
UNIVERSITY OF CAPE TOWN


DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Project Final Paper 2022

Title: Investigating Rule and Template-Based Methods for Automatic Question Generation from Lecture Transcripts

Author: Liam Talberg

Project Abbreviation: StuQuestions

Supervisor(s): Zola Mahlaza

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	5
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	15
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	

Investigating Rule and Template-Based Methods for Automatic Question Generation from Lecture Transcripts

Liam Talberg
University of Cape Town
Rondebosch, Cape Town
liamtalberg@gmail.com

ABSTRACT

Within the realm of education, question generation (QG), specifically from lecture transcripts, offers a powerful means to elevate comprehension and cultivate critical thinking abilities. Such a system holds the potential to enhance the learning outcomes for both students and teachers. The only system thus far to investigate QG from lecture transcripts is Ros et al. who made use of a neural question generation model. Exploration of simpler semantic and template-based models is thus essential as these systems are less resource intensive and have been shown to yield high-quality outputs in different domains. To construct both these systems, Semantic Role belling (SRL) is heavily utilised for content extraction, template creation and template filling. The system of template creation is the main differentiating factor of the two systems. The semantic system makes use of eight manually created rules while the template-based system combines SRL and coded logic to automatically extract 81 templates from sample input questions. These templates are then filled to produce novel questions. During evaluation through a survey, the questions generated by both systems were well received. Both systems achieved satisfactory scores (i.e.: an average of over 3.5 out of 5 in all categories), with no statistically significant difference in quality detected. This underscores the potential strength of the fully automatic template extraction module to alleviate the traditionally tedious process of template creation. Furthermore, the template-based system managed to outperform a neural system in terms of question quality, though it was notably surpassed by a neural system trained on a more extensive dataset. These findings indicate the potential for further refinement of the automatic template-based system to enhance its performance.

KEYWORDS

Natural Language Generation, Question Generation, Template-based Question Generation, Semantic Question Generation

1 INTRODUCTION

Natural language generation (NLG) is the task of producing text from some input dataset. It forms a subset of artificial intelligence (AI) and aims to combine an understanding of language and the application domain to produce explanations, messages or questions [17]. In the case of this paper, question generation (QG) systems are of particular interest. The field of QG is well explored, with a variety of systems successfully producing questions from many forms of input. Such systems have several important applications. A few of them include: outputting questions to assist professionals in medical or legal contexts, generating frequently asked questions for customer service, and, in the context of this paper, producing questions from lecture transcripts to benefit both teachers and students [19].

QG systems differ with respect to how they extract content and form questions. Different types of QG systems include: semantic, template, and more recently neural systems. This study focuses on a semantic system that utilises manually created rules and a template-based system that employs automatic template extraction. Both systems generate questions from the open source dataset of lecture transcripts utilised by Ros el at [18]. Through these systems, three main research goals are investigated. Firstly, evaluating the output quality of the generated questions in terms of grammatical correctness, logical sense and relevance. Secondly, determining if any statistically significant differences in question quality exist between the two systems. Thirdly, assessing the coverage of the two systems over the dataset. These aims will assess whether the fully automatic template creation methodology employed can alleviate the tedious process of rule and template creation, without a loss in question quality.

In addition to these two systems a complementary paper, authored by Adam Vere [22], delves into neural systems operating on the identical dataset. These systems will be used as a baseline to investigate whether the extra computation required to train a neural model yields any substantial improvements in question quality over rule and template-based approaches. These results could potentially form the preliminary steps in revealing if the extra time involved with training neural models could better be utilised refining methods for automatic template extraction and filtering.

Delving into the QG systems of interest, semantic QG systems operate by establishing connections between objects and their corresponding actions [10]. The connections enable the labelling of phrases according to specific roles, based on the predicates (i.e.: verbal phrases) identified in the sentence. Consequently, a phrase can assume different semantic roles in relation to different predicates in a sentence. Notable semantic tags and their meanings, following the PropBank notation, are outlined in Table 1. This process is called Semantic Role Labelling (SRL). After SRL, the system matches the identified semantic tags against manually created rules. These rules define the semantic tag requirements that a sentence must fulfil so that it can be re-ordered to form a question. Upon meeting these criteria, the system extracts the necessary phrases to generate a question.

In contrast to rule matching, the template-based system makes use of pre-defined templates, with placeholder values that can be populated by phrases to produce questions. In this paper, these placeholder values correspond to the semantic tags seen in Table

Table 1: Semantic Tags and their Meanings

Semantic Tag	Tag Expansion	Explanation
ARG0	Grammatical Subject	The subject of the sentence (e.g.: "The boy")
ARG1	Grammatical Object	The object of the sentence (e.g.: "the ball")
ARGM-LOC	Location	A phrase about a location (e.g.: "in the stadium")
ARGM-TMP	Time	Specifies a time phrase (e.g.: "yesterday")
ARGM-PRP	Purpose	Explains the purpose behind the action (e.g.: "to score a goal")
ARGM-CAU	Cause	Explains the cause of an event (e.g.: "by kicking the ball")
ARGM-ADV	Adverbial	Adds information to the verb (e.g.: "quickly")
ARGM-MOD	Modal	Auxiliary verbs (e.g.: "could", "would")
ARGM-EXT	Extent	Indicates the extent or degree (e.g.: "very much")
ARGM-NEG	Negation	The negative of an action (e.g: "not")
ARGM-GOL	Goal	Specifies the goal of an action (e.g.: "to win")

1. By using these pre-defined templates, instead of sentence reordering, template-based systems are able to produce more thoughtful questions, not so tightly linked to the source material [10]. However, this comes with the extra overhead of requiring template creation prior to generation. To remedy this issue of manual template creation, a fully automatic template extraction method, using SRL, from sample input questions contained in the dataset is explored. This approach aims to ease the time-consuming work associated with creating templates while greatly improving the coverage of the template-based system, and not degrading its quality when compared to the semantic QG system.

For the evaluation of the aforementioned systems, the focus is on assessing both question quality and system coverage. To assess question quality, the criteria of *grammatical correctness*, *logical sense* and *relevance* are used. These were evaluated through an online survey, for which ethics clearance was obtained. In this survey, participants were presented with contextual excerpts extracted from the dataset, along with a range of questions generated from the context by the different systems. Participants were asked to rate the questions on a 5-point Likert scale in the above categories. A simultaneous internal evaluation making use of random sampling of transcripts and questions was also performed. This analysis served to complement the survey, aiming to pinpoint any recurring challenges within the systems' question generation abilities. During this internal random sampling, information regarding the systems' coverage was also assessed. This encompassed metrics such as the *number of questions generated per lecture transcript*, the *questions produced per sentence*, and the *number of sentences contributing to the generated questions*.

The results obtained indicate that survey participants generally regarded most questions, generated by both systems, as being of a high quality (i.e.: an average score of over 3.5 in all categories). The survey also showed no statistically significant difference between the semantic and template-based systems across any of the categories. This may highlight the robustness of the automatic template extraction module, which demonstrated its capability to achieve comparable performance to a system based on manually crafted rules. Furthermore, the template-based system exhibited markedly superior coverage, compared to the semantic system, and was even able to outperform a neural system in terms of *logical sense* and *relevance*. However, it fell short in terms of question quality, when compared to a neural system trained on a more extensive dataset. These findings suggest a promising potential for further refining the template-based system to close the performance gap to large neural systems.

The subsequent sections will delve into related work, emphasizing prior semantic and template-based QG systems and the aspects that can be drawn upon and enhanced from them. It will then transition to detail information regarding the dataset and the architectures of the developed systems, subsequently delving into the evaluation of these systems, analysing the results, and culminating in a comprehensive discussion and conclusions, alongside suggestions for potential future advancements.

2 RELATED WORK

Question generation (QG) systems possess the ability to generate questions based on a given input, which can vary from a piece of text to an image [14]. The demand for such systems is continuously increasing, especially in the educational context, where asynchronous learning has resulted in reduced direct interaction between teachers and students. Thus far the only system to investigate question generation from lecture transcripts has been Ros et al.[18]. Their system utilised a neural model, to generate questions from lecture transcripts. This leaves the gap to explore the performance of simpler semantic and template-based methods which have been shown to produce high-quality questions in other domains. The subsequent paragraphs will explore the primary concepts and methodologies of semantic and template-based systems, highlighting previous systems and their key takeaways.

Semantic question generation hinges on the utilisation of semantic role labelling (SRL) to gain an understanding of a sentence's underlying structure. This comprehension serves a dual purpose: not only does it enable the extraction of phrases to generate questions, but it also guides the selection of question types. This is evident in a variety of systems where the semantic roles identified guide the question word selection, such as an *ARGM-MNR* phrase indicating a *how* question can be asked [4, 11, 12, 15]. A key limitation of these systems is that they produce questions tightly coupled to the sentence itself. Therefore, template systems, especially those that can leverage the power of SRL, are of great interest.

Template-based systems operate using a set of pre-defined templates with placeholder slots. These slots correspond to various linguistic elements such as parts of speech, entities, or semantic

phrases, which can then be filled to produce questions. In this process of QG, three key modules are needed: content extraction, template creation and template filling. The following paragraphs will explore the existing research into these three components.

For text-based input, such as lecture transcripts, two main methods to analyse and extract content exist: syntactic and semantic analysis. Syntactic content analysis involves parsing input text, labelling parts of speech and identifying key phrases based on the grammatical structure of the sentences. Many systems utilise this method of content extraction [2, 13, 20, 23]. Syntactic analysis is simple to perform and has been shown to produce valid questions in other domains. Yet, in the educational domain only making use of grammatical structure could limit a system's ability to extract meaningful phrases. Thus, systems which make use of semantic content analysis to extract content are of more interest [1, 7, 10]. Semantic analysis works by trying to achieve an understanding of the relationship between different phrases. This can potentially enable more interesting parts of a sentence which span over many different syntactic patterns to be identified [10]. It is for this reason that semantic analysis is the preferred mechanism of content extraction for this paper.

A significant limitation of template-based systems is the time-consuming process of creating templates, which are inherently tied to the specific context for which they were devised. Many systems employ a manual process for creating templates [2, 10, 13]. Manual template creation allows precise control, enabling templates to be tailored towards specific question types and for the quality of them to be maintained. This comes at the expense of requiring a large time investment, thus reducing the scalability and coverage of these systems. It is for these reasons that methods to automatically create templates have been employed. This automation can come through the extraction of templates from sample questions. Similarly to content extraction, sample questions can be analysed. Using this analysis, the phrases identified can be replaced by slots to form a template [21]. More sophisticated approaches involve using a neural system to perform this same function [6]. These approaches collectively address the challenge of creating templates while enhancing question diversity and adaptability to different domains [9]. For these reasons, the template-based system constructed in this study will employ an automatic template extraction module inspired by the work of Teo and Joy [21]. This will involve performing SRL on sample questions and then replacing these labelled phrases with slots to form templates.

The next key issue is how these templates can be matched to the source content and subsequently filled to produce questions. A simple approach to template filling is to populate a template when all its slots can be satisfied by extracted content [2, 7, 13, 20]. This method is overly general and can lead to inappropriate templates being used. There thus exists a need to use additional matching criteria. One way to do this can be through categorising templates and then linking the extracted content to these categories [21]. Another is to incorporate a slot outside of the template structure that facilitates extra matching criteria [10]. The systems in this paper will utilise both mechanisms. The semantic system will employ phrases outside of the QG rules, while the template system will use categorisation. Using these strategies will provide a more nuanced way of connecting templates with relevant content, thus improving the specificity and relevance of generated questions.

Moving onto other systems, recently neural approaches have emerged as the favoured choice for QG [8, 18]. However, interestingly, comparisons with template-based and even simpler semantic systems have revealed that their performance is not as overwhelmingly superior as one might anticipate. For instance, Flor and Riordan's [4] semantic system exhibited better performance when compared to a neural system developed by Du et al [3]. This highlights the power of the semantic approach to QG. Furthermore, in the E2E challenge, Puzikov and Gurevych concluded that their template-based system was superior, to their neural system, due to its greater grammatical correctness and its retention of the source content in the generated questions [16]. These findings underscore the continued relevance and potential of both semantic and template-based systems in question generation.

By combining the features of previous rule and template-based systems this paper seeks to create fully automatic systems that can generate high-quality questions from lecture transcripts. The analysis of these two systems will encompass their coverage of the dataset and the quality of questions they produce in terms of grammatical correctness, logical sense, and relevance. A comparison to neural systems, developed in a complementary paper [22], will also take place.

3 DATASET AND SYSTEM CREATION

Within this section, a comprehensive overview, encompassing the dataset utilised in this study, along with the methodology used to construct both the semantic and template-based QG systems, will be presented. This will be accompanied by an explanation of the evaluation process employed to assess the systems' performance in terms of coverage and quality.

3.1 Dataset

The dataset used in this investigation is taken from Ros et al. [18], which itself extracts the data from the two larger open source Massive Open Online Course (MOOC) datasets. The dataset contains 12 weeks of lectures on the information retrieval topics of search engines, text retrieval, text mining and analytics. Accompanying the 87 transcripts are files containing questions that were asked by students during these lectures. These questions are used to extract the templates to form novel questions. In total 594 questions were stored as plausible questions from which templates could be extracted.

Table 2: Example of a Transcript and its Accompanying Questions in the Dataset

File Type	Content
Transcript	*"[SOUND] This lecture is about natural language content analysis.... And that's a good starting point though. Thanks.[MUSIC]"*
Question	*"What's the difference between semantic and pragmatic analysis?"*
Question	*"What's the difference between POS tagging and parsing?"*

Table 2 displays the starting and concluding sentences from the lecture labelled *"2 - 1 - 1.1 Natural Language Content Analysis (00-21-05).txt"*. It also showcases two questions that were posed during the lecture.

3.2 Semantic System

The semantic QG system is a rule-based framework which utilises SRL to transform the input sentences into questions. The process entails several stages: segmenting the input transcript into sentences, performing SRL on these sentences and finally matching the semantic tags against the pre-defined rules. This rule-based matching allows for the identification and generation of valid questions directly from the input sentences. A comprehensive depiction of the system architecture can be viewed in Figure 1.

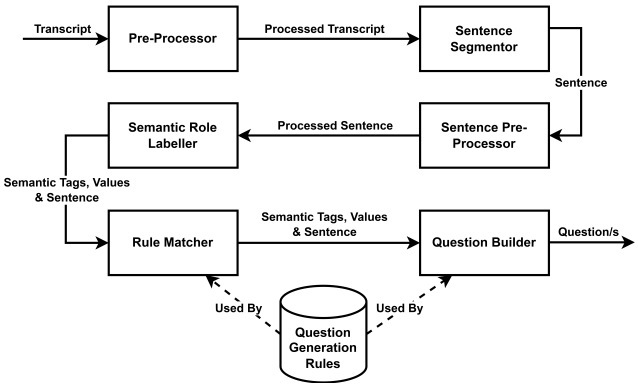

Figure 1: The Semantic QG System Architecture

As per Figure 1, the process of question generation commences with the input of the transcript. The transcript is then pre-processed to expand the contractions within it. This is undertaken as it allows for more accurate tokenisation of phrases during SRL, produces superior consistency throughout the text and plays a role in averting grammatical issues that might arise when phrases are extracted and rearranged to produce questions.

After this pre-processing, the transcript is segmented into sentences using the SpaCy [1] library. At this point each valid sentence undergoes SRL. By making use of SRL a deep understanding of the meaning of sentences, and the phrases they are made up of, can be achieved. This understanding can then be used as a guide to determine what questions can be generated [4]. To perform the SRL, an open source neural model developed by the Allen Institute for AI, specifically their SRL BERT model from December 2020 is utilised [5]. Once imported into the Python environment, this model can process sentences and produce a dictionary containing semantic tags and their corresponding values, adhering to the PropBank specification [2]. The example below demonstrates the application of SRL to the sentence "*The boy loves to kick the ball and play football*".

- Loves:
  - ARG0 = *"The boy"*
  - V = *"loves"*
  - ARG1 = *"to kick the ball and play football"*
- Kick:
  - ARG0 = *"The boy"*
  - V = *"kick"*
  - ARG1 = *"the ball"*
- Play:
  - ARG0 = *"The boy"*
  - V = *"play"*
  - ARG1 = *"football"*

The outcome produced by the SRL model is then reformatted into parallel lists comprising semantic tags and their corresponding values. These semantic tags serve as the criteria for matching the sentence against the pre-defined QG rules. Upon detecting a match (i.e.: all the tags in the sentence match those specified in the rule), the appropriate values are extracted from the values list, facilitating the generation of one or more questions. This process is repeated for all sentences in the transcript.

The foundation of the semantic QG system rests on eight manually crafted rules which were intentionally designed to balance grammatical accuracy and system coverage. The rules utilise the semantic tags identified within the sentences to assess which type of *wh-question* word is applicable [4]. The tag that provides this indication has been termed the *answer tag* as it typically provides the answer to the generated question. It is important to note that this does not necessarily mean all questions are directly answerable from the text. To illustrate this concept consider the examples of the semantic tag *ARGM-MNR* which signals the potential for a *how* question to be generated, while the semantic tag *ARGM-LOC* instead suggests the possibility of a *where* question being produced. The list of rules, and their accompanying *answer tags*, can be seen in Table 3.

Table 3: The Semantic Rules used to Generate Questions

Rule No.	Semantic Rule	Answer Tag
1	Why <ARGM-MOD> <ARG0> <V> <ARG1>?	<ARGM-PRP>
2	Why <ARGM-MOD> <ARG0> <V> <ARG1>?	<ARGM-CAU>
3	What <ARGM-MOD> <ARG0> <V> <ARGM-PRP>?	<ARG1>
4	What <ARGM-MOD> <ARG0> <V> <ARGM-CAU>?	<ARG1>
5	To what extent <ARGM-MOD> <ARG0> <V> <ARG1>?	<ARGM-EXT>
6	How <ARGM-MOD> <ARG0> <V> <ARG1>?	<ARGM-MNR>
7	How <ARGM-MOD> <ARG0> <V> <ARG1>?	<ARGM-ADV>
8	When <ARGM-MOD> <ARG0> <V> <ARG1>?	<ARGM-TMP>

Upon inspection of the rules, it becomes evident that they all include the *ARGM-MOD* semantic tag. The inclusion of this tag, which encompasses the auxiliary verbs, assists in ensuring the grammatical correctness of the questions. To exemplify the need for

[1] https://spacy.io/
[2] https://propbank.github.io/

this semantic phrase consider the sentence: *"We can mine text data to understand their opinions."* paired with the rule: *" Why <ARG0> <V> <ARG1>?"* Applying this rule yields the question: *"Why we mine text data?"*. While this question is valid, it falls short of grammatical correctness. However, by adhering to *rule 1* detailed in Table 3, the question transforms into: *"Why can we mine text data?"*, a far grammatically superior question.

In addition to the enhanced grammatical precision, the inclusion of the *ARGM-MOD* tag enables the question rule to become more refined by enabling the inclusion of additional semantic tags. This can be viewed through the simple rule: *"Why <V> <ARG1>?"*, which using the same example sentence generates the question: *"Why mine text data?"* Again, this question is valid but unfocused due to its lack of a subject. However, by including the *ARGM-MOD* tag it enables us to generate the more focused question presented in the paragraph above.

An example of the system's output and intermediate representations can be seen in Table 4. In this example the the input sentence undergoes SRL, resulting in the identification of semantic tags within the sentence. These semantic tags are then compared against the rules within the system. Upon examining the rules in Table 3, it becomes apparent that due to the presence of the tags: *ARG0, ARG1, ARGM-MOD, ARGM-TMP, V* and *ARGM-PRP* the sentence can be matched with rules 1, 3 and 8. The phrases associated with these semantic tags can then be re-arranged to produce three distinct questions in accordance with the matched rules.

Table 4: Semantic Question Generation Example

Sentence: *That is a function that will take a document and query as input, and then give a zero or one as output, to indicate whether this document is relevant to the query, or not.*
SRL: *That is <ARG0> <R-ARG0> <ARGM-MOD> take a document and query as input, <ARGM-TMP> <V> <ARG1 <ARGM-PRD> ,<ARGM-PRP>.*
Rule 1 Match: *Why <ARGM-MOD> <ARG0> <V> <ARG1>?. Answer Tag = <ARGM-PRP>*
Generated Question: *Why will a function give a zero or one?*
Rule 3 Match: *What <ARGM-MOD> <ARG0> <V> <ARGM-PRP>? Answer Tag = <ARG1>*
Generated Question: *What will a function give to indicate whether this document is relevant to the query, or not?*
Rule 8 Match: *When <ARGM-MOD> <ARG0> <V> <ARG1>? Answer Tag = <ARGM-TMP>*
Generated Question: *When will a function give a zero or one?*

3.3 Template-based System

The template-based system functions as a larger iteration of the semantic system. It is split into three key modules that each work together to successfully generate questions from the dataset. The modules are the *template extraction module*, *content extraction module* and *template filling module*. A simplified diagram of the system can be seen in Figure 2 while a full system architecture is available in Figure 8 in Supplementary Information A.

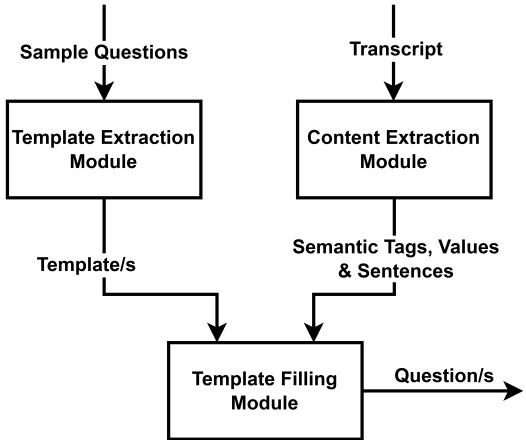

Figure 2: The Simplified Template QG System Architecture

3.3.1 Template Extraction Module. The initial step in the question generation process involves the extraction of templates. This is visualised in Figure 3. This process begins with the input of a text file containing sample questions. These questions were posed by students in connection with the lecture transcripts contained in the dataset. These questions are then clustered, using k-means clustering, where k is equal to 12. The clustering was executed through the *Sci-Kit (sklearn) library* in Python [3]. The random seed of 42 was used to make this stochastic process reproducible. Clustering was undertaken to fulfil two key objectives. Firstly, to establish more linkage between sentences and suitable templates. While this could have been accomplished through the manual creation of template categories, k-means clustering allowed the system to remain automated. Secondly, clustering was essential to reducing the number of "valid" templates per sentence. Without the clustering of templates, an excess of redundant questions (often 10s per sentence), were generated. The clustering process efficiently mitigated this issue by minimising and focusing the number of generated questions.

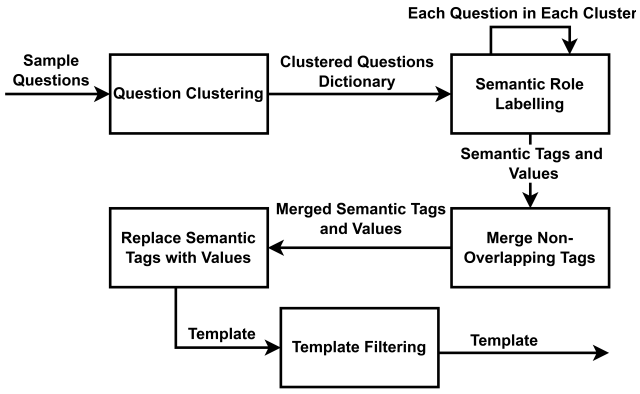

Figure 3: The Template Extraction Module

After clustering, the questions are stored in a dictionary with the following structure: *{Cluster1: [q1,q2,q3,...], Cluster2: [q1,q2,q3,...],...}*.

[3] https://scikit-learn.org/stable/modules/clustering#k-means

This dictionary of clustered questions is then passed to a method which performs the *template extraction*. Within this method, each question undergoes SRL. However, due to the output structure of the role labeller, some further logic is applied to merge non-overlapping semantic tags thereby consolidating them into a single set of tags for each question. A full example of this can be seen with the sample input question: *"What is the meaning of postings? Is it just another word for inverted index?"* The question undergoes SRL with the output of this viewable in Figure 4. The 3 separate frames of tags are then merged together to form the template: *"What is <ARG1>? Is <ARG1> <ARG2>?"*

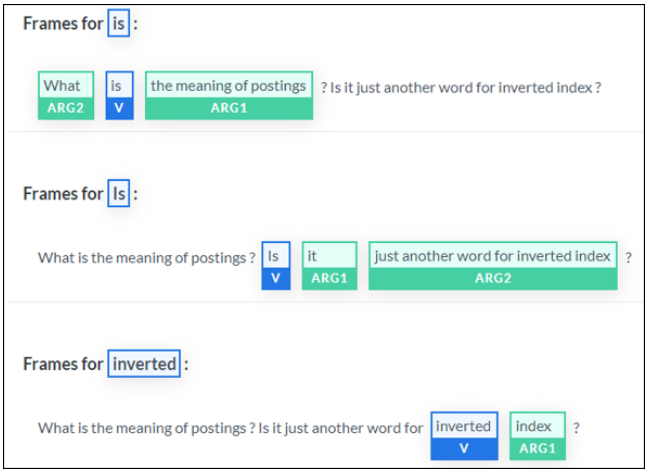

Figure 4: Semantic Role Labelling of a Sample Input Question [5]

Following the semantic tagging of a sample question, the next step is to transform it into a potential template. This process commences by ensuring that the first word in the question is either a *wh-question word* or an ARGM-MOD semantic tag (i.e.: an auxiliary verb). These criteria were chosen to maintain the naturalness and consistency of the questions while also enhancing the likelihood of the questions being grammatically correct. If these criteria are met, the phrases within the question are substituted with their respective semantic tags, thereby establishing slots that will later be filled by the content.

After the template is generated, it undergoes an evaluation against further predefined criteria:

- **No duplicate templates:** The template must not already have been extracted.
- **Minimum of three slots:** The template must have at least three slots that can be filled. This ensures the template is complex enough to generate high-quality questions.
- **No duplicate slots:** The template must not contain more than one of the same slot value. This limits potential issues when slot filling.
- **50% of the template must be slots:** This requirement details that at least half of the words in the sentence must be a fillable slot. This prevents the template from containing excessive non-tag content, making it too context-specific and thereby reducing its usefulness beyond its original domain.

Satisfying all these criteria leads to the template being appended to the output dictionary, ensuring it remains in the same cluster as the sample question from which it was extracted. As an example of these rules in action, the template extracted in Figure 4: *"What is <ARG1>? Is <ARG1> <ARG2>?"*, would not be a valid template due to *ARG1* appearing twice.

3.3.2 Content Extraction Module. The *content extraction* module is an adapted version of the semantic system. It contains the same steps of removing contractions, segmenting the sentences and performing SRL on the sentences. The module can be seen in figure 5. Instead of performing rule matching against the semantic tags, the template-based system utilises the sentence itself, along with the semantic tags, to retrieve valid templates that can then be filled. This is performed in the *template filling* module.

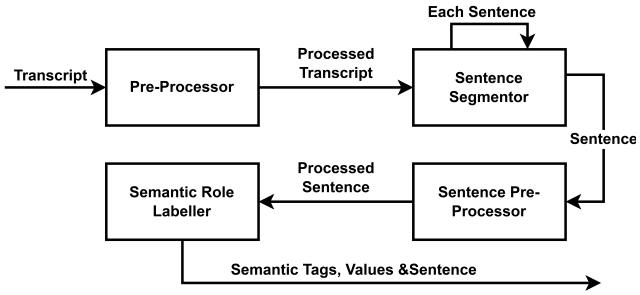

Figure 5: The Content Extraction Module

3.3.3 Template Filling Module. The *template filling* module is responsible for filling the templates and thus producing the questions. Upon receiving a sentence, accompanied by its associated semantic tags and values, the module initiates its procedures by predicting into what cluster the sentence should be allocated. This prediction is performed using the same *SKLearn* cluster model trained to cluster the sample questions and thus the templates. After prediction, the system retrieves the templates corresponding to the sentence's assigned cluster.

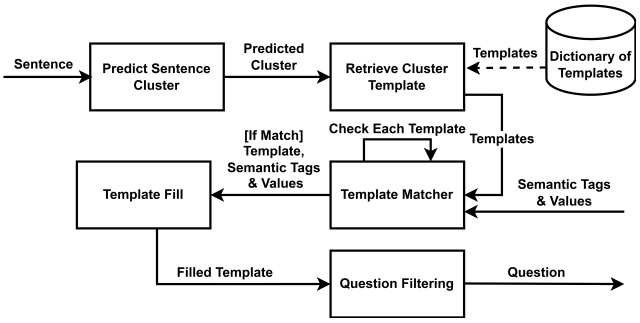

Figure 6: The Template Filling Module

The list of retrieved templates is then iterated over with the slots in each template checked against the semantic tags within the

sentence. If a match is found for all slots in the template, the slots are filled with the appropriate values extracted from the sentence, thus forming a question. Once the question is produced further checks are performed to mitigate the generation of redundant questions. This evaluation entails: ensuring that only a single question is produced per *wh-question* word (i.e.: one *why* question) and per *ARGM-MOD* phrase (i.e.: *Could*, Should, Would, etc.). Doing this ensures that many versions of very similar questions are not generated. The module and its components can be viewed in Figure 6.

3.4 Evaluation

Evaluation of the systems was performed through two complementary processes: an online survey and an internal assessment. These evaluation methods were devised to evaluate the output quality, assess any statistically significant differences in question quality and assess the coverage of the two systems.

The online survey presented participants with 10 contexts and questions generated from these contexts by the different systems. In the case of multiple questions being generated from the context by a system, one question was randomly selected. Participants were asked to judge the question quality based on three criteria: *grammatical correctness*, *logical sense*, and *relevance*. These criteria were selected due to their comprehensive coverage of the attributes essential for a well-constructed question and were rated on a 5-point Likert scale. The definitions provided to the survey participants were as follows:

- **Grammatical Correctness**: *The degree to which the tense, punctuation, word order and word usage are correct.*
- **Logical Sense:** *The degree to which the meaning of the question can be understood.*
- **Relevance:** *The degree to which the question is deemed appropriate based on the context from which it is being asked.*

Incorporated within the online survey were questions generated by 3 neural systems developed by Adam Vere from identical contexts [22]. By comparing the scores of these models, to the comparatively simpler semantic and template-based systems developed in this paper, it allows for valuable insights to be gained. These insights specifically focus on using the neural systems as baseline, for comparing system performances. The results of this could motivate further research into enhancing automatic template extraction mechanisms.

To compare the systems across the three categories, p-values were computed using the Mann–Whitney U test. This test was chosen because the data was non-parametric, meaning it did not follow a normal distribution. This analysis served to identify whether statistically significant differences existed among the systems.

Simultaneously, internal assessment sought to provide insights into the underlying reasons for the systems' performance and aimed to identify potential areas of vulnerability. This internal assessment involved randomly selecting five transcripts from the dataset. These transcripts were then processed by both systems to generate questions. The generated questions were then analysed and the coverage was assessed through the following metrics: the *number of questions generated per lecture transcript*, the *number of questions produced per sentence*, and the *number of sentences from which questions were generated.*

4 RESULTS

The survey received responses from 15 participants, all of whom were home language English speakers. Furthermore, with the exception of one participant who had a high school level education, all possessed an academic background of at least an undergraduate degree. This section will present the survey results, emphasising the mean scores attained by the semantic, template, worst, and best neural systems developed by Adam Vere [22]. A full overview of survey results is available in Table 5, and a detailed breakdown by context can be found in Figure 7. Table 6 explores any statistically significant differences in performance, while Table 7 showcases the coverage of the two systems.

Table 5: Counts of all Likert Ratings for Each Category and System

Category	System Name	1	2	3	4	5	Mean
Grammatical Correctness	Semantic	13	23	23	18	75	3.77
	Template	24	12	22	36	56	3.59
	Worst Neural	6	25	31	36	52	3.69
	Best Neural	0	3	3	24	120	4.74
Logical Sense	Semantic	19	19	20	27	65	3.67
	Template	11	16	20	33	70	3.90
	Worst Neural	10	22	28	39	51	3.66
	Best Neural	2	2	7	9	130	4.75
Relevance	Semantic	15	23	23	40	49	3.57
	Template	11	18	24	46	51	3.72
	Worst Neural	13	24	35	45	32	3.38
	Best Neural	5	17	20	49	59	3.94

At a high level, Table 5 demonstrates remarkably similar, and satisfactory performance of the semantic, template, and the worst performing neural system. In contrast, there is a noticeable superiority of the best-performing neural system across all categories. In assessing statistical significance, no significant differences emerge between the semantic and template systems (i.e.: p-value > 0.05). When comparing the template-based system to the worst neural system statistical significance emerges (i.e.: p-value < 0.05) in the categories of *logical sense* and *relevance*.

By combining this p-value observation, which was obtained using ranked survey scores in the Mann-Whitney U test, with the mean scores of each category, it becomes evident that participants found the questions generated by the template-based system to be superior to those generated by the worst neural system in these two categories. However, the same cannot be said when comparing the template-based system to the best neural system. In this scenario, statistically significant differences were found in all three assessed categories (i.e.: p-value < 0.05). This combined with the neural system's superior mean scores indicates that participants strongly favoured the questions generated by the neural system over the template-based system. These findings are outlined in Table 6.

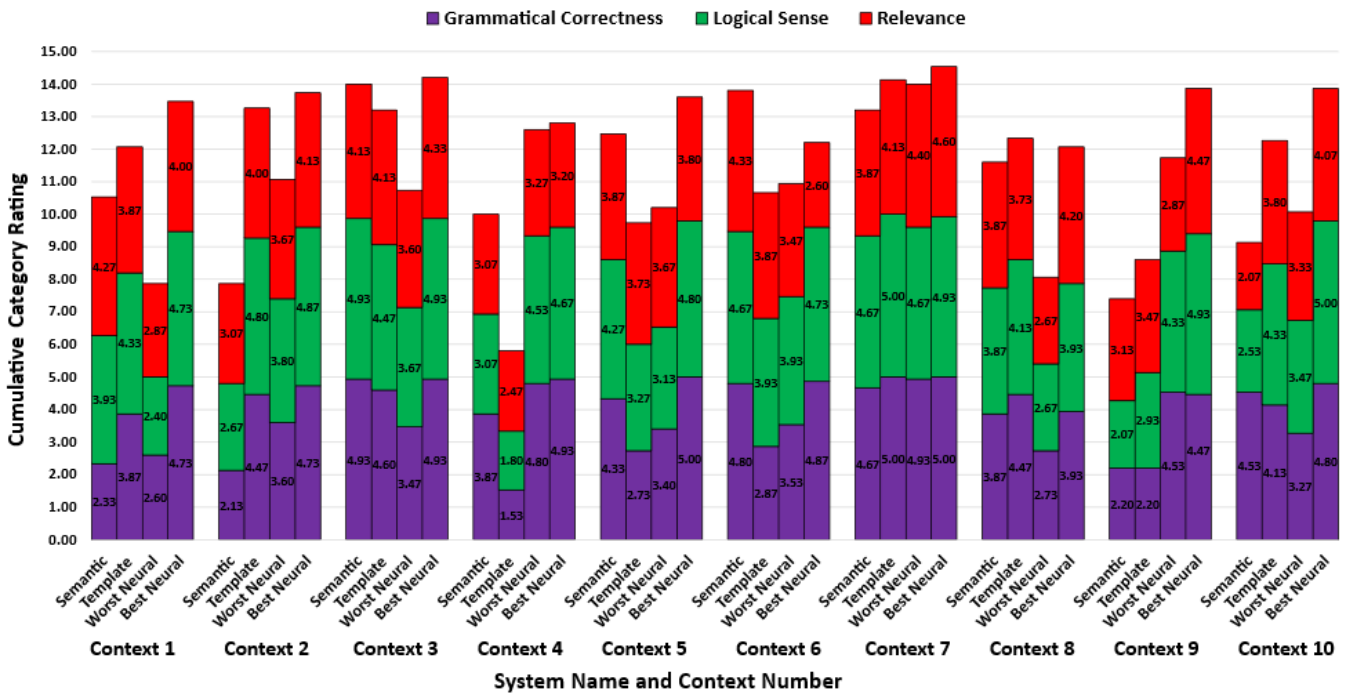

Figure 7: Cumulative Average Score for Each System per Context

Table 6: Comparison of the Semantic, Template, Worst and Best Neural Systems

Category	Semantic System Mean	Template System Mean	p-value
Grammatical Correctness	**3.77**	3.59	0.103
Logical Sense	3.67	**3.90**	0.128
Relevance	3.57	**3.72**	0.203
	Template System Mean	**Worst Neural System Mean**	**p-value**
Grammatical Correctness	3.59	**3.69**	0.442
Logical Sense	**3.90**	3.66	**0.031**
Relevance	**3.72**	3.38	**0.007**
	Template System Mean	**Best Neural System Mean**	**p-value**
Grammatical Correctness	3.59	**4.74**	**2.116e-13**
Logical Sense	3.90	**4.75**	**5.442e-10**
Relevance	3.72	**3.94**	**0.003**

Looking deeper at the results on a per-context basis, as shown in Figure 7, it is seen that both the semantic and template-based systems generally scored over 3.5 in each category for every context. Demonstrating their satisfactory performance over a variety of randomly selected contexts. However, there are a number of contexts where one or both of the systems performed poorly. Investigating these contexts further can provide interesting insights. This will be explored in the discussion section.

Table 7: Coverage of the Semantic and Template System

	Semantic System	Template System
Number of Rules or Templates	8	81
Total Sentences	668	668
Total Sentences Utilised	90 (13.47%)	**468 (70.06%)**
Total Questions Generated	131	**1206**
Questions per Sentence	1.46	**2.58**

During the internal evaluation, the coverage of the semantic and template-based systems was assessed. The results of this assessment on five randomly selected transcripts are presented in Table 7. In the table *Total Sentences Utilised* signifies the number of sentences from which questions were generated. The value beside it describes the percentage of the sentences for which questions were generated. The results demonstrate the anticipated outcome that the template-based system exhibits significantly higher coverage compared to the semantic system. In fact, the ratio of templates employed, 8 rules vs. 81 templates, roughly lines up with the number of questions generated, 131 questions vs. 1206 questions.

5 DISCUSSION

The results demonstrate the achievement of the template-based system in meeting its objective of maintaining the quality of questions while concurrently enhancing coverage as compared to the semantic system. This means that the automatic template extraction module produced performance equal to the manual rule-creation of the semantic system, which formed the main baseline in this study. This is evidenced by the insignificant p-values (i.e.: larger than 0.05) when the systems were compared in the three assessed categories. This underscores the potential for the automatic template extraction module to alleviate the tedious process of template creation.

Furthermore, the template-based system exhibited the capability to surpass a neural system in the aspects of *logical sense* and *relevance*, even though it was decisively outperformed by a different neural system, trained on a considerably larger corpus. A possible explanation for this performance gap is that the semantic and template-based systems extract text directly from sentences, whereas neural systems can learn deeper relationships. This is particularly evident as the model trained on a larger corpus showcased superior performance. This deeper understanding, obtained from a larger collection of texts, can result in more comprehensive transformations of contexts into questions, compared to simple phrase extraction. The consequences of direct text extraction can be seen in contexts 1, 2, 4 and 9 as per Figure 7. These specific contexts and their generated questions can be seen below:

- **Context 1:**
  - *Sentence*: "And in order to recover the, the exact document ID, we have to first recover the previous document ID, and then, we can add the difference to the previous document ID to restore the, the current document ID"
  - *Semantic Generated Question*: "What can we add to restore the, the current document ID?"
- **Context 2:**
  - *Sentence*: "Now in Wilcoxon, test, it's a non parametrical test, and we would be not only looking at the signs we'll be also looking at the magnitude of the difference."
  - *Semantic Generated Question*: "How will we looking at the magnitude of the difference?"
- **Context 4:**
  - *Sentence*: "And that basically, means, the probability that this result is in fact from random fluctuation."
  - *Template Generated Question*: "How do that means the probability that this result is in fact from random fluctuation basically?"
- **Context 9:**
  - *Sentence*: "But if you look at these two experiments and look at the detailed results, you will see that we'll be more confident to say that in the case one."
  - *Semantic Generated Question*: "How will you see that we will be more confident to say that in the case one?"
  - *Sentence*: "We can easily say that, well, System B is better, right?"
  - *Template Generated Question*: "How to say that, well, System B is better easily?"

Looking at these contexts the poor performance of the systems can generally be explained by direct content extraction. This means that grammatical issues in the sentences can permeate through the generated questions. This is evidenced in context 1, where the repetition of the word *"the"* leads to grammatical issues in the generated question, and in context 2 where the verb form *"looking"* is incorrect. Another significant issue identified was the systems' inability to handle words or phrases that make sense in spoken speech but not in written text, this can be seen with the word *"basically"* in context 4 and the word *"well"* in context 9.

Many of these flaws could be fixed with iterative improvements to the systems. These include: stemming of verbs, such as changing the verb *"looking"* to *"look"* in context 2, and introducing detection mechanisms for colloquial words like *"basically"* as well as for the repetition of words such as *"the, the"*. These mechanisms were not implemented due to time constraints and full evaluation only taking place after development. However, to greatly improve the overall question quality it is likely that further enhancements to the *template extraction* and *template filling* modules would be needed.

To perform the *template extraction* and *template filling*, SRL and k-means clustering were the only mechanisms utilised. This had the benefit of simplicity and allowed the system to remain fully automatic. However, it led to the creation of overly general templates that only contained semantic tags. This generality meant that a large number of questions were generated per sentence. While k-means clustering reduced the number of generated questions, examining these questions indicates a mix of poor and high-quality outputs, usually with at least one high-quality question per sentence. The example below demonstrates six questions produced from a single sentence. Among these six questions, it becomes apparent that *Question 4* stands out as the most well-structured. Conversely, the other five questions display grammatical or logical inconsistencies, highlighting the challenges inherent in generating high-quality questions through solely semantic-driven templates.

- **Sentence:** *And in this case, a user typically would use a search engine to fulfill the goal.*
- **Question 1:** *How do a user use a search engine?*
- **Question 2:** *Why a user would use a search engine typically?*
- **Question 3:** *Will a user use a search engine?*
- **Question 4:** *Why would a user typically use a search engine?*
- **Question 5:** *How do a user fulfill the goal?*
- **Question 6:** *Will a user fulfill the goal?*

To enhance the system's performance, a separate classification system for selecting the best questions could be considered. This could utilise automated metrics such as BLEU or ROUGE. However, this was beyond the scope of this study. In addition, relying solely on SRL and k-means clustering limited the system's capacity to effectively link logically related sentences and templates. Again a separate classification system could be built to conduct separate analysis and categorisation of templates and sentences. Another simpler approach could be to incorporate the *answer tag* that was utilised in the semantic system. This could assist in linking question words to related sentences (i.e.: if an *ARGM-PRP* phrase is detected then retrieve the *why* templates).

An additional challenge observed within the internal evaluation of the template-based system arose with lengthy sentences containing multiple commas. In such cases, the system may produce excessively long questions that, although syntactically correct, deviate from the concise nature of typical classroom questions. The sentence: *"We also keep a lot of ambiguities because we assume the receiver, or the hearer could know how to discern an ambiguous word, based on the knowledge or the context."* highlights this as the question: *"How could the receiver, or the hearer know how to discern an ambiguous word, based on the knowledge or the context?"* is generated. This question is grammatically valid but is overly verbose and not aligned with the form of questions commonly encountered in educational contexts.

A final noteworthy observation regarding the template-based system comes from its comparatively poor performance compared to the best neural system in terms of *grammatical correctness* and *logical sense*. Typically, template-based systems have the advantage that they can enforce stricter rules, resulting in more grammatically correct questions. However, in this system due to SRL being the sole driver to create the templates, some of this traditional strictness was removed leading to comparatively poorer performance than expected.

Delving deeper into the performance of the semantic system two interesting insights arise. Firstly, as expected, the inclusion of the *ARGM-MOD* tag in the semantic rules contributed to the enhanced *grammatical correctness* of the questions produced. This is evidenced by the higher mean score in comparison to the template-based system, and it represents the primary enforced difference between the rules and templates in the two systems. However, it is worth noting that this difference is not statistically significant.

The second insight comes from the behaviour of the system, in particular with *how* questions. The system frequently demonstrated the tendency to paraphrase *how* questions that already exist within the transcript. Although not inherently advantageous or disadvantageous, as these questions are legitimate inquiries a student might ask, the phenomenon is of interest. An example of this is with the sentence: *"So, this lecture and the following lectures will be mainly about how we can mine and analyze opinions buried in a lot of text data"* from which the system produced the question: *"How can we analyze opinions buried in a lot of text data?"*

Finally, while the template-based system drastically exceeded the coverage of the semantic system it still left 200 sentences for which questions were not generated. These were ignored as they did not match any template in their assigned cluster. While it's unrealistic to expect a question for every sentence in a lecture, these remaining sentences could potentially contain important questions. Further refinement of matching criteria or the utilisation of a larger number of sample questions for template extraction could potentially increase the coverage of the system.

6 CONCLUSIONS

The obtained results show that both the rule-based semantic and template-based systems possessed the ability to generate high-quality questions when assessed by human evaluators. In addition, no statistically significant differences in performance between the semantic and template-based systems were detected. This highlights the potential robustness of the automatic template extraction module. Notably, the incorporation of a greater number of templates substantially enhanced the coverage of the template system in comparison to the semantic system. This was achieved without compromising the quality of the generated questions. Additionally, the template-based system managed to surpass the performance of a neural system, although it fell short in comparison to a neural system trained on a larger corpus. This was likely caused by the template-based system struggling to adapt to the nuances of spoken speech, reliance on direct text extraction, and its lack of ability to learn deeper relationships within the text. Nevertheless, these findings underscore the continued performance of semantic and template-based systems and indicate the possibility of reallocating some of the extensive training time required for neural models to improve fully automated template-based systems.

7 FUTURE WORK

Further research into refining template and sentence matching criteria could potentially bridge the gap between template-based and large neural systems. These refinement could include: introducing classification systems to better categorise templates, to better link sentences to appropriate templates and to automatically evaluate and filter generated questions. In addition, improvements to the content extraction module such as stemming of verbs, and detection for colloquial and repeated words could help enhance the grammatical correctness of the generated questions.

REFERENCES

[1] Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1415–1425. https://doi.org/10.3115/v1/P14-1133

[2] Wei Chen and Gregory Aist. 2009. Generating Questions Automatically from Informational Text. *PHD Thesis, Carnegie Mellon University* (2009). https://www.cs.cmu.edu/~listen/pdfs/QG2009-Wei-informational-text-questions-final.pdf

[3] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1342–1352. https://doi.org/10.18653/v1/P17-1123

[4] Michael Flor and Brian Riordan. 2018. A Semantic Role-based Approach to Open-Domain Automatic Question Generation. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, New Orleans, Louisiana, 254–263. https://doi.org/10.18653/v1/W18-0530

[5] Allen Institute for AI. 2020. AllenNLP Semantic Role Labeling. Available at: https://demo.allennlp.org/semantic-role-labeling. Last Updated: 15 December 2020, Accessed: 23 August 2023.

[6] Yunfan Gu, Yang Yuqiao, and Zhongyu Wei. 2019. Extract, Transform and Filling: A Pipeline Model for Question Paraphrasing based on Template. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Association for Computational Linguistics, Hong Kong, China, 109–114. https://doi.org/10.18653/v1/D19-5514

[7] Hafedh Hussein, Mohammed Elmogy, and Shawkat Guirguis. 2014. Automatic English Question Generation System Based on Template Driven Scheme. *International Journal of Computer Science Issues (IJCSI)* 11 (11 2014), 45–53.

[8] Myeong-Ha Hwang, Jikang Shin, Hojin Seo, Jeong-Seon Im, Hee Cho, and Chun-Kwon Lee. 2023. Ensemble-NQG-T5: Ensemble Neural Question Generation Model Based on Text-to-Text Transfer Transformer. *Applied Sciences* 13, 2 (2023), 903. https://doi.org/10.3390/app13020903

[9] Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education* 30 (2020), 121–204. https://doi.org/10.1007/s40593-019-00186-y

[10] David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating Natural Language Questions to Support Learning Online. In *Proceedings of the*

14th European Workshop on Natural Language Generation. Association for Computational Linguistics, Sofia, Bulgaria, 105–114. https://aclanthology.org/W13-2114

[11] Prashanth Mannem, Rashmi Prasad, and Aravind Joshi. 2010. Question generation from paragraphs at Upenn: QGSTEC system description. In *Proceedings of QG2010: The Third Workshop on Question Generation*. Pittsburgh, Pennsylvania, USA, 84–91.

[12] Karen Mazidi and Paul Tarau. 2016. Infusing NLU into automatic question generation. In *Proceedings of the 9th International Natural Language Generation conference*. Edinburgh, UK, 51–60. https://doi.org/10.18653/v1/W16-6609

[13] Jack Mostow and Wei Chen. 2009. Generating Instruction Automatically for the Reading Strategy of Self-Questioning. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*. IOS Press, NLD, 465–472.

[14] Nikahat Mulla and Prachi Gharpure. 2023. Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications. *Progress in Artificial Intelligence* 12, 1 (2023), 1–32. https://doi.org/10.1007/s13748-023-00295-9

[15] Hugo Patinho Rodrigues, Luísa Coheur, and Eric Nyberg. 2016. QGASP: a Framework for Question Generation Based on Different Levels of Linguistic Information. In *Proceedings of the 9th International Natural Language Generation conference*. Association for Computational Linguistics, Edinburgh, UK, 242–243. https://doi.org/10.18653/v1/W16-6640

[16] Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG Challenge: Neural Models vs. Templates. In *Proceedings of the 11th International Conference on Natural Language Generation*. Association for Computational Linguistics, Tilburg University, The Netherlands, 463–471. https://doi.org/10.18653/v1/W18-6557

[17] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering* 3, 1 (1997), 57–87. https://doi.org/10.1017/S1351324997001502

[18] Kevin Ros, Maxwell Jong, Chak Ho Chan, and ChengXiang Zhai. 2022. Generation of Student Questions for Inquiry-based Learning. In *Proceedings of the 15th International Conference on Natural Language Generation*. Association for Computational Linguistics, Waterville, Maine, USA and virtual meeting, 186–195. https://aclanthology.org/2022.inlg-main.14

[19] Vasile Rus, Sidney D'Mello, Xiangen Hu, and Arthur C. Graesser. 2013. Recent Advances in Conversational Intelligent Tutoring Systems. *AI Magazine* 34, 3 (2013), 42–54. https://doi.org/10.1609/aimag.v34i3.2485

[20] Liana Stanescu, Cosmin Stoica Spahiu, Anca Ion, and Andrei Spahiu. 2008. Question generation for learning evaluation. In *2008 International Multiconference on Computer Science and Information Technology*. IEEE, 509–513. https://doi.org/10.1109/IMCSIT.2008.4747291

[21] Noor H Ibrahim Teo and Mike S Joy. 2018. Categorized Question Template Generation for Ontology-Based Assessment Questions. *International Journal of Knowledge Engineering* 4, 2 (2018), 72–75. https://doi.org/10.18178/ijke.2018.4.2.103

[22] Adam Vere. 2023. Investigating the use of pre-trained data-driven models for automatic question generation from lecture transcripts. *Honours Project, University of Cape Town* (2023).

[23] Bambang Dwi Wijanarko, Yaya Heryadi, Hapnes Toba, and Widodo Budiharto. 2021. Question generation model based on key-phrase, context-free grammar, and Bloom's taxonomy. *Education and Information Technologies* 26, 2 (2021), 2207–2223. https://doi.org/10.1007/s10639-020-10356-4

A SUPPLEMENTARY INFORMATION

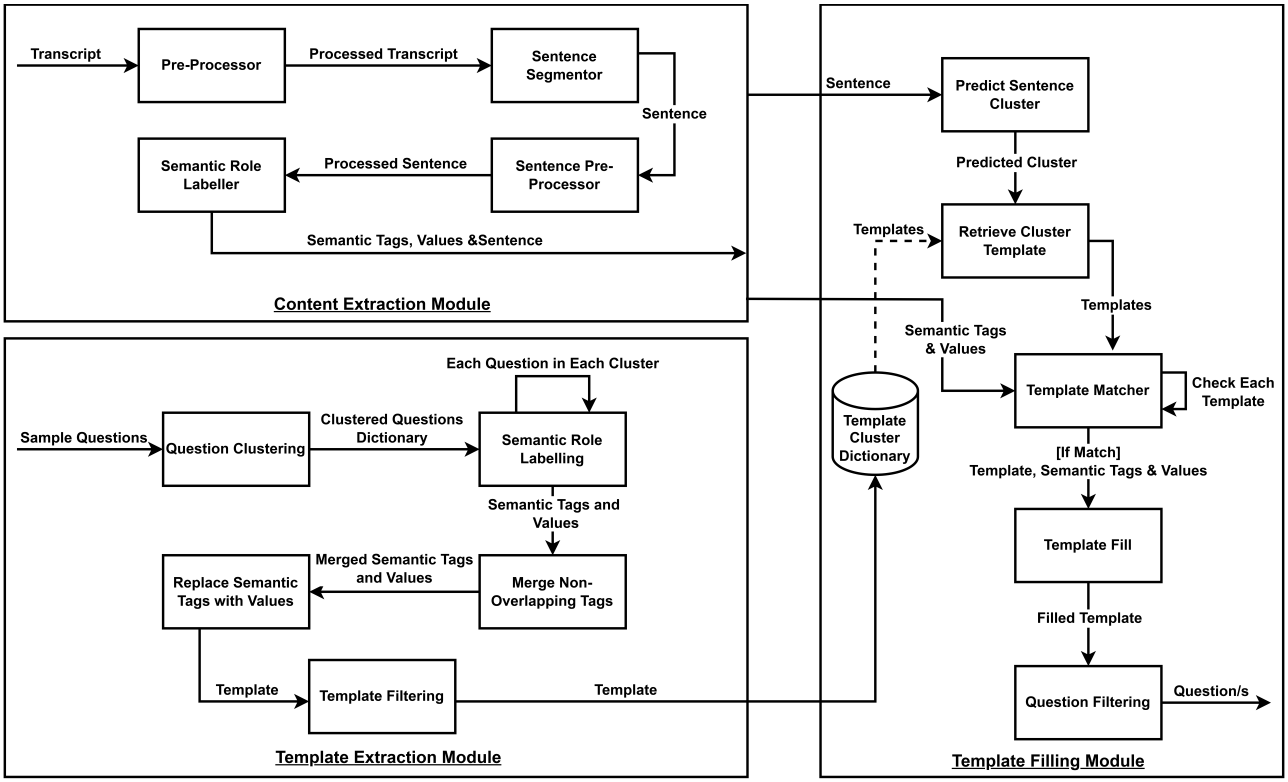

Figure 8: The Template QG System Architecture