Received: 7 September 2019 | Revised: 16 June 2020 | Accepted: 17 June 2020

DOI: 10.1002/widm.1382

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY WILEY

Automatic question generation

Mark Last | **Guy Danon**

Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Correspondence
Mark Last, Department of Software and Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel.
Email: mlast@bgu.ac.il

Funding information
IBM Research, Grant/Award Number: 15/12/144

Abstract

Automatic generation of semantically well-formed questions from a given text can contribute to various domains, including education, dialogues/interactive question answering systems, search engines, and more. It is well-known as a challenging task, which involves the common obstacles of other natural language processing (NLP) activities. We start this advanced review with a brief overview of the most common automatic question generation (AQG) applications. Then we describe the main steps of a typical AQG pipeline, namely question construction, ranking, and evaluation. Finally, we discuss the open challenges of the AQG field that still need to be addressed by NLP researchers.

This article is categorized under:
Algorithmic Development > Text Mining

KEYWORDS

automated question generation, natural language generation

1 | INTRODUCTION

Automatic question generation (AQG) was defined by Rus, Cai, and Graesser (2008) as "the task of automatically generating questions from various inputs such as raw text, database, or semantic representation." Question generation (QG) is an important element of learning and educational environments, information retrieval systems, help systems, and other applications. Writing good questions, though, is a challenging work, which can be a time-consuming and labor-intensive process. The increasing availability of digital information, together with the deployment of various question-answering applications, led to the development of research in AQG.

AQG from text has three major application domains (Le, Kojiri, & Pinkwart, 2014):

1. *Educational assessment*. This application domain of AQG systems aims at generating test items that assess the understanding and knowledge of students. According to Mitkov, Le An, and Karamanis (2006), the time required by the instructors for postediting automatically generated questions is expected to be significantly less than for a completely manual process. Moreover, a semi-automatic process may result in test items of higher quality.
2. *Knowledge and skills acquisition*. These systems focus on improving the student skills, such as reading comprehension, academic writing, and others, by generating feedback questions. Evaluation studies show that such systems can outperform human peer reviews after handling grammatical and semantic errors.

Abbreviations: AQG, automatic question generation; NLG, natural language generation; NLP, natural language processing.

Mark Last and Guy Danon equally contributed equally to this work.

3. *Dialogues/interactive question answering (QA) systems*. These systems are aimed at generating questions for tutorial dialogues. This category also includes construction of large corpora of source–question–answer triplets for training QA systems such as IBM Watson.

Brown, Frishkoff, and Eskenazi (2005) describe an approach to automatically generating six types of questions for vocabulary assessment: definition, synonym, antonym, hypernym, hyponym, and cloze questions. In order to generate questions of these types, their system uses data from WordNet (Miller, 1995). According to the assessed vocabulary knowledge, the system can automatically provide users with texts to read targeted to their individual reading levels. It has been shown that using the automatically generated questions students achieved a measure of vocabulary skill that correlates well with performance on independently developed human-generated questions.

Another approach to vocabulary and test-takers' proficiency assessment was presented by Sumita, Sugaya, and Yamamoto (2005). Their method generates fill-in-the-blank questions using a corpus, a thesaurus, and the Web. The questions are created by replacing verbs with gaps in an input sentence, while the distractors are retrieved from a thesaurus.

There are several studies (e.g., Heilman, 2011; Kalady, Elikkottil, & Das, 2010) aimed at creating systems for AQG that can take as input a text document and create as output a ranked list of factoid questions. These questions are used to ask for fact-based answers, and usually start with one of these question tools: what, where, when, and who. A user could then review, filter, and revise these automatically generated questions in order to create practice exercises or part of a quiz to assess the learners comprehension after they finished reading the material and their knowledge about the topic. The main drawback of such factual-question methods is that the generated questions may be less meaningful than manual questions.

Deep questions have a great educational value because they require deep thinking and recall rather than a rote response, expecting the students to manipulate various pieces of previously acquired information in order to formulate an answer. Labutov, Basu, and Vanderwende (2015) developed an approach for generating high-level comprehension questions rather than factoid questions from a novel text. The method involves three stages: first the text is decomposed into a two-dimensional ontology as category-section pairs (e.g., category: person, section: early life), then high-level templates are solicited from the crowd, and finally a subset of the templates is retrieved for a target text segment based on its ontological categories. Deep and factual questions may complement each other: lower-level questions ensure that students possess the basic knowledge needed to answer higher-level questions.

The purpose of the second context of AQG systems includes knowledge and skills acquisition. Le et al. (2014) introduced a technical approach to generating questions semantically related to a given discussion topic in order to help students develop further arguments. As an input, the QG system takes an English text from the discussion topic, which can be an individual word, a list of words, a phrase, a sentence/question, or a paragraph. They combined different natural language processing (NLP) technologies and exploit semantic information that can give ideas related to the discussion topic and support the users in developing their arguments in a discussion session via tailored questions of different types.

When students are asked to prepare a literature review or write an essay, they usually have to learn and reason from multiple documents. Providing simple generic questions (e.g., *Have you clearly identified the contributions of the literature reviewed?*) leads to better writing. However, on a specific topic more content-related questions need to be asked as feedback to students. A pilot study by Liu, Calvo, and Rus (2010) suggests that AQG system can produce questions that are as helpful to promote students' reflection on their academic writing as those by human tutors.

Dialogues and interactive answering systems are additional applications of AQG. Bednarik and Kovacs (2012) gave an example of robot control, which would benefit from using generated questions. They claimed that the functionality of robotic assistants, which often need to communicate with the users, can be significantly improved with a sequence of question–answer pairs.

Chali and Hasan (2015) studied the problem of topic-to-question generation. Their method is focused on automatic generation of all possible questions from a topic of interest, where each topic is associated with a body of texts containing useful information. The main goal of QA systems is to retrieve relevant answers to natural language questions from a collection of documents, in contrast to search engines that use keyword matching techniques to extract documents. The input of such system must be a well-formed question. Because of the fact that humans are not very skilled in asking good questions about a topic of their interest, an AQG system can assist in receiving their requested answer.

QG has been used for automatically producing dialogues from expository texts (Piwek & Stoyanchev, 2010). These dialogues can serve for presenting medical information in the form of written text or be used by virtual agents with

speech synthesis. Another benefit of AQG is that it can be a good tool to help improve the quality of the QA systems, because the quality of the interactions between the system and the user can be improved if a QA system is able to predict some of the questions that the user may wish to ask (Hussein, Elmogy, & Guirguis, 2014).

Olney, Graesser, and Person (2012) presented a QG approach suitable for tutorial dialogues, based on previously developed psychological theories that hypothesize questions are generated from a knowledge representation modeled as a concept map. Their approach automatically extracts concept maps from texts in the domain of biology using frequency measures and external ontology. Five question types are generated from the concept maps: hint, prompt, forced choice question, contextual verification question, and causal chain questions.

Studies on AQG regard the AQG task as a process composed of four main steps: (a) defining the trigger to ask the question, (b) text preprocessing and relevant content selection, (c) question construction, and (d) ranking the constructed questions. The first step is particularly relevant to dialogues and QA systems, where it is important when to pose questions. For example, a system may generate a question as a response to the user activity/question. In the next three sections of this review, we focus on steps 2, 3, and 4, respectively. Then we cover methods for evaluating the quality of automatically generated questions. Finally, we discuss the limitations of existing works and the future directions of AQG research.

2 | TEXT PREPROCESSING AND RELEVANT CONTENT SELECTION

The step of preprocessing and relevant content selection deals with a major challenge of identifying the content related to the key concepts in a given text over which questions may be asked. This sub-task of the AQG process involves various techniques from the field of NLP, such as parsing, sentence simplification, anaphora resolution, named entity recognition (NER), semantic role labeling (SRL), and more. Chali and Hasan (2015) propose to simplify complex sentences with the intention of generating more accurate questions, as it is easier to generate questions from simple sentences. For this, they use the simplified factual statement extractor model of Heilman and Smith (2009). However, sentence simplification may lead to the loss of important semantic information. For example, given the sentence "The quantity theory of money also states that the growth in the money supply is the primary cause of inflation," the following simplified sentence may by created: "The growth in the money supply is the primary cause of inflation," which leads to the generation of the question: "What is the primary cause of inflation?". That question takes the cause of inflation out of the context of discussing a theory and places it at the level of a fact Mazidi and Nielsen (2015).

Named entities (NEs) are generally noun phrases in an unstructured text, for example, names of persons, posts, locations, and organizations. Recognition of NEs is an essential task in many NLP applications nowadays. In the task of QG, it is important to identify those facts (terminology, entities, and semantic relations between them) that are likely to be important for assessing test-takers' familiarity with the instructional material. State-of-the-art NER taggers are Stanford NER Manning et al. (2014) and Illinois NER Ratinov and Roth (2009).

Bhatia, Kirti, and Saha (2013) proposed a methodology for potential sentence selection with the help of existing test items in the web. For selecting the potential sentences, they first collect domain-specific multiple choice questions (MCQ) available from different sources. Next they form syntactically correct sentences from the MCQs, by replacing the WH-phrase in the first option, which is not necessarily the correct answer. They extract patterns from these sentences by replacing identified entities from Stanford NER by variables and finding the most frequent n-grams that are not occurring in the general domain pages (e.g., *PER be the captain of LOC*). A pattern is likely to extract sentences containing a particular type of entities. For example, "the Man of the Tournament" pattern will extract sentences from the sports domain having the name of the player who was awarded the Man of the Tournament in a particular competition. The sentences are tagged using the NER system and the corresponding entity is selected as a key concept for the question.

Afzal and Mitkov (2014) looked for an information extraction (IE) method that has a high precision and at the same time works with unrestricted semantic types of relations in the domain text. For this purpose, they use unsupervised relation extraction to identify the most important named entities and terminology in a document and then recognize semantic relations between them, without any prior knowledge. They assume that relations can be extracted between NEs stated in the same sentence. The extraction is done by first processing unannotated text by GENIA tagger for biomedical named entities (Tsuruoka et al. (2005) and then extracting patterns from the text through the linked chain pattern model. For example: from the sentence "Fibrinogen activates NF-kappaB transcription factors in mononuclear phagocytes," the pattern "PROTEIN activates PROTEIN PROTEIN in CELL" is created. The use of unsupervised IE for

AQG offers some important advantages: it is able to capture a wide range of important facts contained in instructional texts and do so with a high accuracy; it has a potentially unrestricted coverage as it does not target any predefined types of semantic relations; and it is applicable to situations where manually annotated text is unavailable.

SRL is used to identify semantically meaningful parts for predicates in a sentence along with their semantic roles, that is, the predicate-argument structure of sentences. When presented with a sentence, an SRL system performs a full syntactic analysis of the sentence, automatically identifies all verb predicates in that sentence, extracts features for all constituents in the parse tree relative to the predicate, and identifies and tags of the constituents with the appropriate semantic arguments. Semantic roles can describe WHO did WHAT to WHOM, WHEN, WHERE, WHY, HOW, and so on. for a given situation, and contribute to the construction of meaning. The arguments can be used to ascertain whether a particular constituent is appropriate for generating a question or not, and therefore use of SRL is attractive for AQG. State-of-the-art SRL parsers are ASSERT (Pradhan, Ward, & Martin, 2008) and SENNA (Collobert, 2011), which are in use for example in the works of Mannem, Prasad, and Joshi (2010) and Mazidi and Nielsen (2015), respectively.

Both NER and SRL can contribute to understanding the meaning of the text, and thus the recognition of named entities and semantic roles may be an essential task in the step of content selection for AQG. The attributes extracted from both NER and SRL usually act as the target words, which are searched for in the sentence whose question type is being identified. Fattoh, Aboutabl, and Haggag (2015) propose an AQG model for evaluating the understanding of semantic attributes in a sentence. As a preprocessing and feature extraction step, they use the University of Illinois' SRL and NER to extract attributes from the input sentence. Then, sentence and question patterns are generated for sentences used in the training phase, and a learning algorithm is used to generate a classifier able to classify the test sentence according to its question type and also use the classifier patterns to generate a question pattern for the test sentence. The question types considered here are the set of WH-questions like who, when, where, why, and how.

Chali and Hasan (2015) generated questions using a set of general-purpose rules based on named entity information and the predicate argument structures of the sentences (along with semantic roles) presenting in the associated body of texts. They first tag plain text with named entities (people, organizations, locations, miscellaneous) using Illinois NER, and by some general purpose rules they create a basic question. For example: Apple Inc. is tagged as an organization, so a question such as "Where is Apple Inc. located"? is generated. Such questions are generated even though the answer is not present in the body of the text, in order to add variety to the generated question space. Additional parsing is done using ASSERT SRL, and the output arguments are used to generate specific questions from the sentences. For example: the output of the SRL system for the words "Apple's first logo" from the sentence "Apple's first logo is designed by Jobs and Wayne" is the tag AGR1 of the target-verb "designed." That argument can be replaced with "What" to generate a question like: "What is designed by Jobs and Wayne"?

Additional strategy for relevant content selection is to extract specific sentences, which contain some words or phrases required by the expected type of question. Agarwal and Mannem (2011) presented a system that automatically generates questions from natural language text using discourse connectives. Discourse connectives play a vital role in making the text coherent. They connect two clauses or sentences exhibiting discourse relations such as temporal, causal, elaboration, contrast, result, and so on. The researchers provide analysis for four subordinating conjunctions (*since*, *when*, *because*, and *although*), and three adverbials (*for example*, *for instance*, and *as a result*), which have been found to be good candidates for generating wh-questions, and cover 41.97% of the total number of connectives in Wikipedia articles. The system goes through the entire document and identifies the sentences containing at least one of the selected connectives. The suitable content for each discourse connective, which is referred to as a *target argument*, is determined based on its own structural and anaphoric properties. A target argument for a discourse connective can be a clause(s) or a sentence(s). The content of the discourse connectives *for example* and *for instance* is considered as the sentences in which the connectives occur at the beginning of the sentence together with their immediate previous sentence. In the case of other connectives, the content is selected using the dependency tree of the sentence.

The research of Wang, Hao, and Liu (2007) focuses on health-related concepts in order to provide interesting questions from given learning material of medical text. The medical concepts are extracted using the Unified Medical Language System (UMLS) Bodenreider (2004), which contains over 1 million biomedical concepts and 5 million concept names, 135 semantic types and 54 relationships. The system maps the noun phrases in the text to the best matching UMLS concept, and generates synonyms, acronyms, abbreviations, and spelling variants for query expansion. For QG, only sentences with at least two medical entity types are selected.

A different approach to content selection as a sub-task of AQG is presented by Bednarik and Kovacs (2012), and it is based on clustering and classification. The words of the source text are clustered into partitions to select the best

suitable sentences for QG. Their system uses a semi-automated selection method for determining the base sentences for generating the questions. Initially, an annotation framework is used by human experts to denote which sentences are suitable and optimal for QG. Then a classification algorithm is trained on that training set to find the relationships between the sentence features and its annotation.

3 | QUESTION CONSTRUCTION

The main step in AQG is the construction of questions from the selected content. Most AQG systems generate questions by selecting one sentence at a time. Graesser and Person (1994) propose to organize questions by five characteristics: their purpose, the type of information they seek, their sources of information, the length of the expected answer, and the cognitive processes they involve. Based on the type of information involved, they present a set of 16 categories for questions ranging from simple to complex questions. Simple questions are considered as verification/concept completion/example, and so on, while the complex ones are causal consequence/goal orientation/enablement/expectation/judgmental, and more. Wang et al. (2007) consider some criteria that can prevent generation of arbitrary questions: a question should be answerable, interesting, and domain-related (the medical domain in their research).

Existing AQG approaches can be classified into three categories: syntax-based, template-based, and semantics-based Le et al. (2014). Syntax-based approaches (also called transformation-based approaches) are especially effective for short sentences, where questions are generated about explicit factual information at the sentence level. They usually work through the following three steps Le et al. (2014): (a) delete the identified target concept, (b) place a determined question key word at the first position of the question, and (c) convert the verb into a grammatically correct form considering auxiliary and model verbs. For example, the sentence *Barack Hussein Obama II served as the 44th president of the United States from 2009 to 2017* can be transformed into the following question: *Who served as the 44th president of the United States from 2009 to 2017?* An algorithm can perform the above steps without knowing any underlying meaning of the transformed sentence and it may generate ungrammatical questions, which come from the shallow realization of transformed parse trees representing the input sentences.

Heilman (2011) focus on automatically generating factual WH-questions by applying combinations of general rules. Their system breaks AQG down into a multistep process: simplified factual statements are first extracted from complex inputs by (optionally) altering or transforming lexical items, syntactic structure, and semantics. Next, the sentences are separately transformed into questions by applying sequences of simple, linguistically motivated transformations such as subject-auxiliary inversion and WH-movement. They employ some core NLP tools in their system in order to analyze the linguistic properties of input sentences: Stanford Phrase Structure Parser to automatically sentence-split, tokenize, and parse input texts; the Tregex Tree Searching Language and Tool to identify the relevant syntactic elements (e.g., subjects of sentences); Supersense Tagger to label word tokens with high-level semantic classes such as noun.person, noun.location, and so on; and ARKref Noun Phrase Coreference Tool to identify the antecedents of pronouns.

A set of transformation rules for question formation is also in use in the system of Varga and Ha (2010). For subject-verb-object clauses whose subject has been identified as a target concept, a "Which Verb Object" template is selected and matched against the clause. For key concepts that are in the object position of a subject-verb-object, the verb phrase is adjusted (i.e., auxiliary verb is used).

The system of Agarwal and Mannem (2011) uses analysis of the senses of discourse connectives to generate questions of the type *why*, *when*, *give an example*, and *yes*/*no*. For sentences containing at least one connective, the system finds the question type on the basis of discourse relation (e.g., the sense of the discourse "because" is "casual," so the selected question type is "why"). In order to get the final question, a set of transformations is applied to the content and the arguments found in the task of content selection. For example, from the sentence "Because shuttlecock flight is affected by wind, competitive badminton is played indoors," the following question is generated: "Why is competitive badminton played indoors"?

The template-based approaches rely on the idea that a question template can capture a class of context-specific questions with the same structure. For example, Mostow and Chen (2009) developed templates such as: "What would happen if <X>?" for conditional text, and "Why <auxiliary-verb><X>?" for linguistic modality, where <X> is the placeholder mapped to semantic roles annotated by a semantic role labeler. These question templates can only be used for these specific entity relationships, whereas for other kinds of entity relationships, new templates must be defined. These approaches are mostly suitable for applications with a special purpose, which sometimes comes within a closed

domain. In a common template, there are three components: question and answer, which are the output from the learning document, and entries, which are used for template matching.

The AQG approach of Liu et al. (2010) generates trigger questions as a form of support for students' learning through writing. The approach is based on a set of templates and the content elements which are extracted from citations in student compositions. In order to extract the required features, they use Tregex and Stanford Parser for the syntactic features and LBJ NER for the semantic. The system uses reporting verb lists to determine the corresponding citation category (opinion, result, aim of study), and once the features extracted from a citation match the predefined patterns in the repository of templates, the corresponding questions are generated.

The approach of Wang et al. (2007) generates questions automatically based on question templates, which are created by training on a corpus of medical articles. In that research, nearly 100 templates were created in every aspect of medical domain by experts based on parsed articles. In addition to the question, answer, and entries, they add keywords to every template, in order to indicate relationships and relevant attributes of the key concepts represented by entity slots. The templates contain medical concepts (e.g., disease, medicine, symptom, etc.) in order to provide interesting questions. A sample template is given as follows: "What is the symptom for <disease>?," where the required entries are <symptom> and <disease>, the keywords may be <feel> and <experience>, and the answer is <symptom>. The authors introduce a method to calculate the weight of the templates for template selection, which is affected by two factors: (a) the number of concepts and the concept types found in the sentence, and (b) the quality of the questions generated from this template.

Hussein et al. (2014) propose a rule-based QG system that is trained by storing a huge amount of template rules in the data store. The proposed system uses pure syntactic pattern-matching approach to generate content-related questions in order to improve the independent study of any textual material. The training phase is performed by annotating various sentences related to an application domain with certain keywords and POS (part-of-speech) tags available in OpenNLP. Based on the associated POS tags, the system tries to find a similar template rule in its data store, and if there is no match, the user is asked to add a new template. For doing this, the user is prompted for a WH-type, a new verb, and question tags.

In addition to questions that can be generated using phrases in a statement, semantic information and background knowledge related to the statement topic can also be exploited to generate questions. The semantic-based approaches usually retrieve semantic information from diverse sources, such as WordNet (Miller, 1995 and other domain ontologies, in order to generate more interesting and challenging questions.

Le et al. (2014) propose a QG approach, which makes use of semantic information available on WordNet in the purpose of giving students ideas related to a discussion topic and guiding them how to expand it. After analyzing and parsing a given topic text to extract important concepts, every extracted noun or noun phrase is used as a resource to search for its related concepts in WordNet. The proposed approach consists of three steps. First, question templates are generated without using WordNet (e.g., "What is <X>," whereas X is a noun or a noun phrase extracted from the discussion topic). The second step is question construction using retrieved hyponyms and the previously generated templates (e.g., "What is activation energy," where "activation energy" is one of the hyponyms of the noun "energy"). Finally, questions are generated using the example sentences provided by WordNet and ARK (Heilman & Smith, 2009, a syntax-based tool for generating questions from English sentences or phrases, which achieved 43.3% acceptability for the top 10 ranked questions and produced an average of 6.8 acceptable questions per 250 words of Wikipedia texts.

For domain-specific questions, specific ontologies can be used during the QG process. Papasalouros, Kanaris, and Kotis (2008) present an approach, which creates multiple choice question items using Ontology Web Language, which is independent of lexicons such as WordNet or other linguistic resources. The proposed approach is domain-independent since questions are generated according to three specific ontology-based strategies. In class-based strategies the generated answers are of the type "instance a is a class A," which is "is-a" relationship between a member and a class. The property-based category contains strategies that create question items and distractors based on properties (roles), that is, relationships between individuals in the ontology (e.g., "Eupalinian Aqueduct brings water to ancient city of Samos"). Other strategies are terminology-based, which are based on concept/subconcept relationships, without dealing with ontology individuals at all (e.g., "sovereigns are politicians"). In this approach, all types of questions are in the format of "Choose the correct sentence," where one choice is the correct answer, while the others are distractors generated from the ontology.

Wikipedia provides definitions of words and descriptions of concepts. Using Wikipedia as a source of information, as opposed to an ontology or WordNet, lets systems be easily transferred to any domain or language. Bhatia et al. (2013) use Wikipedia in their system for automatic generation of multiple choice test items. Potential sentences are searched

on Wikipedia according to their matching with the templates of sentences from which sample MCQs were derived. In addition, they use Wikipedia for generating interesting and relevant distractors.

Each of the different approaches has its own advantages in the task of AQG. While the template-based methods usually generate domain-related questions that are correct grammatically, the syntax-based methods provide better coverage of the text, and the semantic-based methods use some background resources in order to provide more interesting questions. Some of the more recent techniques try to combine the different approaches in order to develop high quality questions.

Mazidi and Nielsen (2015) present a template-based QG system that is built on multiple views of text: syntactic structure retrieved from dependency parse, paired with information from semantic role labels and discourse cues. Their system builds a lexical-semantic tree structure using dependencies from the Stanford dependency parsers and semantic roles extracted from SENNA. The inclusion of modifiers such as casual, temporal, and locative from SENNA allows the generation of semantically oriented questions such as *how* and *why* questions. For QG, each sentence is matched against a list of around 50 possible patterns, which were manually constructed to match features that might appear within the tree structure of each sentence. Questions are generated whenever a pattern matches the sentence. Their results show that the dependency parse can provide a good foundation for QG, particularly when combined with information from multiple sources.

The research of Afzal and Mitkov (2014) is aimed at generating questions regarding the important concepts presenting in a domain by relying on unsupervised extraction of semantic relations (dependency-based patterns). They traverse the entire dependency tree of an extracted sentence and identify all words, which depend on the main verb presenting in the dependency pattern. For example, the sentence "The predicted periplasmic domain of the PhoQ protein contained a markedly anionic domain that could interact with cationic proteins and that could be responsible for resistance to defensing" is matched with the pattern "PROTEIN of PROTEIN contain," which contains its instantiation. Based on the presence of the main verb from the dependency pattern, the first part of the sentence is transformed to "Which protein of the PhoQ protein contained a markedly anionic domain"?

Labutov et al. (2015) propose a crowdsourcing-based methodology for generating deep comprehension questions from novel text. They use an ontology-crowd-relevance workflow for generating high-level questions that cannot be answered from a single sentence in the article. Their method involves three stages: first the text is decomposed into a two-dimensional ontology represented as category-section pairs (e.g., category: person, section: early life), then high-level templates are solicited from the crowd workers (e.g., "Who were the key influences on <Person> in his/her childhood?"), and finally a subset of templates is retrieved for a target text segment based on its ontological categories. In this research, the ontology is a Cartesian product of article categories and article section names (derived from Freebase and Wikipedia, respectively), and it is used by the crowd workers for creating the relevant templates. The templates are converted into questions by filling in the article-specific entity extracted from the title. If ontological labels are not available, they are inferred from the text by a standard text classification algorithm using basic tf-idf features. Similarly, the relevance of a question template to an article can be estimated using the Euclidean distance between the question features and the article segment.

In contrast to the previous, mostly rule-based methods, Du, Shao, and Cardie (2017) have defined the task of QG as a purely data-driven sequence-to-sequence learning problem that directly maps a preselected text passage, spanning a sentence or a paragraph, to a question. Both the input text and the output question are modeled as two distinct word sequences, which does not restrict the generated questions to the words in the input text or even to the entire vocabulary of input sequences. This is an important property of good reading comprehension questions focused on understanding rather than just memorizing the learned material. The sequence-to-sequence QG algorithm selects the next output token by maximizing the conditional log-likelihood of the predicted question sequence, given the input text and the previously selected tokens. The conditional probability of each token is calculated using the long short-term memory encoder–decoder architecture trained on a corpus of sentence–question pairs, where each word is represented as a pretrained embedding of 300 dimensions. In Du et al. (2017), the sequence-to-sequence method is trained and tested on disjoint subsets of the SQuAD corpus Rajpurkar, Zhang, Lopyrev, and Liang (2016), where it naturally outperforms the rule-based method of Heilman (2011), which apparently was not tuned on the same dataset.

Recently, the work of Du et al. (2017) was extended in several directions. Thus, assuming that the answer contains certain spans of the text from the input passage (like in the SQuAD corpus), Song, Wang, Hamza, Zhang, and Gildea (2018) enrich the encoder with the question context calculated as matching between the target answer and the entire passage. Another answer-aware QG method, which utilizes paragraph-level context, is presented in Zhao, Ni, Ding, and Ke (2018). In contrast to these two works, Scialom, Piwowarski, and Staiano (2019) show how to adapt a

novel, transformer architecture to the task of neural question generation when the answer is not provided as part of the input. To deal with rare/unseen words, they enhanced the basic Transformer architecture with a copying mechanism, a placeholding strategy, and contextualized word embeddings.

4 | QUESTION RANKING

Using various AQG methods, an exhaustive list of questions can be generated from an input text passage. A high percentage of these questions may be useless and unacceptable, due to incorrect syntactic structures, nonrelevance to the main topics in the text, and so on. Manual filtering of automatically generated questions is a time-consuming task. On the other hand, generating *all* possible questions from a given text is rarely necessary. Therefore in AQG, precision is much more important than recall (Afzal & Mitkov, 2014). Some of the recent AQG systems improve the precision rate by over-generating questions and selecting the most meaningful ones based on a ranking method. The ranking step is common in the syntax-based approaches, where the questions are generated using transformation rules and multiple questions are generated from every input sentence.

The question-ranking module of the system proposed by Chali and Hasan (2015) ranks the questions by combining the topic relevance scores and the syntactic similarity scores. Their module uses Latent Dirichlet Allocation to identify the important subtopics from a given body of texts and Extended String Subsequence Kernel to measure the similarity of the generated questions to those subtopics. To judge the syntactic correctness of each generated question, they compute the syntactic similarity of each question with the associated content information, as the number of common subtrees between their parse trees. Each sentence contributes a similarity score to the questions and then the questions are ranked by the average of their syntactic similarity scores.

A statistical model of question acceptability, based on least squares linear regression, is proposed by Heilman (2011). They model quality as a continuous variable ranging from 1 to 5, which is a linear function of a vector of linguistic factors such as grammaticality, vagueness, the use of an appropriate WH word, and so on. In total, 179 features were defined by analyzing questions generated from development data.

Bunescu and Huang (2010) present a machine learning approach for the task of ranking previously answered questions with respect to their relevance to a new, unanswered reference question. The relations between the reference question and the ranked questions are divided into three categories: reformulation, useful, and neutral. They propose a question usefulness classification framework for the possible relations between questions, which is based on both syntactic and semantic similarity measures. The scores of different similarity measures are used as features for training the classifier, such as Bag-of-Words Similarity and Syntactic Dependency Similarity. The motivation behind these relations is that many applications can benefit from them. For instance, similar questions (i.e., paraphrases of a given question) can be found and complex questions can be decomposed (e.g., into entailment/subset relations).

Mannem et al. (2010) rank the list of generated questions based on two criteria. First, the questions are ranked by the depth of the predicate in the dependency parse tree. This ranking method ensures that the questions generated from main clauses get a higher rank than the ones generated from subordinate clauses. Next, questions with same rank are ordered by the number of pronouns occurring in the question. However, these two ranking criteria are insufficient for choosing the most meaningful questions. Therefore, McConnell, Mannem, Prasad, and Joshi (2011) have implemented two new components to improve the ranking process. They use topic scoring, a technique developed for automated text summarization, to identify important information for questioning and language model probabilities to measure the question grammaticality.

5 | QUESTION EVALUATION

Automatic evaluation of any natural language generated text is a complicated task, and similarly to other natural language generation techniques, most AQG systems are evaluated manually. Each question is usually evaluated by two or more domain experts with good language proficiency and its final score is calculated as the average of the judges' scores. The evaluation in Agarwal and Mannem (2011) is completed on the scale of 1 to 4 (4 being the best score) with respect to syntactic and semantic correctness of the question and an overall rating on the scale of 1 to 8 (4 + 4) is assigned to each question. The syntactic correctness is evaluated from grammatically unacceptable question to grammatically correct and idiomatic/natural. The semantic correctness is varied from semantically unacceptable to semantically correct.

Brown et al. (2005) compared the student performance (accuracy and response time) on computer and human-generated questions. Behavioral measures of vocabulary knowledge were acquired for 75 target words using various computer-generated as well as human-generated question types. They have presented evidence that the computer-generated questions provide a measure of vocabulary skill for individual words that correlates well with human-written questions and standardized assessments of vocabulary skill.

The judges in Liu et al. (2010) were asked to rate each question along several aspects of quality: (a) this question is correctly written; (b) this question is clear; (c) this question is appropriate to the context; (d) this question makes me reflect about what I have written; and (e) this is a useful question. In addition, the judges were asked to identify whether the question was generated by a human (lecturer, tutor, generic) or a system. Another scale to evaluate the quality of questions was used in Mazidi and Nielsen (2015), where workers were asked to evaluate each question using the labels "not acceptable," "borderline acceptable," and "acceptable." In the research of Bhatia et al. (2013), the parameters for evaluation of multiple-choice questions are: whether the question is relevant to the domain of interest, the key is chosen properly, the question is formed properly, the question is over-informative or under-informative, the distractors are related to the key, and at least one distractor is close to the key.

Chali and Hasan (2015) evaluate the performance of QG systems using two criteria: topic relevance and syntactic/grammatical correctness. The score of each of the above varies from very poor to very good on the Likert scale of [1...5]. Topic relevance is guided by the consideration of the following aspects: (1) semantic correctness (i.e., the question is meaningful and related to the topic), (2) correctness of question type (i.e., a correct question word is used), and (3) referential clarity (i.e., it is clearly possible to understand what the question refers to).

In Du et al. (2017), sentence-question pairs are evaluated by four professional English speakers with respect to two modalities: *naturalness*, which indicates the grammaticality and fluency; and *difficulty*, which measures the sentence-question syntactic divergence and the reasoning needed to answer the question. In addition, the human raters are asked to rank the questions according to their overall quality. The annotators of Scialom et al. (2019) evaluate each question using the following criteria: answerability, relevance, correctness, soundness, and fluency.

With the advance of neural question generation methods, automatic evaluation of question quality against gold standard questions provided by benchmark datasets, like SQuAD, is becoming increasingly popular. For example, Du et al. (2017) use the following evaluation metrics: BLEU 1, BLEU 2, BLEU 3, BLEU 4, METEOR, and ROUGE-L. BLEU-n measures the average n-gram precision on a set of reference questions, with a penalty for overly short questions. METEOR calculates the similarity between generated and reference questions by considering synonyms, stemming and paraphrases. ROUGE-n metrics calculate n-grams recall of the generated questions with gold standard questions as references. ROUGE-L is based on longest common subsequence. Song et al. (2018), Zhao et al. (2018), and Scialom et al. (2019) use the same metrics and the same source of reference questions (SQuAD; Rajpurkar et al., 2016).

6 | CONCLUSION

In this paper, we have reviewed various approaches to the challenging task of QG in natural language. As indicated by numerous studies, automated QG tools can contribute to enlarging the training sets of QA systems, generating questions for student evaluation activities such as reading comprehension tests, enhancing conversational systems in digital assistants, and other applications that require natural language understanding. After more than a decade of research on rule-based QG systems, such as Heilman (2011), the work by Du et al. (2017) has initiated a new, neural-based approach, which learns a QG model from a corpus of sentence-question pairs. Unfortunately, at the time of writing this review, we could not found any study performing an objective comparison of these two approaches on independent text corpora, which were neither used for tuning the hand-crafted rules, nor served for training a neural model. The claim made by Du et al. (2017) that their system "significantly outperforms" the state-of-the-art rule-based system (Heilman, 2011) is based on the testing part of the SQuAD corpus, while the other part of *the same dataset* was used by them for training their system. Thus, we believe that their results cannot be generalized to other corpora and domains, particularly those used for developing the competing rule-based systems.

The application of AQG systems to real-world problems still faces multiple challenges. For example, most educational assessment QG systems are developed for generating large sets of test questions and thus their average per-question performance may not be the most important quality measure. Instead, one may compare educational QG systems based on measures like the total number of questions generated from a given text corpus, the percentage of useful questions, the instructor time saved by the system, and so on. In contrast, the skill acquisition QG systems may be evaluated

by their effect on the students' learning outcomes. In case of automatically generating questions for training an interactive QA system, the accuracy improvement of the trained QA system should be evaluated. Thus, we need more explicit evaluation studies, which will focus on specific objectives of a given QG-based system. These new evaluation studies are likely to lead to development of practical QG methods and tools, tailored to the needs of specific applications, rather than to some popular benchmarks like SQuAD.

Last but not least, AQG applications in multilingual and cross-lingual domains need further exploration. Such applications may include generating test questions in multiple, especially resource-poor languages, improving reading and writing skills of students studying a foreign language, and building multilingual QA systems.

ACKNOWLEDGMENT

This research was partially supported by IBM Cyber Security Center of Excellence (CCoE), Beer Sheva, Israel under grant no. 15/12/144.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

Mark Last: Conceptualization; data curation; formal analysis; funding acquisition; methodology; writing-original draft; writing-review and editing. **Guy Danon:** Conceptualization; data curation; formal analysis; methodology; writing-original draft.

ORCID

Mark Last https://orcid.org/0000-0003-0748-7918

RELATED WIREs ARTICLE

Text mining in education

REFERENCES

Afzal, N., & Mitkov, R. (2014). Automatic generation of multiple choice questions using dependency-based semantic relations. *Soft Computing*, *18*, 1269–1281.

Agarwal, M., & Mannem, P. (2011). Automatic gap-fill question generation from text books. In *Proceedings of the 6th workshop on innovative use of NLP for building educational applications* (pp. 56–64). Stroudsburg, PA: Association for Computational Linguistics.

Bednarik, L., & Kovacs, L. (2012). Implementation and assessment of the automatic question generation module. In *2012 IEEE 3rd international conference on cognitive infocommunications (CogInfoCom)* (pp. 687–690). IEEE.

Bhatia, A. S., Kirti, M., & Saha, S. K. (2013). Automatic generation of multiple choice questions using wikipedia. In *International conference on pattern recognition and machine intelligence* (pp. 733–738). Springer.

Bodenreider, O. (2004). The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Research*, *32*, D267–D270.

Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 819–826). Association for Computational Linguistics.

Bunescu, R., & Huang, Y. (2010). Learning the relative usefulness of questions in community QA. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, EMNLP'10 (pp. 97–107). Stroudsburg, PA: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1870658.1870668

Chali, Y., & Hasan, S. A. (2015). Towards topic-to-question generation. *Computational Linguistics*, *41*, 1–20.

Collobert, R. (2011). Deep learning for efficient discriminative parsing. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 224–232).

Du, X., Shao, J., & Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (Vol. 1: Long Papers, pp. 1342–1352). Vancouver, Canada: Association for Computational Linguistics. https://www.aclweb.org/anthology/P17-1123

Fattoh, I. E., Aboutabl, A. E., & Haggag, M. H. (2015). Semantic question generation using artificial immunity. *International Journal of Modern Education and Computer Science*, *7*, 1.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, *31*, 104–137.

Heilman, M. (2011) *Automatic factual question generation from text*. (Ph.D. thesis). Carnegie Mellon University.

Heilman, M., & Smith, N. A. (2009). Question generation via overgenerating transformations and ranking. Tech. rep., Carnegie-Mellon Univ, Pittsburgh, PA, Language Technologies Inst.

Hussein, H., Elmogy, M., & Guirguis, S. (2014). Automatic english question generation system based on template driven scheme. *International Journal of Computer Science Issues (IJCSI)*, *11*, 45.

Kalady, S., Elikkottil, A., & Das, R. (2010) Natural language question generation using syntax and keywords. In *Proceedings of QG2010: The third workshop on question generation* (Vol. 2). questiongeneration.org.

Labutov, I., Basu, S., & Vanderwende, L. (2015). Deep questions without deep understanding. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing* (Vol. 1: Long Papers, pp. 889–898).

Le, N.-T., Kojiri, T., & Pinkwart, N. (2014). Automatic question generation for educational applications – The state of art. In *Advanced computational methods for knowledge engineering* (pp. 325–338). New York, NY: Springer.

Liu, M., Calvo, R. A., & Rus, V. (2010). Automatic question generation for literature review writing support. In *International conference on intelligent tutoring systems* (pp. 45–54). Springer.

Mannem, P., Prasad, R., & Joshi, A. (2010). Question generation from paragraphs at upenn: Qgstec system description. In *Proceedings of QG2010: The third workshop on question generation* (pp. 84–91). Milton Keynes, England: The Open University.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for computational linguistics (ACL) system demonstrations* (pp. 55–60). Stroudsburg, PA: Association for Computational Linguistics (ACL). http://www.aclweb.org/anthology/P/P14/P14-5010

Mazidi, K., & Nielsen, R. D. (2015). Leveraging multiple views of text for automatic question generation. In *International conference on artificial intelligence in education* (pp. 257–266). Springer.

McConnell, C. C., Mannem, P., Prasad, R., & Joshi, A. (2011). A new approach to ranking over-generated questions. In *2011 AAAI fall symposium series*, Palo Alto, CA: Association for the Advancement of Artificial Intelligence (AAAI).

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM*, *38*, 39–41.

Mitkov, R., Le An, H., & Karamanis, N. (2006). A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering*, *12*, 177–194.

Mostow, J., & Chen, W. (2009). Generating instruction automatically for the reading strategy of self-questioning. In *Artificial Intelligence in Education* (pp. 465–472). Amsterdam, Holland: IOS Press.

Olney, A. M., Graesser, A. C., & Person, N. K. (2012). Question generation from concept maps. *Dialogue & Discourse*, *3*, 75–99.

Papasalouros, A., Kanaris, K., & Kotis, K. (2008). Automatic generation of multiple choice questions from domain ontologies. In *e-Learning* (pp. 427–434). Princeton, NJ: Citeseer.

Piwek, P., & Stoyanchev, S. (2010). Generating expository dialogue from monologue: Motivation, corpus and preliminary rules. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics* (pp. 333–336). Association for Computational Linguistics.

Pradhan, S. S., Ward, W., & Martin, J. H. (2008). Towards robust semantic role labeling. *Computational Linguistics*, *34*, 289–310.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *Squad: 100,000+ questions for machine comprehension of text*. arXiv preprint arXiv: 1606.05250.

Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning* (pp. 147–155). Association for Computational Linguistics.

Rus, V., Cai, Z., & Graesser, A. (2008) Question generation: Example of a multi-year evaluation campaign. In *Proc. WS on the QGSTEC*.

Scialom, T., Piwowarski, B., & Staiano, J. (2019). Self-attention architectures for answer-agnostic neural question generation. In *Proceedings of the 57th annual meeting of the Association for Computational Linguistics* (pp. 6027–6032). Florence, Italy: Association for Computational Linguistics. https://www.aclweb.org/anthology/P19-1604

Song, L., Wang, Z., Hamza, W., Zhang, Y., & Gildea, D. (2018). Leveraging context information for natural question generation. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: Human language technologies* (Vol. 2, Short Papers, pp. 569–574). New Orleans, Louisiana: Association for Computational Linguistics. https://www.aclweb.org/anthology/N18-2090

Sumita, E., Sugaya, F., & Yamamoto, S. (2005). Measuring non-native speakers' proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the second workshop on building educational applications using NLP* (pp. 61–68). Association for Computational Linguistics.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., & Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic conference on informatics* (pp. 382–392). Springer.

Varga, A., & Ha, L. A. (2010). Wlv: A question generation system for the qgstec 2010 task b. In *Proceedings of QG2010: The third workshop on question generation* (pp. 80–83). Milton Keynes, England: The Open University.

Wang, W., Hao, T., & Liu, W. (2007). Automatic question generation for learning evaluation in medicine. In *International conference on web-based learning* (pp. 242–251). Springer.

Zhao, Y., Ni, X., Ding, Y., & Ke, Q. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3901–3910). Brussels, Belgium: Association for Computational Linguistics. https://www.aclweb.org/anthology/D18-1424

How to cite this article: Last M, Danon G. Automatic question generation. *WIREs Data Mining Knowl Discov*. 2020;e1382. https://doi.org/10.1002/widm.1382