An Analysis of Using Templates to Generate Questions for Inquiry-Based Learning

Liam Talberg
Department of Computer Science
University of Cape Town
Cape Town, South Africa
tlblia001@myuct.ac.za

ABSTRACT

Template-based question generation is an area of natural language generation that holds much importance to education, with the topic of question generation from lecture transcripts being of particular interest. A large of variety of methods to generate questions using templates exists with these methods building on functionality found syntactic, semantic, and ontology-based systems for content extraction and slot filling. The main problem that needs to be addressed in a template-based approach is how to produce the templates. Earlier template-based systems made use of a limited number of templates, that were pre-loaded into the system, relating to one specific domain, while later systems experimented with methods to allow for domain independence through extracting templates from sample input questions. Overall, it was found that the best methods for template-based question generation incorporate a semantic approach to content extraction, due to its superior ability at discerning the context of sentences in a larger body of text, as well as an automated method for template creation, through removing key phrases from input questions. Both these methods allow for the creation of a domain independent system which is a crucial requirement of a system designed for question generation from lecture transcripts.

KEYWORDS

Natural Language Generation, Question Generation, Template-Based Question Generation

1 Introduction

In an educational setting the act of asking questions is immeasurably important to the learning process [1]. Questions form a fundamental interaction between teacher and student and not only allow for clarification on topics that may not be fully understood but also facilitate the development of critical thinking skills. The benefit of question asking does not end there as the questions that a teacher receives are an extremely useful tool to judge the difficulty level of the content for their class [2]. Unfortunately, with the rise in asynchronous online teaching this crucial task of question asking has become substantially reduced thus if one could automatically generate questions based on the transcript of a lesson it would prove an extremely useful tool for teachers and students alike. By automatically producing questions it would not only allow educators to more easily prepare questions for tests, giving them more time for academic enrichment activities such as hands on projects with their students or discussion based sessions, but would also and enable to students to have an easy mechanism for self-assessment [3].

By making use of Natural Language Generation (NLG) and more specifically the progress made in the field of Question Generation (QG) a tool to produce questions from educational material, such as lecture transcripts, can be developed. However, a review of existing systems of QG needs to be undertaken to determine the core functionality that needs to be included in such a system. This review focusses on template-based QG and highlights its ability to produce high-quality questions from many domains, which is an important requirement given the variety of topics that exist in educational content. It will also provide context with regards to the history and direction of the field of template-based question generation. In achieving these goals, analysis of different systems has been undertaken with a focus on investigating their methods of content extraction, template creation, their ability to achieve domain independence and finally through the actual performance of the systems in both human evaluation and automatic metrics. All these areas are covered in section 3 of this review with section 2 giving a brief overview of NLG as whole.

2 Natural Language Generation

NLG is the task of producing text from some input dataset. It is a subfield of artificial intelligence and is concerned with combining knowledge about language and the application domain to produce explanations, messages or questions [4]. In essence a NLG system needs to determine main criteria: *What to say?* and *How to say it?* [5].

The field of NLG is broad but covers two main categories: data-to-text and text-to-text generation. Although a distinction is made, the lines between these two categories are very blurred as many techniques used in one are also used in the other. This distinction will be explained in subsequent paragraphs but mainly relates to the input into the systems. It must however be noted that the general consensus is that NLG systems should all produce text as their output [6].

In data-to-text generation the input data will be some form of non-linguistic information such as a database or image [7], while in a text-to-text generation system the input will, as the name implies, be text. From these inputs the output of the can be categorized into many different categories with examples including summarization,

distractor generation for multiple choice questions (i.e.: to generate wrong information compared to the input text), question generation [7], report generation or natural language rendering of an ontology. The next section of this review focuses on the output category of question generation highlighting its uses as well as the different approaches to implement a QG system.

3 Question Generation

QG falls under the umbrella of NLG and is, as the name implies, focused on generating questions using input data [8]. This input data could be a larger piece of text or a more structured piece of data such as an ontology. QG is a highly useful tool and has many applications such as: suggesting questions a student may ask while consuming a piece of learning material, producing question that can assess a student's deeper understanding of the content taught to them during a class, outputting questions that may be frequently asked by customers of a business and in assisting professionals by producing questions that could be used in legal or medical contexts [9].

In order to produce questions a few main methods of QG are considered in this review namely; syntactic-based, semantic-based, template-based [10], ontology verbalization and data-driven based systems.

A template-based system is built upon many of the functions used in syntactic, semantic and ontology based QG. Thus, it is important to understand how these systems work before looking at template-based systems.

3.1 Syntactic Question Generation

The syntactic approach to QG took much of the focus in the early years of the field. In a syntactic-based system the text is fed through the system with the complex sentences being simplified while a syntactic parser identifies key phrases. Transformations are then undertaken to convert the sentence into a question, these can include adding a question word [10]. An example of this process can be seen below:

- Sentence: *Jeromy kicked a ball at 10am.*
- Syntactic Phrases Identified: *Jeromy = Subject, Kicked = Verb, Ball = Object, 10am = Time*
- Questions: *When was the ball kicked?, Who kicked the ball?, What did Jeromy kick?*

The first QG system to use a syntactic approach was the AUTOQUEST system [11]. The AUTOQUEST system was an entirely syntactic model and focused on QG from input books and manuals relating to naval training. After input the sentences were parsed and reassembled to form a question [11]. This reassembling involved locating the verb of the sentence. If the verb was auxiliary, such as *"is"* or *"was",* it was simply placed at the front of the sentence to form a question. For non-auxiliary verbs an extra step of analysis was done to determine an appropriate question word to be placed at the front of the sentence [11].

While this system was able to produce questions, it suffered from a variety of errors. The most common being incorrect identification of the verb [11]. This was followed by semantic errors where the wrong question word was selected leading to non-sensical questions being produced. This issue is particularly concerning in the educational domain where incorrect questions can create much discomfort for learners especially in a test setting. The issues of this system were mainly due to the syntactic pattern matcher which was unable to make use of all the syntactic information in the sentence [11]. However, much progress in syntactic pattern matching has been made with newer syntactic systems improving substantially on the AUTOQUEST system.

Research into syntactic models for QG has continued with a recent approach being implemented by Dannon and Last [12] who built on work done by Heilman [13]. Their system aimed to produce a diverse set of factoid questions relating to the specific domain of cyber security, this differed from AUTOQUEST's domain of naval training. Another key difference was their novel use of paraphrasing in which key phrases from the sentences were replaced with synonyms that the system had learnt through pretraining of 1.1million documents from the field [12]. These paraphrased sentenced were then fed into Heilman's [13] question generator to produce questions. The questions produced from the paraphrased sentences were then compared to the questions output from the original sentences. In evaluating their system, using two experts in the field of cyber security, they found that using paraphrasing was able to enhance the quality of 62 out of 148 questions [12]. This improvement shows that the idea of paraphrasing has some validity in producing better quality questions and, as will be seen later in this review, is a concept that can be incorporated in a template-based approach [14].

The syntactic approach to QG is a category that offers a very plausible solution to QG from educational content. However, the reach of syntactic analysis can be limited especially in relation to sentences that may be ambiguous or where the meaning of words is not clear, thus semantic QG needs to be considered as well.

3.2 Semantic Question Generation

In contrast to syntactic-based approaches, semantic-based systems work by determining relationships between objects and their associated actions [10] and understanding the context the phrases are used in. To highlight the difference between the two, consider the sentence: "*Orange is my favourite!"* In syntactic analysis the word *Orange* would simply be taken as the subject however in semantic analysis the context the word is used in will be taken into account as in this case *Orange* could refer to a colour or a fruit, thus, to fully understand the sentence the context of it needs to be considered as well. This method of content analysis and extraction is another one of the ways in which template-based models function and thus the performance of these systems needs to be reviewed.

A recent example of a semantic-based system was developed by Flor and Riordan to produce questions for an educational setting [15]. Their system used semantic role labeling (SRL) to generate both *wh-questions* (i.e.: where, what, why etc) as well as *yes/no* questions. SRL involves using a tool, in this case SENNA [16], to label phrases into various semantic categories such as *agents (subjects), locations, directions, time* or *manner.*

Using SRL, generating yes/no questions involved moving the verb to the front of the sentence to form a question. An example of this is converting the sentence *"He ate quickly"* to the question *"Did he eat quickly?*" [15] To produce *wh-questions* the semantic role assigned to phrase dictates what question word is most appropriate [15]. As an example, if a phrase has been given a role referring to *location* then the question word *where* is most appropriate, while if it has been given the role *time* the question word *when* is best suited. This use of SRL to determine the correct question word is something that could be expanded in a template-based approach to select the correct template.

Their SRL based system was evaluated by using the input of three initial paragraphs from Wikipedia articles and two short articles from education websites [15]. The three criteria evaluated were: grammar, semantics and relevance which were scored out of 5 by two linguistic experts. Their model was also compared to a neural model developed by Du et al [17]. In their findings the SRL system scored exceptionally well in the areas of grammar and semantics averaging 4.33 for *yes/no* type questions and 3.84 for *wh-questions* compared to the neural systems average of 3.18. Although when looking at relevance the SRL system averaged just 2.75 [15]. While useful on its own the use of semantic role labeling combined with templates is what is of particular interest in this review.

The papers discussed in this section highlight how both syntactic and sematic approaches to QG are viable in producing good quality questions but they both suffer drawbacks in terms of the diversity of questions they can produce as well whether the questions actually have any benefit to be used in an educational setting. The template-based approach, to be discussed in the next section of this report, has some key advantages over the methods discussed thus far. One of which is its ability to produce questions not so tightly linked to the source text (i.e.: not simply reordering input sentences to produce questions) [8]. In relation to educational content this has the key benefit of allowing broader thought-provoking questions to be produced as opposed to the purely factoid nature of questions produced by syntactic and semantic systems. This can be seen in the example below [8]:

- Sentence*: As recently as 12,500 years ago, the Earth was in the midst of a glacial age referred to as the Last Ice Age*
- Semantic/Syntactic Question: *When was the last ice age?*
- Template Questions: *How would you describe the Last Ice Age?, Summarize the influence of a glacial age on the environment*.

To produce these thought provoking questions template-based systems, as mentioned earlier, are built upon many of the functions used in semantic and syntactic systems primarily for content extraction and slot filling. In addition, other methods such as ontology or neural methods also exist. All of these functions will be discussed in the next section focusing on Template-Based QG.

3.3 Template-Based Question Generation

QG consists of two main steps: content selection and question construction [18]. In a template-based approach a set template, with placeholder values, for a question is pre-defined. This differs from the approaches used in syntactic and semantic QG where the questions are formed through simply restructuring the input sentences. This is an important feature of template-based systems and allows for more thoughtful questions to be produced but has the extra overhead of requiring the templates to be created prior to question generation.

In the template-based approach after analysis of the input data the placeholder values in the template are replaced with words/phrases extracted from the input to form the questions [8]. An example of this is:

- Template: *What did <subject> <verb>?*
- Sentence: *Simon kicked the ball..*
- Question: *What did Simon kick?*

There are many different ways to select the words/phrases with different systems implementing syntactic, semantic, ontology-based, and neural approaches, examples of which will be discussed later in this section.

Although templates allow for good quality questions to be produced, in terms of grammar and content they do have some key cons. Two of these cons are the limited domain of each template [18] as well as the time-consuming process to construct the templates if done manually. However, ways to overcome these issues do exist and will be explored in the in the sections to come.

3.3.1 Datasets and Content Extraction

In the analysis of template-based systems an important consideration is the type of dataset from which the systems have been designed to produce questions. This dataset could purely text based or more structured such as an ontology. The methods in which information from this dataset is extracted, to fill the template slots, is another area of great interest with methods using syntactic, semantic, and neural approaches all being implemented.

Some of the first systems to make use of templates for QG were Mostow and Chen [19], who developed a system to generate questions from narrative fictional passages with the aim of improving comprehension in children, Chen and Aist [20] who made use of informational text as their dataset and Staenscu et al [21] who created a tool into which passages of text, with a particular focus on educational content, could be input.

To extract the content from these input datasets the above systems all made use of syntactic analysis. This process can be seen in the work done by Mostow and Chen [19] where syntactic parsing is used to extract key details, in this case, *character names, verbs*, and *participles* from the input text. Their system paid particular attention to verbs and used them as a guide as to when questions should be generated (I.E: a verb such as *surprise* or *decide* indicates a change in a character's belief indicating a question relating to this should be generated) [19]. Similarly, Chen and Aist extracted many of the same syntactic patterns but due to their system focusing on informational text particular attention was paid to temporal, condition and linguistic modularity phrases [20]. Regarding temporal phrases, sequences such as *"for several days"* or expressions like "*While he watched TV"* were extracted. In terms of conditional phrases, they analyzed the text for phrases beginning

with words such as *'If"*, *"even if"* or *"as long as".* Finally, in looking for linguistic modularity auxiliary verbs such as "*should*", "*must*" and "*could*" were considered [20].

The system developed by Wijanarko et al [22] is another which used a syntactic approach to content extaction. They made use of the Binus Online Learning repository which contained syllabuses, lecture materials and student-lecturer discussion logs relating to an undergraduate engineering course [22]. Their system focused on key-phrase selection (noun selection) to correctly determine the main topics of the input documents which was then combined Blooms Taxonomy [23] to produce questions. The principles behind using Blooms' Taxonomy for question construction will be explained in next section of this review. This use of syntactic extraction is a method that be of use for extraction of content from lecture transcripts, however another method is through the use of semantic analysis.

In contrast to the systems above, another method of content extraction is through semantic analysis. Systems such as Lindberg [8], Hussein, Elmogy & Guirguis [1] and Berant and Liang [14] all make use of this method. Lindberg's system made use of a dataset containing a grade 10 science curriculum focusing on climate change and global warming. Berant and Liang's dataset was question answers pairs of commonly asked questions on the web (WebQuestions) as well as a dataset of questions that had been manually annotated [14]. The dataset of Hussein, Elmogy & Guirguis [1] was similar to that of Staenscu et al [21] as their systems was developed as tool into which text passages could be input and questions output.

Even through these varying datasets the key ideas of semantic analysis remained and semantic patterns such as: *A0, A1, … , An* to refer to the agent to which the verb the attached, *V* to indicate the verb, *AM-TMP* for the temporal phrase, *AM-LOC* for the location phrase and *AM-MNR* for the manner phrase were extracted. The CoNLL SRL shared task naming convention has been used here [24]. A key benefit of semantic, as opposed to syntactic, extraction is the ability of more interesting parts of the sentence which span over many different syntactic patterns to be identified. This benefit is highly applicable to the educational domain and could allow for questions that cover multiple sentences to be generated. An example of this type of semantic analysis can be seen in figure 1.

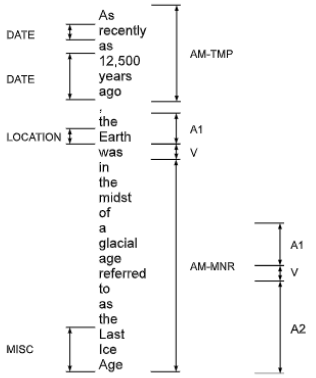

Figure 1: A Sentence and its Semantic Patterns [8]

A third method for content extraction involves using a neural system. A system which uses this approach is that of Gu, Yang and Wei [25]. As input into their system, they received question answer pairs. Through this input they sought to use a neural system extract the content from the question thereby separating the question into a template and content based components [25]. Their neural method for content extraction used a Seq2Seq neural network. Seq2Seq models work by taking in a sequence of words as input, in this case the question. They then output another sequence of words. This output is an encoded intermediate representation which can then be decoded through another neural network to get the final output which for Gu, Yuqiao and Wei's system was the content and template parts of the question. [18]. A large benefit of this approach to content extraction is that it can be trained to work with many different domains however the need for it to be trained and the extra time that training requires is the downside of using a neural approach.

The final method considered in this review for content extraction involved the use of ontologies. An ontology is defined as conceptualization of a an explicit specification [26]. In the case of question generation an ontology is an application-independent representation of a specific subject domain [27] . Two papers which make use of ontologies Teo and Joy [28] who used an ontology about operating systems and Kusuma, Siahaan and Fatichah [29] who used an ontology of natural science and animals. Given the structured nature of ontologies content extraction mainly involves using the relationships between concepts to extract the correct concepts, properties and accompanying relationships. These can then be input in the templates to form questions. In the example of lecture transcripts, this method of content extraction would involve an extra step of forming the ontology. Thus, the time taken to complete this extra step needs to be consideration in relation to any performance benefit that be exist. This will be considered in later sections of the report.

All these means of content extraction lead to the production of phrases that need to be inserted into templates. The next key consideration of a template-based systems is how these templates are created. This is the process that will be reviewed in the following section.

3.3.2 Template Construction and Types of Questions

The next, and perhaps most important, component of a template-based systems is how the actual templates are constructed. This step can be the most time consuming and also dictates the types of questions that the system is able to produce. Another consideration is that in the educational domain a system which can produce a large variety of questions is particularly important. Methods to construct templates include using manually predefined templates, allowing users to add new templates as needed and automatically generating questions from sample input questions. In a system with predefined templates the variety of questions able to be produced is undoubtedly going to be limited, allowing users to add templates can eliminate some of this however being able to automatically generate templates from input questions is the best way to ensure adequate question variety and domain independence. Each of these methods will be discussed below.

In systems which make use of predefined templates the types of questions they are able to produce are very limited. Systems like this include Mostow and Chen [19] and Chen and Aist [20]. Mostow and Chen's system was able to produce just three types of questions using the following templates [19]:

1. "*What did <character> <verb>?"*
2. *"Why/How did <character> <verb> <complement>?"*
3. *"Why was/were <character> <past-particle>?"*

As can be seen these templates are very limited in terms of the type of questions they cover, which limits its use in an educational setting. Chen and Aist marginally improved in this are making use of four templates covering the three categories [20].

1) Conditional Context:
   a) *"What would happen if <x>?"*
2) Temporal Context:
   a) *"When would <x>?"*
   b) *"What happens <temporal-expression>?"*
3) Linguistic Modularity:
   a) *"Why <auxiliary-verb> <x>?"*

In the above templates *<x>* is a term created semantically using the subject of the sentence combined with the verb and any theme that may exist depending on the context.

Other systems which used this method of predefined templates, with some different implementations, include Hussein, Elmogy, and Guirguis [1] and Lindberg [8]. On one hand, Hussein, Elmogy, and Guirguis used a large database of templates covering many different question types (*wh-questions*, *yes/no* etc..). They made use of analysis prior to generation to gain knowledge about keywords relating to location, organizations, people, money and time [1]. This knowledge was then used to select appropriate types of questions/templates prior to generation. The use of a large database of templates coupled with the analysis before generation is something that could be of use in an educational setting as it would allow for multiple groups of templates relating to different domains to exist and be accessed accordingly.

On the other hand, Lindberg's approach did not contain a large database of templates but rather allowed for his templates to have slots outside of the question string (I.E: not included in the actual output of the question). This allowed for extra pattern matching criteria to be undertaken to produce higher quality sentences [8]. An example of this is *"What is one consequence of <A0>? ## <A1>"* where *<A1>* exists outside the bounds of the question string [8]. In saying this, his templates were still premade and part of the system prior to generation. By allowing slots outside of the question string it also meant that only relevant extracted phrases needed to be included in each question.

While using predefined templates can produce questions the diversity of questions produced is extremely limited and as a result is not particularly useful to a system that would take in a wide variety of lecture transcripts as input. Thus, one needs to have a system in which templates can either be added, as is the case with Staenscu et al [21], or ideally a system that can create templates using input questions. This creation of templates can be undertaken through different mechanisms. The mechanisms looked at here are manually extracting the templates, key phrase identification and neural systems.

A system that used a manual from of template extraction is Kusuma, Siahaan & Fatichah [29]. Their templates were constructed through collating domain specific questions from textbooks. These questions were then manually categorized by an expert in that domain and then again by an ontology engineer to analyze the components of the ontology before converting the question into a template [29].

In contrast Teo and Joy [28] saught to semi-automate this process of template creation to produce description and comparion type questions. Simiarly to the process desribed above, specific questions relating to the domain were collected and classified into categories. From there the duplicate questions were removed and the keyphrases in the questions were replaced with placeholder tokens (X, Y, etc..) which could then be replaced by ontology concepts to produce questions [28]. The question template was then further refined by removing the question phrase and replacing it with the acronym *QWC* to produce the formalized template. An example of this process is [28]:

Source Question: *What is the difference between non-pre-emptive and pre-emptive?*

Template Created*: What is the difference between X and Y?*

Formalized Template*: QWC X and Y?*

Key phrase selection was also used in the system developed by Wijanarko et al [22] where it was combined with Blooms Taxonomy [23] to produce questions at various difficulty levels. These levels cover questions asking facts (remembering), questions asking casual relations (comprehension), questions requiring application, questions requiring analysis and finally questions requiring evaluation [22]. Once the key phrases had been selected they were paired with an appropriate verb based on Blooms Taxonomy and the level of questioning required. For simpler questions verbs such as *"define", "identify", "compare"* or "*solve*" were used. For more thought-provoking questions words such as *"contrast", "criticize"* or "*design*" were selected. This use of set verb-based-question-words to construct the questions makes it particularly suited for inquiry-based learning.

In a differing approach, Gu, Yang, Wei's [25] used a Seq2Seq neural network to extract templates. After the content had been extracted from the input questions, a method that was explained in the above section, the remaining text would form the template. These automatic methods of template creation are no doubt the best suited to the educational domain as they facilitate not only a large amount of variety in questions but also reduce the time needed manually create the templates. In addition, they also facilitate in domain independence which will be explored in the section below.

3.3.3 Question Domain and Slot Filling

When creating a system to generate questions from lecture transcripts a crucial issue is that of domain. The transcripts input into the system could come from a variety of different subjects and thus the system needs to be domain independent. In this review the

term domain independent will be used to refer to systems which, due to their content extraction and template creation methods, are able to produce questions from many different domains. An overview of the systems and whether they are domain independent can be seen in table 1.

System Name	*Domain Independent?*	*How is Domain Independence Achieved?*	*Slot Types*
Mostow and Chen [19]	No	-	Specific to the domain of narrative text
Chen and Aist [20]	No	-	Specific to the domain of informational text
Staenscu et al [21]	Partial	User can add templates	User Specified
Hussein et al [1]	Partial	Large database of templates	Varied
Wijanarko et al [22]	Yes	Blooms Taxonomy	-
Berant and Liang [14]	Yes	Logical forms and Knowledge Base	Logical statements (to be filled from knowledge base)
Kusuma et al [29]	Yes	Ontology content extraction	Ontology concepts, properties, and relationships
Gu et al [25]	Yes	Neural template and content extraction	Extracted from sample input questions

Table 1: Overview of Systems highlighting their Question Domain and Slot Types

Of the systems looked at in this review the systems developed by Mostow and Chen [19] and Chen and Aist [20] both do not achieve domain independence. This is due to them both having a very limited quantity of templates to produce questions from. In addition, the slots in their templates are very linked to the content their systems were designed to receive as input with slots like <*character*>[19] or <*temporal-expression*> [20] being used which may not be applicable to other types of input text.

A slight improvement in terms of domain independence exists in the systems developed by by Staenscu et al [21] and Hussein, Elmogy, and Guirguis [1]. In the case of Staenscu et al [21] their tool for QG allowed users to create and add their own templates to the system. This process was enabled by tagging questions into categories such as *define*, *what is*, *discuss* or *state* and from there allowing the user to construct templates in these categories. The syntax of a "*#"* was used as a placeholder for the word/phrase to be replaced. An example of a template in this system is *"What is a/an #"* [21]. In doing this it allowed users create templates that could match the domain of the input text better, thus at least partially achieving domain independence. Hussein, Elmogy, and Guirguis [1] also achieved partial domain independence through the use of a large database of templates with many different slot types. While not perfect, a method like this does allow for better questions to be generated when compared to a system that would has just a limited number of templates, particularly in the case of a system receiving input different from its intended domain.

Partial domain independence is undoubtedly an improvement, but the gold standard are systems that can achieve full domain independence. Examples of systems that achieve this include; Teo and Joy [28], Kusuma, Siahaan and Fatichah [29], Berant and Linag [14], Gu, Yang, Wei [25] as well as Wijanarko et al [22].

Both Kusuma, Siahaan and Fatichah [29] and Teo and Joy [28] use ontologies to achieve domain independence. As explained earlier, ontologies are structured and thus by making use of slots in the templates relating to this structure it allows the templates to be filled from ontologies covering different domains. An example of this is the template: "*What does a <class> <property>?"* which could produce the question "*What does a wolf eat?"* or with a different domain could produce the question "*What does a car burn?"*. While interesting, a downside of the use of ontologies is that with an input of lecture transcripts a separate tool would be needed to construct the ontology before template-based QG could take place.

Berant and Linag [14] achieved domain independence through the use of logical forms accompanied with a large knowledge base and paraphrasing. The knowledge base would contain information such as: *(BillGates, PlaceOfBirth,, Seattle)* to say that Bill Gates was born in Seattle. The knowledge base could then be queried using a logical statement like *(PlaceOfBirth.Seattle ∩ Founded.Microsoft)* to ask for Microsoft founders who were born in Seattle [14]. These logical forms also formed the basis of their templates with an expression like *"p.(p1.e1 ∩ p2.e2)"* linking to the logical expression "*Character.(Actor.BradPitt ∩ Film.Troy)"* where *"e"* represents an entity and *"p"* a property of that entity. Using this same representation question templates like *"WH d(t) is the NP of d(e)?"* which could produce the question *"What location is the place of birth of Elvis Presley".*[14]. In the template the characters *WH* refer to the question word, *NP* to the noun phrase and *d(e)* to the description of an entity. After generation an association model was then used to link similar questions from the datasets so that paraphrasing could take place. As an example, questions like *"What type of music did Richard Wagner Play?"* and *"What is the musical genres of Richard Wagner?"*, would be associated through the phrases *"type of music"* and *"musical genres"* [14]. As can be seen the slots in the templates are extremely general, this

accompanied with the knowledge base allows for the systems to be domain independent.

In both Wijanarko et al [22] and Gu, Yuqiao & Wei [25] approaches the very nature of their systems mean domain independence can be ensured. In the case of Wijanarko et al, using Bloom's taxonomy to create their templates means simply using the verbs outlined in the previous section. In doing this templates such as *"Discuss <x>"* or "*Compare <x> and <y>"* are produced. Just by looking at these templates we can see that no matter the domain a valid question can still be produced. The full architecture of this system can be seen in figure 2. Gu, Yuqiao & Wei's [25] approach to QG through a neural network to extract both the question content and template mean that the system is domain independent. In addition, they also made use of paraphrasing system in which after template generation a retrieval search of the template dataset was undertaken to find the most similar template based on cosine-similarity. Once selected this template was then filled with the extracted content [25]. The full process can be seen in figure 3. By using a neural network that can adapt and learn it means that no matter the domain of the content if given a sufficient quantity of input the system will be able to learn to output both the content and template correctly.

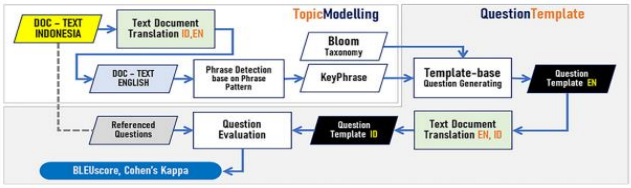

Figure 2: Architecture and Data Flow of the System [22]

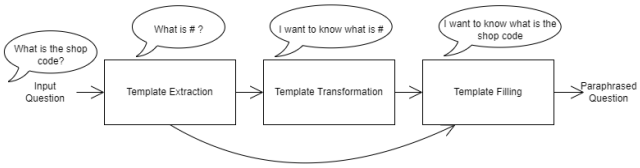

Figure 3: Example of an Input Question being Paraphrased [25]

As can be seen, creating a system that is domain independent needs to be a core principle of template-based QG for inquiry-based learning. Another way to view the importance of this characteristic is through evaluating the performance of the various systems discussed in this review.

3.3.4 Evaluation of Template-Based Systems

To evaluate the performance of template-based systems the main metrics, used by the systems in this review, were human evaluation, particularly for grammar and logical sense, as well the automatic metric BiLingual Evaluation Understudy (BLEU). A BLEU score is a number from 0 to 1 that measures the similarity of machine produced text to that of a high-quality human creation. An important consideration that needs to be noted is that performance of systems under human evaluation is of much more importance than that its performance in automatic metrics. This is because while automatic metrics can determine if the questions produced are similar to human produced questions, human evaluation is able to also consider the value of the question in relation to the content it was produced from.

As can be interpreted the from the previous sections of this review the early systems of Mostow and Chen [19] and Chen and Aist [20] did not perform well. These systems were both evaluated by human evaluators judging for grammatical correctness and the whether the questions made logical sense. In Mostow and Chen's system just 35.6% of questions produced were deemed of acceptable quality. [19]. Chen and Aist's system using different categories of questions was able to improve on this with conditional and linguistic modularity-based questions being produced successfully 87% of the time. However temporal-context questions were output successfully just 65.9% of the time. In both these systems parsing errors lead to incorrect labelling of phrases and thus questions were found to be counterfactual and had grammatical errors. This shows the weakness of a purely syntactic approach to content extraction and thus a semantic, approach whose results are explored in the next paragraph, should instead be considered.

Moving onto systems which made use of semantic extraction, Lindberg's system was able to produce grammatically correct questions 85% of the time, with 66% of these making logical sense, when judged by human evaluators [10]. Similarly, Berant and Liang's [14] semantic extraction system was able to improve 12% over the reference performance for the WebQuestions dataset. These systems both highlight the strengths of semantic content extraction, especially when compared to the syntactic systems discussed above. One key downside Lindberg's system [10] was that just 17% of the questions produced had value in relation to the curriculum. This is a major issue in the case of QG from lecture transcripts but was found to be caused by issues pertaining to the slot filling and not from the semantic extraction. Another note on Lindberg's system is that just 20% of the questions were answerable from information found within in the input document [10]. This lack of answers in the input text is not necessarily a bad feature, as it could encourage critical thinking among learners and once again was likely caused by the slot filling mechanisms. In a similar manner, Hussein, Elmogy and Guirguis's system suffered from outputting grammatically incorrect questions, mostly due to incorrect tenses, but when compared to Lindberg had the benefit, in terms of QG for an educational setting, of all answers to the questions being found within the text [1]. In the context of this project the use of semantic analysis does seem to be the most promising method of content extraction. Another area of interest in this review was looking at ontologies, the results of which can be seen below.

When looking at systems which made use of ontologies, Kusuma, Siahaan and Fatichah's system was able to produce questions with an accuracy of over 90% while having a large varierty in terms of question category [29]. This question variety, accuracy and domain indepence is a key reason why ontologies can prove very useful in the case of template-based QG for of inquiry based learning. However, the scope of this project can not make use of them unless a sepreate tool to transform the lecture transcripts into ontologies were to be created.

The two systems whih made use of the BLEU metric were Gu, Yuqiao and Wei [25] and Wijanarko et al [22]. Wijanarko's system

achieved an average BLEU score of 0.891 meaning the output questions did indeed have a very close resemblance to the questions produced by experts. In addition, when comparing the questions to the specified Blooms Taxonomy level (i.e.: which verb was selected to form the questions) the system achieved a score of 0.99. This means that questions targeting a specific difficulty level almost always met that target, a key requirement in an educational setting. Gu, Yuqiao and Wei's neural system achieved a BLEU score of 0.902 which indicates an even higher degree of similarity between the questions produced to that a human performing the same task would produce [25]. This highlights the excellent performance of neural systems on automatic metrics, although in order to truly gauge the quality of these questions human evaluation would have to be undertaken.

Through analysis of these results, looking specifically through the lens of QG from lecture transcripts the best functionality from the above systems must be selected and combined. This is what will be explored in the discussion below.

4 Discussion

In reviewing template-based QG, with the focus of developing a system to produce questions from lecture transcripts, the crucial steps to create performant systems are the method of analysis of the input text to extract the content, method to create the templates, and achieving domain independence.

In saying this the systems developed by Mostow and Chen [19], and Chen and Aist [20], while limited in their domain and template construction, show how syntactic analysis can be used to extract content from input passage and produce questions. Syntactic analysis is thus an entirely acceptable method of content extraction however the superior results obtained through semantic analysis suggest it is a better method of extraction for the input content. This can be seen through the better performance of the systems developed by Lindberg [8], Hussein, Elmogy and Guirguis [1] as well as Berant and Liang [14].

In order to construct templates, systems which are able to extract templates from sample input questions have proved to be the most versatile with the work done by Teo and Joy [28] and Gu, Yuqiao and Wei [25] showing that this process can in fact be automated and produce good results. This is especially noticeable when compared to the systems developed by Mostow and Chen [19], Chen and Aist [20] who made use of a very small number of manually created templates which lead to lack of diversity in questions types, which as mentioned before is an issue in the domain of education. Given the scope of the project, and the fact the template-based system will be compared to a neural system, the neural approach taken by Gu, Yuqiao and Wei [25] cannot be undertaken thus the best approach is Teo and Joy's method of removing key phrases from input questions to produce templates.

Moreover, when aiming to achieve domain independence, a partial solution seems be having a large database of templates to begin with or allowing for the addition of templates into the system. This is the approach taken by Hussein, Elmogy and Guirguis [1] and Stanescu et al [21]. Yet, the use of ontologies accompanied with templates appears to achieve more complete domain independence This is witnessed in the systems developed by Teo and Joy [28] and Kusuma, Siahaan and Fatichah [29] who showed how templates with "slots" into which ontology concepts could be inserted, based on their relations, are capable of producing high quality domain independent questions. This success of domain independence is also mirrored by Gu, Yuqiao and Wei's neural method of template and content extraction [25]. The system developed by Wijanarko et al [22] is another that, by using Blooms Taxonomy, achieves domain independence. However, it is unable to produce *wh-questions* and thus its use can be limited. Given that the project takes in lecture transcripts as input (i.e.: not ontologies), and cannot make use of neural systems, the best approach to achieve domain independence is to make use of a similar system to Wijanarko et al combined with a comprehensive system to extract templates from input questions. This would allow for the inclusion of both *wh-questions* and questions derived from Blooms Taxonomy while achieving domain independence. A summary of the key methodologies and to which paper they were taken can be seen in the following paragraph.

The approach taken to generate questions from lecture transcripts will make use of semantic analysis, most similar to Lindberg [8] accompanied with a system to extract templates from existing question, based on Teo and Joy's [28] work, in conjunction with the domain independent system produced by Wijanarko et al [22]. This will allow for a model that can produce high quality questions with substantial variety in terms of both question type and difficulty.

5 Conclusions

This review has given an outline of the field of QG by first introducing syntactic and semantic approaches, both of which are used in the template-based approach which then formed the bulk of this review. While focusing on template-based approaches the main mechanisms of content extraction, template formation, whether the systems are domain independent as well as their performance were evaluated.

Through this process it was found that the use of semantic content extraction over syntactic yields better performance. This is due to its ability to better understand the context of a sentence and its phrases before labelling and extracting them from the input content. Another criteria identified was the need for the system to be domain independent which can be achieved through automatically extracting templates from sample input questions in conjunction with making use of Blooms taxonomy. These features of template-based QG are thus the crucial aspects that must be combined to create a tool to produce high quality questions from lecture transcripts.

REFERENCES

[1] Hussein, H., Elmogy, M. and Guirguis, S. Automatic english question generation system based on template driven scheme. *International Journal of Computer Science Issues (IJCSI)*, 11, 6 (2014), 45

[2] Ros, K., Jong, M., Chan, C. H. and Zhai, C. *Generation of Student Questions for Inquiry-based Learning*, 2022.

[3] Kurdi, G., Leo, J., Parsia, B., Sattler, U. and Al-Emari, S. A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education*, 30, 1 (2020/03/01 2020). DOI: https://doi.org/10.1007/s40593-019-00186-y.
[4] Reiter, E. and Dale, R. Building applied natural language generation systems. *Natural Language Engineering*, 3, 1 (1997),57-87.DOI: https://doi.org/10.1017/S1351324997001502
[5] Thompson, H. S. Strategy and tactics: A model for language production. In *Proceedings of the 13th Regional Meeting of the Chicago Linguistics Society* (Chicago, 1977).
[6] McDonald, D. D. Issues in the Choice of a Source for Natural Language Generation. *Computational Linguistics*, 19, 1 (1993), 191-197. https://aclanthology.org/J93-1009.pdf.
[7] Dong, C., Li, Y., Gong, H., Chen, M., Li, J., Shen, Y. and Yang, M. A survey of natural language generation. *ACM Computing Surveys*, 55, 8 (2022), 1-38. DOI: https://doi.org/10.1145/3554727
[8] Lindberg, D. *Automatic question generation from text for self-directed learning*. Simon Fraser University, 2010 https://summit.sfu.ca/item/12985.
[9] Rus, V., D'Mello, S., Hu, X. and Graesser, A. C. Recent Advances in Conversational Intelligent Tutoring Systems. *AI Magazine*, 34, 3 (2013) DOI: https://doi.org/10.1609/aimag.v34i3.2485
[10] Lindberg, D. *Generating Natural Language Questions to Support Learning On-Line*, 2013.
[11] Wolfe, J. H. *Automatic question generation from text - an aid to independent study*, 1976.
[12] Danon, G. and Last, M. A syntactic approach to domain-specific automatic question generation. *arXiv preprint arXiv:1712.09827* (2017)
[13] Heilman, M. *Automatic factual question generation from text*. Carnegie Mellon University, 2011
[14] Berant, J. and Liang, P. *Semantic parsing via paraphrasing*, 2014.
[15] Flor, M. and Riordan, B. *A semantic role-based approach to open-domain automatic question generation*. City, 2018.
[16] Barnickel, T., Weston, J., Collobert, R., Mewes, H.-W. and Stümpflen, V. Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts. *PloS one*, 4, 7 (2009), e6393
[17] Du, X., Shao, J. and Cardie, C. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106* (2017)
[18] Zhang, R., Guo, J., Chen, L., Fan, Y. and Cheng, X. A Review on Question Generation from Natural Language Text. *ACM Transactions on Information Systems*, 40 (2021). DOI: https://doi.org/10.1145/3468889
[19] Mostow, J. and Chen, W. *Generating Instruction Automatically for the Reading Strategy of Self-Questioning*. City, 2009.
[20] Chen, W. and Aist, G. *Generating Questions Automatically from Informational Text*. Carnegie Mellon University, 2009 http://www.cs.cmu.edu/~listen/pdfs/QG2009-Wei-informational-text-questions-final.pdf.
[21] Stanescu, L., Spahiu, C. S., Ion, A. and Spahiu, A. *Question generation for learning evaluation*. IEEE, 2008.
[22] Wijanarko, B. D., Heryadi, Y., Toba, H. and Budiharto, W. Question generation model based on key-phrase, context-free grammar, and Bloom's taxonomy. *Education and Information Technologies*, 26, 2 (2021/03/01 2021), 2207-2223. DOI: https://doi.org/10.1007/s10639-020-10356-4
[23] Bloom, B. S. *Taxonomy of educational objectives: Handbook 1*. Longman, 1956.
[24] Carreras, X. and Marquez, L. *Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling*, 2005.
[25] Gu, Y., Yuqiao, Y. and Wei, Z. *Extract, transform and filling: A pipeline model for question paraphrasing based on templates*, 2019.
[26] Guarino, N., Oberle, D. and Staab, S. What is an ontology? *Handbook on ontologies* (2009), 1-17
[27] Keet, M. *An introduction to ontology engineering*. Maria Keet Cape Town, South Africa, 2018 http://hdl.handle.net/11427/28312.
[28] Teo, N. H. I. and Joy, M. S. Categorized Question Template Generation for Ontology-Based Assessment Questions. *Int. J. Knowl. Eng*, 4, 2 (2018), 72-75
[29] Kusuma, S. F., Siahaan, D. O. and Fatichah, C. Automatic question generation with various difficulty levels based on knowledge ontology using a query template. *Knowledge-Based Systems*, 249 (2022), 108906