of verbs, as described in the previous subsection, that we believe, if used in a sentence, will make a good candidate to ask a question about. Each item on this list of verbs is described in the following format:

use:81161188:DGNP:ap

The number 81161188 is a WordNet [23] sense ID, as we use WordNet with the RiTa toolkit [24] to automatically expand our list of verbs with their synsets, i.e. the narrowest set of synonyms with that same sense. In this example, the corresponding synset is [utilise, employ, use, apply, utilize].

The code "DGNP" is simply a set of trailing letters of the Penn Treebank II [25] POS tags for verbs: VBD, VBG, VBN, and VBP, respectively. We use the RiTa [24] toolkit to conjugate the expanded list of verbs and obtain only those verb forms we deem appropriate. In the above example, for the verb *use* we generate forms *used, using, used*, and *use*. The code "ap" denotes that both active and passive voice are accepted.

Our two-level synonym and conjugate catchword expansion increased the number of catchwords from the original 177 different verb forms to 1,331 different verb forms.

Next, we raise the quality by using pronoun/verb pairs of catchwords instead of simple words. Our approach automatically supplies pronoun catchwords to every verb catchword from our expanded list. Following simple heuristic rules, we introduce pronouns "I" and "we" for active voice detection, and allow also the pronoun "it" in passive voice forms (as well as some special cases of active voice, e.g. "it seems").

We further stabilize the quality of our candidate selection process by adding a so-called phrase catchword parameter for selected verbs of our list, since we observed that certain verbs yield best candidates especially or only when followed directly by a certain word, e.g. "that", as in the phrase "we show that", or when not followed by a certain word, e.g. "to" in "we used to", which obviously carries a completely different meaning from "we used [this method]".

The final step of the coarse-grained selection phase is using a simple standard regular expression matcher to check our two- or three-word long catchphrases against the sentence candidate

under scrutiny. If there is a match, the candidate is progressed into the next, fine-grained phase of the selection process.

3) Fine-Grained Sentence Selection: In this phase we use the full Stanford CoreNLP POS tagger and syntax parser [26]. Even if the performance price is high with these complex tools, it becomes affordable since we only run them on a very small subset of pre-selected candidates. At first we used the default parser that comes with the package, but we changed for Shift-Reduce Constituency Parser model which performed up to 10 times faster than the default model.

We first discard all sentences containing words or phrases defined by a stop list, in order to further raise the probability of selecting a proper candidate for yielding a valid and relevant question. An example of such a filter we use, is the word "because", since we consider that this type of sentences already contain the argumentation and that therefore our question starting with "why" would not be relevant. For the moment the stoplist is quite short, but will be extended in future work.

We then use the parse tree of a candidate sentence and try 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. We skip any other phrases, such as adjective, adverb, or conjunction phrase, and focus solely on a verb phrase. If itis found, we enter the phrase and search within for the verb token from our catchphrase, e.g. "show". If the verb is not found in the exact form, we abandon the whole match and continue with the next catchphrase from our list, on the same candidate sentence. If the verb is found, we immediately test also for the optional third word of our catchphrase, in our example "that".

Everything within the subtree currently under consideration that follows our catchphrase is marked as so-called "object X", which is used later in the process of formulating the question. An example of a candidate sentence that was successfully matched with the catchphrase "we use" is:

"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."

B. Question Formulation

1) Question Generation Using Templates and Syntactic Trees: We use a template based question formulator, combined with the power of syntactic trees to help understand the underlying grammar structure of the selected sentence. Object X of a sentence is defined and bound by the syntactic rules that follow from the parse tree. In the example above, object X is simply "ConceptNet".

We have a pre-constructed list of templates corresponding to the list of catchphrases that we relied on in the selection phase. Both lists, together with the aforementioned filters, or stop lists, and some other fine-tuning parameters, are provided as an external input to our application and are thus completely separate from the core algorithm. This means we can easily adapt and improve the program without any code modification, simply at the level of text input. Moreover, we have an option of opening up this control even for the end user.

An example question formulation template is given here:

```
!usage "What if you $VBD something else
instead of $X?"
```

Command "!" and identifier "usage" start a new category of catch phrases, or formulas, that we use for pattern matching in the selection phase. The text enclosed in the double quotes is the template proper text. It is a syntactically well-formed question that may contain template variables. Template variables are replaced with actual words. Variable "$VBD" is replaced by a past-tense form of the verb from the formula that was used when finding a pattern match, in our case, word "used". And variable "$X" is replaced by the whole object X, as extracted from the candidate sentence in the selection phase, in our example, word "ConceptNet". Thus the formulated question becomes:

```
"What if you used something else
instead of ConceptNet?"
```

Instead of only one single template for a given category of pattern-matching formulas, we usually define a list of them to choose from. At the time of question formulation, one of the candidates is randomly chosen and applied. In this way, we achieve higher diversity of generated questions.

2) Head Noun Detection and Hypernym Finding: The last phase of question formulation adds domain knowledge and variety to the process. Going deeper into object X, we try to detect its head noun, or semantic head, by using [26].

In addition, we use a domain-specific list of keywords and ontologies that we assembled while processing the training corpus of 200 articles. For keyword extraction we selected a little over a hundred keywords from a list of automatically extracted terms from the training corpus [27] by comparing relative frequencies of words in the domain corpus compared to the BNC reference corpus of English [28]. In addition we used also a proof-of-concept taxonomy, with ten ontology categories with just a few values each. In future work, this ontology might be replaced by automatically induced domain taxonomies (cf. [29]). By using keywords and ontologies we were able to mimic background knowledge. To work with our domain-specific keywords and ontologies, we introduced a new type of question template. Here is an example:

```
&~"Have you considered $VBG some other
~X instead?"
```

Command "&~" is different from the above mentioned "!", denoting that we use keywords and ontologies. The verb form template variables are the same. And the new template variable "~X" is a plug that gets replaced by new text.

If a taxonomy entry that matches the extracted head noun is found then the ~X variable gets replaced by its hypernym. If not, then the list of keywords is tried, and if a match is found, the ~X is replaced by the keyword. If neither a match for an ontology entry nor a match for a keyword is found, we fall back to using the original generic question template with the whole object X. Since, in our example above, the head noun ConceptNet is a match to our ontology entry with a hypernym "resource", the following question gets formulated:

```
"Have you considered using some other
resource instead?"
```

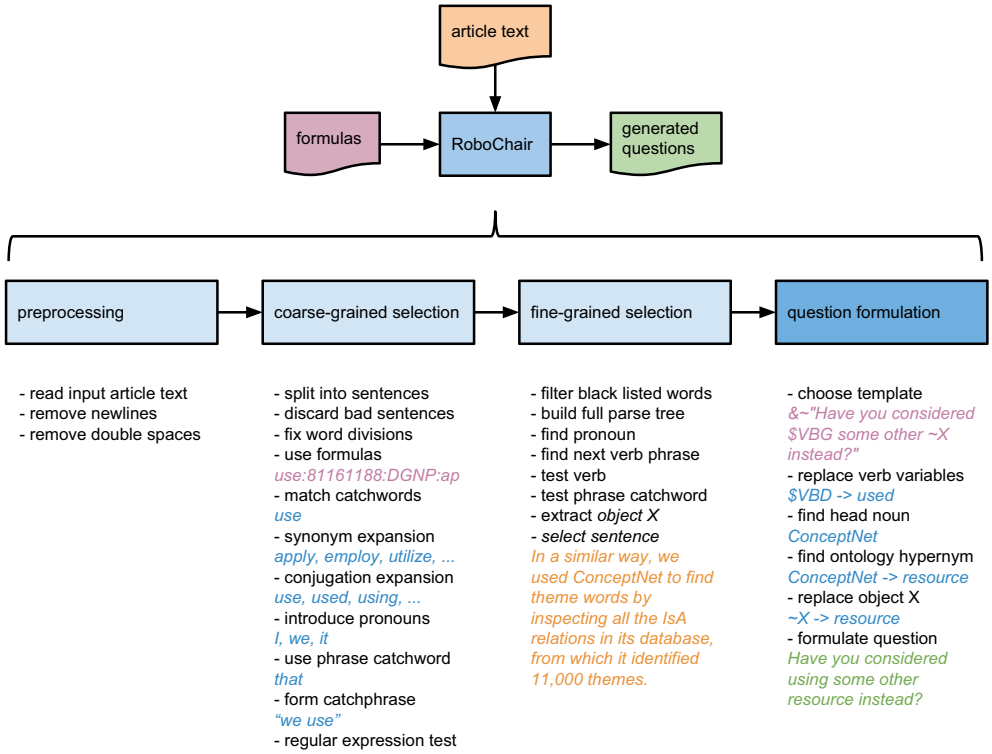

Fig. 1. Question generation methodology overview.

III. The RoboChair Question Generation Platform with a Crowdsourcing Interface

We have developed an online platform that generates questions based on the scientific papers that the users upload. The platform is web-based and is hosted at http://kt-robochair.ijs.si.

The main usage of the platform is as follows: the user uploads a scientific paper, waits for the system to generate questions, rates the questions based on predefined criteria, optionally edits and comments the questions, and finally formulated new questions for the particular paper. The resulting ratings of questions were used to generate the model described in Section VI.

A. Platform Architecture and Technologies Used

The crowdsourcing platform consists of two parts: the question generation module and the website that provides the database and user interface.

The question generation module (described in Section II) is written in Java and was exposed to the user interface by exposing its main function as an internal REST API service with a single endpoint. This API endpoint receives the text of the paper as the input and returns the generated questions.

The website is built using the Django framework for the back-end and AngularJS for the user interface. The website stores all the questions generated by the question generation module and allows users to rate them.

The platform also exposes its functionalities as an external API which can be used to produce other user interfaces such as the robot interface described in Section IV.

B. Platform Functionality

Uploading files. Files can be uploaded in three different formats (.pdf, .tex, .txt). The files are processed in order to alleviate potential problems regarding the PDF to text and TeX to text transformations. New line characters are replaced by space characters, and then adjacent spaces, or double spaces, are condensed into single ones. Word divisions left from the

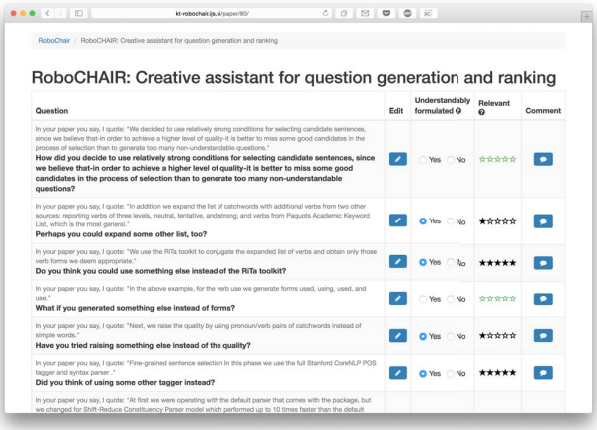

Fig. 2. Screenshot of the crowdsourcing platform opened in a web browser, showing questions generated for the given paper, together with the user's evaluations of generated questions. Available at http://kt-robochair.ijs.si

original PDF input document are removed by "- " string replacement.

Question rating. After the questions have been generated the user is prompted to rate them. The user interface for rating the questions is shown in Figure 2. The questions are rated using two criteria. The user decides whether the question is understandably formulated and determines the relevance of the question on a scale from 1 to 5. When clicking on the evaluation category, a more precise explanation is provided. For details regarding the evaluation measures, see Section V. The scores can be used for evaluation, system improvement, as well as for selecting the best questions in the SESSION CHAIR ASSISTANT mode.

Question editing: The generated questions can be edited by the users. These alterations are stored and could be used in future improvements of the system.

Question commenting: This enables getting feedback for specific questions (users can comment why a certain question is good/bad or provide suggestions for improvement).

Suggesting new questions: The user can either suggest his own questions or enter questions he has received from reviewers or from the conference audience. He can also tag the question accordingly. In a longer run, these questions can be used as positive examples for training.

Information about the paper: The user is asked to provide information whether he is the author of the paper or not and whether he gives his permission that the questions become public. Uploading the paper as "public" is also used in the SESSION CHAIR ASSISTANT mode, in which the conference audience can rate the same paper.

General comments. We gather also comments about the system not only about the specific questions.

IV. The Robot Interface

The robot interface of the RoboCHAIR system was implemented on the Aldebaran robotics' NAO platform. We used the Choregraphe software suite which enables the construction of workflows that can be deployed and executed on the robot. Two workflow solutions both of which use the REST JSON API provided by our CrowdSourcing web application were developed.[1]

The first one is the actual front-end of RoboCHAIR which can be used at public events. It is a simple one-way robot-human communication where the robot selects the top ranked questions provided by the API and asks them using its speech synthesis module. NAO's built-in animated speech feature can also be used to provide a more lively performance.

The second workflow is a crowdsourcing solution where the robot is used to collect the data from individual users (possibly anonymous) in order to obtain training examples from which the models for question ranking can be built. It is organised as an interview where the robot interviews the user about the relevance of the presented questions. This workflow is more complex and features several bidirectional robot-human interaction parts. Speech synthesis and speech recognition are used by the robot to present the questions to the user and recognize the answers. In addition, several features such as head tracking and blinking were added in order to provide a more pleasant user experience.

[1]Note that the robot interface was not used as an ingredient of the RoboCHAIR solution evaluated in Section V.

V. Evaluation

The evaluation was performed using the developed crowdsourcing platform. We performed two types of evaluation, first one aiming at evaluating the question generation system and the second one the inter-rater agreement. We also provide system efficiency results.

A. Evaluation Criteria

In our study (inspired by the related work), the evaluators were asked to evaluate the questions based on two criteria:

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

Relevance is scored on the scale from 1 star (irrelevant) to 5 stars (very relevant), where from 3 on questions are concerned as good questions. To avoid ambiguity, we provided also more detailed information:
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)

The users were allowed also to leave the category empty, since if they are not the authors of the paper, they cannot always adequately judge the relevance.

As will be further discussed in Section VII, the selection of evaluation categories was inspired by previous studies, where the most similar to our work is [6] where the authors are also interested in triggering questions for academic support. Their users evaluated the questions using the Likert scale (from 1 to 5) for five categories. Our relevance can be aligned with their scores for "usefulness" and "appropriateness to the context" and their binary criterion "acceptable vs. unacceptable" can be compared to our binary criterion "understandable/acceptable".

B. Evaluation for the Articles Uploaded by their Authors

The main evaluation addressed the members of our department, as well as the students participating in a recently organized student conference, who were asked to evaluate the papers they wrote (or reviewed). The results are shown in Table I. On average, 0.87 of sentences were judged acceptable and understandable, while the mean relevance score is 2.99.

Examples of questions with relevance score 5 (for an article on sentiment analysis of tweets):

- Have you tried using some other preprocessing instead?

TABLE I. EVALUATION BY UNDERSTANDABILITY (PROPORTION OF UNDERSTANDABLE QUESTIONS) AND RELEVANCE (ON THE SCALE FROM 1 TO 5 (TOTAL AND BY FORMULA CATEGORIES)

CATEG.	#QUEST.	UNDERST. (proport.)	RELEVANCE ($\pm$ st. dev.)
Divide	9	1.00	4.11 ($\pm$0.99)
Focus	2	1.00	4.00 ($\pm$1.00)
Certainty	13	1.00	3.62 ($\pm$1.08)
Usage	86	0.89	3.44 ($\pm$1.47)
Academic	73	0.84	2.77 ($\pm$1.37)
Likelihood	12	0.83	2.67 ($\pm$1.25)
Improve	16	0.87	2.53 ($\pm$1.31)
Speech act	12	0.83	2.42 ($\pm$1.19)
Attempt	4	1.00	2.25 ($\pm$0.83)
Construct	20	0.89	1.90 ($\pm$1.14)
TOTAL	247	0.87	2.99 ($\pm$1.44)

- Perhaps you could use some other lexicon instead?

and an example of the lowest scored questions using the same formula category:

- Have you tried making something else instead of use?

Our system is not directly comparable to other systems, since we address a highly creative task of proposing relevant questions to a scientific audience. However, we can list results of related research. The most similar is the work of [6] for AQG for creative related work understanding. We outperform their results on acceptability (0.65 without ranking, 0.76 on top 25% of ranked questions). Our score of relevance that is just below 3 can be compared to their criteria of whether the question is *useful* or *appropriate to the context*, which is on top 126 questions reported as 4.02 and 3.99, respectively. Scores without ranking are not reported. The comparison with factual questions is not very relevant, because the nature of the task is so different, 0.52 are marked acceptable in [30] but their criteria are slightly different (factual questions should not have any grammatical and semantic deficiencies from the categories of mistakes that they define). For topical factual questions [31] report a score of 2.15 without and 3.48 with ranking for *relevance* evaluated on the scale from 1 to 5.

C. Evaluation of Inter-Rater Agreement

For computing the inter-annotator agreement we took 10 papers of the ICCC 2015 International Conference on Computational Creativity. Automatically generated questions were evaluated by two reviewers each. For these papers the evaluators were familiar with the field but have not read the papers. The pairwise average agreement was 36.5, while the ordinal *Krippendorff's* α [32], caluculated by Freelon's calculator [33], is 0.509 (on the scale where 1.0 indicates perfect agreement, and alpha value of 0.0 a random agreement).

D. Efficiency Evaluation

The processing time for an average article is a few seconds, if the system is run as a service with all the sentence selection and question generation models pre-loaded. The complete corpus of around 200 articles (from ICCC-2014, DS-2014 and ECML PKDD-2010 conferences) we have used in the process of model construction, containing of 61,170 sentences, resulted in 2,917 questions generated in 71.9 seconds.

VI. MACHINE LEARNING EXPERIMENTS

From the data collected with the developed crowdsourcing platform we have constructed a dataset that consists of 352 instances (questions), which we have described by 74 descriptive attributes. The target attribute, *Relevance*, which has an almost uniform distribution tells the relevance of the question with respect to the paper on the $1, \ldots, 5$ scale. The distribution of values is as follows: $1:20\%, 2:21\%, 3:20\%, 4:21\%, 5:18\%$. Our aim was to evaluate whether a significantly better than random ranking of questions can be obtained by using machine learning methods.

Since the attributes describe mostly linguistic and structural features of sentences and many of them have a very large number of values, they were expected to carry very low or zero information about the target. Consequently, the possibility of obtaining a good ranking of questions seemed unlikely. This was confirmed both by attribute evaluation and model evaluation. The Weka machine learning software was used to perform several experiments.

We have used all the available attributes (62 in total) except those which carry no information about the target variable, e.g., *id*, *date*, *comment*, *session key*, etc. Using algorithms for attribute ranking (Relief, Information gain, Pearson correlation and Symmetric uncertainty) we have shown that the selected attributes carry a small amount of information about the target (the methods were quite consistent about the best 10 attributes) and that the majority of attributes are more or less irrelevant. Consequently, when using decision tree and rule learning algorithms to construct understandable models, these did not lead to good results. However, the Random forest algorithm using 10-fold cross validation gave the best score, which was 14% above the baseline (i.e. 35% classification accuracy).[2]

In the second experiment the target attribute *Relevance* was combined with the binary attribute called *Understandability* to construct a new target attribute which simply tells whether the sentence is appropriate to be asked in a real world situation (it is understandable and the relevance is 4 or 5). The distribution is as follows: *no: 62%, yes: 38%*. This way, the possibility of producing trivial models, fitted on the strong correlation between *Understandability* and *Relevance* was removed. In our experiments the J48 decision tree learning algorithm performed best but only after making the pruning stricter (the default parameters gave the majority classifier).[3] As the target variable was constructed from two informative and correlated attributes the resulting tree is not trivial and contains relevant information about the learning problem. The final J48 decision tree is shown in Figure 3.

The results of our experiments lead to the conclusion that the current set of attributes does not contain enough information to be able to perform consistently better than

[2]Given a small number of examples and low informational value of attributes this model still significantly overfitted the data (this was confirmed by further experiments). The application of such a model would result in a better-than-random ranking of the few best and worst questions, but only on a very similar set of instances. We conducted 50 randomized iterations of a 3-fold cross-validation scenario and the average prediction was consistent with the data on the few ($\approx$10) best and worst ranked questions.

[3]This resulted in a 6% improvement over majority, which is consistent with the observation about the low informativeness of attributes and small number of learning examples.

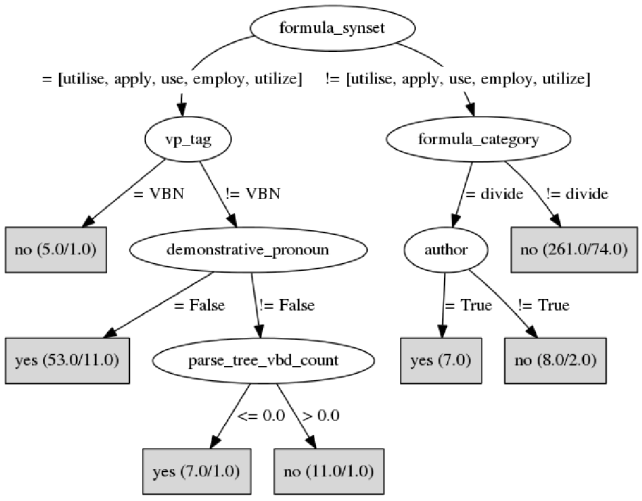

Fig. 3. A J48 decision tree which predicts the appropriateness of questions.

random ranking of questions. However, more data needs to be collected using our crowdsourcing platform in order to obtain statistically well-founded results.

VII. Related Work

Our work can be positioned in the field of AQG based on the final application (goals, question types), the question generation method and the evaluation criteria used.

Application tasks: Majority of AQG systems focus on factual questions, where question generation is the most often the inverse task of Question Answering (e.g. [8], [30], [34]). Only few systems focus on questions for supporting creative and critical thinking in academic writing, one of these being the work of Liu et al. [35] that is the most similar to our work. The authors propose an automated system to help students in critical literature review writing by generating contextualized feedback in form of trigger questions relevant to the target topic. They propose two approaches, one using key phrases and formulating questions based on the information in Wikipedia articles and conceptual graphs [35] and another based on citations [6]. Compared to their systems, our work is not intended for critical literature review but more ambitiously as critical and creative paper review support. Therefore we did not focus on sentences containing citations but on sentences reporting authors decisions and opinions, as well as presuppositions and claims that are not adequately argumented. We agree with [36] and [31] who emphasize that an effective AQG system should focus deeply on the importance of the generated questions.

Question generation methods: Most AQG systems base question generation on linguistic analysis (with some exceptions, e.g. [12]). As it is the case in our approach, the question generation usually consist of sentence selection (by using linguistic anchors, predefined or automatically learned patterns) and question generation phase. One can distinguish between syntax-based, semantic-based and template-based systems. In several cases, data mining is performed for question ranking. There are several systems that are similar to ours by using a syntax and template-based combinations (in form of hand crafted patterns and templates and transformation rules (e.g. [37]). However our linguistic anchors differ from other systems, since we focus mainly on decisions. Unlike other authors, we extend the list of initial patterns by using WordNet. Our data mining methods can be compared to [6] who use 11 features for automatic ranking of citation-based questions. Some of our features are very similar to their system, but we add many features that are specific to our question generation module.

Evaluation criteria: The selection of evaluation categories was inspired by previous studies. First, the binary score evaluating if the question was *understandably formulated/acceptable* can be compared to the "acceptable vs. unacceptable" binary scores in [31] and [6]. Like in their work we combine basic semantics and syntax together. Second, our 5 star *relevance* score scale can be aligned with the evaluation of (topic) relevance in [31]. However, there factual questions were concerned. The relevance is evaluated on the scale from 1 to 5, while their sub-criteria comprise semantic correctness, question type and referential clarity. For us, only the first aspect is relevant, in which 5 is similarly to our case formulated as "the question is meaningful and related to the topic", while the other two sub-criteria do not correspond to our task (since we do not evaluate factual questions and since for understanding by the users who are the authors or reviewers of the paper, we do not need that much referential clarity). We do not precisely evaluate the syntactic aspects, but our binary category on understandability/acceptability can be compared to their test of overall acceptability. Some other related measures, though concerning factual questions, are 4 level semantic and syntactic score scales used in shared task of question generation based on paragraphs [7]. While we do not focus on detailed evaluation of grammaticality, the semantic scores are partly related to our relevance score. In the same competition, where the shared task was AQG based on sentences [7], the "relevance" to the input sentence is measured. It is partially related to our relevance score, but our relevance concerns relevance to the article and not to the input sentence.

VIII. Conclusions and Further Work

The main functionality of the presented online RoboCHAIR system is to automatically generate questions serving the authors as a feedback, and to conference chairs or audience as support in delivering relevant questions. Our AQG system is based on linguistic anchors for sentence identification, while question generation is a combination of a template- and a syntax-based system. User evaluation of questions understandability and relevance shows that 87 percent of automatically generated questions were understandable, while relevance score was approx. 3. The best categories of source sentence selection formulas were evaluated with scores, averaging 4.

Generating relevant questions automatically is a very difficult task. The main challenge for future work is how to surpass the sentence level by avoiding questions that were already answered in the paper and, more ambitiously, to detect what is missing in the article (e.g. evaluation), as well as to generate questions which link the contents of the paper to the related work (e.g. using proceedings from previous conferences and citation analysis). In shorter term, we plan to

improve automated question generation by considering triplet-based question generation, enabling generation of categories uncovered anew. To extend the quality and variety of generated questions, we plan to collect more evaluations through crowdsourcing, as well as to incorporate user-suggested and reviewers' questions to automatically learn from examples of good questions.

ACKNOWLEDGMENT

This work has been partly supported by the EC FP7 projects WHIM (grant no. 611560) and MUSE (296703). We thank also our colleague Bernard Ženko.

REFERENCES

[1] M. Boden, *The Creative Mind: Myths and Mechanisms*. London: Weidenfeld and Nicholson.

[2] S. Colton and G. A. Wiggins, "Computational creativity: The final frontier?" in *Proceedings of 20th European Conference on Artificial Intelligence (ECAI)*, 2012, pp. 21–26.

[3] P. Piwek, H. Prendinger, H. Hernault, and M. Ishizuka, "Generating questions: An inclusive characterization and a dialogue-based application," in *Proc. Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA, Sep. 2008, pp. 25–26.

[4] T. T. Chen, "The development and empirical study of a literature review aiding system," *Scientometrics*, vol. 92, no. 1, pp. 105–116, 2012.

[5] R. D. King, K. E. Whelan, F. M. Jones, P. G. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver, "Functional genomic hypothesis generation and experimentation by a robot scientist," *Nature*, vol. 427, no. 6971, pp. 247–252, 2004.

[6] M. Liu, R. A. Calvo, and V. Rus, "Automatic generation and ranking of questions for critical review," *Journal of Educational Technology & Society*, vol. 17, no. 2, pp. 333–346, 2014.

[7] V. Rus and A. Graesser, Eds., *Workshop Report: The question generation task and evaluation challenge*. Institute for Intelligent Systems, Memphis, 2009.

[8] V. Rus, P. Piwek, S. Stoyanchev, B. Wyse, M. Lintean, and C. Moldovan, "Question generation shared task and evaluation challenge: Status report," in *Proceedings of the 13th European Workshop on Natural Language Generation*, ser. ENLG '11. Stroudsburg, PA, USA: ACL, 2011, pp. 318–320.

[9] S. Kalady, A. Elikkottil, and R. Das, "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)*, Pittsburgh, PA, Jun. 2010, pp. 1–10.

[10] P. Piwek and S. Stoyanchev, "Question generation in the coda project," in *Proc. Third Workshop on Question Generation, The Tenth International Conference on Intelligent Tutoring Systems (ITS'2010)*, Pittsburgh, PA, Jun. 2010, pp. 29–34.

[11] J.Sullins, A. Graesser, K. Tran, S.Ewing, and N. Velaga, "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)*, Pittsburgh, PA, Jun. 2010, pp. 21–28.

[12] N. Khodeir, N. Wanas, N. Darwish, and N. Hegazy, "Bayesian based adaptive question generation technique," *Journal of Electrical Systems and Information Technology*, vol. 1, pp. 10–16, 2014.

[13] F.-Y. Yu, "Scaffolding student-generated questions: Design and development of a customizable online learning system," *Computers in Human Behavior*, vol. 25, no. 5, pp. 1129–1138, 2009.

[14] E. V. Wilson, "Examnet asynchronous learning network: Augmenting face-to-face courses with student-developed exam questions." *Computers & Education*, vol. 42, no. 1, pp. 87–107.

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

[16] L. P. Steffe, "The constructivist teaching experiment: Illustrations and implications," in *Radical constructivism in mathematics education*, E. von Glasersfeld, Ed., 1991, pp. 177–194.

[17] D. R. Geelan, "Epistemological anarchy and the many forms of constructivism," *Science & education*, vol. 6, no. 1, pp. 15–28, 1997.

[18] F.-Y. Yu and Y.-H. Liu, "Creating a psychologically safe online space for a student-generated questions learning activity via different identity revelation modes," *British Journal of Educational Technology*, vol. 40, no. 6, pp. 1109–1123, 2009.

[19] A. Graesser, P. Chipman, B. Haynes, and A. Olney, "Autotutor: an intelligent tutoring system with mixed-initiative dialogue," *Education, IEEE Transactions on*, vol. 48, no. 4, pp. 612–618, Nov 2005.

[20] D. Biber, *Discourse on the Move: Using Corpus Analysis to Describe Discourse Structure*. John Benjamins Publishing, 2007.

[21] G. Sharpling. (2015, Apr.) Learning English online at Warwick university - reporting verbs. [Online]. Available: http://www2.warwick.ac.uk/fac/soc/al/learning_english/leap/grammar/reportingverbs/#Q1

[22] M. Paquot, *Academic vocabulary in learner writing: From extraction to analysis*. Bloomsbury Publishing, 2010.

[23] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[24] D. C. Howe, "Rita: Creativity support for computational literature," in *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, 2009, pp. 205–210.

[25] A. Bies, M. Ferguson, K. Katz, and R. MacIntyre, *Bracketing Guidelines for Treebank II Style Penn Treebank Project*. [Online]. Available: http://languagelog.ldc.upenn.edu/myl/PennTreebank1995.pdf

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

[27] S. Pollak, A. Vavpetic, J. Kranjc, N. Lavrac, and S. Vintar, "NLP workflow for on-line definition extraction from English and Slovene text corpora," in *Proc. 11th Conference on Natural Language Processing, KONVENS 2012*, Vienna, 2012, pp. 53–60.

[28] M. Scott. (2008) BNC word list from WordSmith Tools version 5. [Online]. Available: www.lexically.net/wordsmith/

[29] P. Velardi, S. Faralli, and R. Navigli, "Ontolearn reloaded: A graph-based algorithm for taxonomy induction," *Computational Linguistics*, vol. 39, no. 3, pp. 665–707, 2013.

[30] M. Heilman and N. A. Smith, "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*, ser. HLT '10, 2010, pp. 609–617.

[31] Y. Chali and S. A. Hasan, "Towards automatic topical question generation," in *Proceedings of International Conference on Computational Linguistics (COLING 2012)*, Mumbai, India, December 2012, pp. 475–492.

[32] K. Krippendorff, "Systematic and random disagreement and the reliability of nominal data," *Communication Methods and Measures*, vol. 2, no. 4, pp. 323–338, 2008.

[33] D. Freelon. (2015, Apr.) ReCal: reliability calculation for the masses. [Online]. Available: http://dfreelon.org/utils/recalfront/recal-oir/

[34] L. Becker, S. Basu, and L. Vanderwende, "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*, ser. NAACL HLT '12, 2012, pp. 742–751.

[35] M. Liu, R. A. Calvo, A. Aditomo, and L. A. Pizzato, "Using wikipedia and conceptual graph structures to generate questions for academic writing support," *Learning Technologies, IEEE Transactions on*, vol. 5, no. 3, pp. 251–263, july-sept 2012.

[36] L. Vanderwende, "The importance of being important: Question generation," in *Proc. Workshop on the Question Generation Shared Task and Evaluation Challenge*, Arlington, VA, Sep. 2008.

[37] S. Curto, A. C. Mendes, and L. Coheur, "Exploring linguistically-rich patterns for question generation," in *Proceedings of the UCNLG+Eval: Language Generation and Evaluation Workshop*, ser. UCNLG+EVAL '11, 2011, pp. 33–38.