Automated Generation of Knowledge Assessment Items

William Smith

MSc in Machine Learning and Autonomous Systems (with Placement)
The University of Bath
2020-2022

Automated Generation of Knowledge Assessment Items

Submitted by: William Smith

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of MSc in Machine Learning and Autonomous Systems in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

Question generation is an important topic in the field of intelligent tutoring and a challenge for natural language processing techniques. Traditionally, in-depth linguistic knowledge has been encoded in complex pipelines of rules, but more recently neural network based techniques derived from translation were utilised. Here, a hybrid approach was taken, where sophisticated preprocessing was applied until it was beneficial to use the power and flexibility of modern machine learning. GPT-2 was then fine-tuned on the SQuAD dataset to select an answer, and generate questions from the answer and context. Lastly, an LSTM network, trained on tagged questions, ranked the generated questions by suitability.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Dr Ekaterina Kochmar, for their supervision, direction and assistance.

Further gratitude is due to the University of Bath Machine Learning teaching team for helping me understand and enjoy a field new to me.

Lastly, I am very appreciative of my family and friends, including Margot Navellou and Stephanie Libby for their great support, encouragement and proofreading.

Chapter 1

Introduction

1.1 Description of the societal challenges

The manual generation of questions for a classroom, tutorial, or online course is time consuming and frequently duplicated between many institutions. Although there have been some recent collections of resources, these remain limited and new material is still continually required due to curricula changes.

Traditionally educators have used textbooks as one source of questions. For some fortunate fields such as mathematics or engineering, it is possible to write templates to generate an unlimited list of questions through number substitution. However, even within STEM subjects, the tools for generating questions are more limited for fields such as biology [1].

As well as providing assessment resources for a whole cohort, it would be ideal if individual students could receive personalization for their work as this can help with motivation and development. However, the time required to do this is typically far beyond the reach of many institutions. Furthermore, the growth in self-led study using resources available on the internet provides an additional demand for questions, as it is particularly hard to generate material to assess your own knowledge level [2].

These are some of the compelling reasons for why an intelligent tutoring system (ITS) can help the education of many. Once written, an ITS can be distributed rapidly, widely and used simultaneously by many users. However, the time required to create the ITS can be lengthy with one hour of material taking hundreds of hours of expert work in some cases [3]. Question generation is one of the important bottlenecks of writing these systems.

1.2 Description of the technical challenges

Syntactic transforms are a straightforward way to generate questions. The sentence ‘John is the driving instructor’ can be readily converted to ‘Who is the driving instructor?’. However, a naive algorithm running such replacements may convert ‘Parrots live long lives’ into ‘Parrots live what?’. It may also suggest ‘What lives long lives?’, which would not challenge a student who knows that they are being tested on parrots.

As well as failing to identify what is important, this approach is also limited in that there are many sentences that are unsuitable to be coherently converted this way, such as when there are multiple pieces of information. Converting ‘John and Jane went for a walk, where John fell over’ to ‘Who and Jane went for a walk, where John fell over’ is both grammatically poor and trivial to answer.

Using single sentences for conversion may seem preferable due to the difficulty of capturing the meaning of a paragraph, but can often lead to questions being overly narrow [4]. To compensate for this, it is important to select which sentences are key. This process of selection, also known as central concept identification or target concept identification, has been argued to be one of the most important stages of question generation for ITS [5]. There are many methods that have been used for central concept identification, from simple heuristics to concept relation graph based ranking [6].

Once the individual sentences, or short runs of sentences have been chosen, it is necessary to determine what form the questions will take. It is more common to address the ‘who’, ‘what, ‘where’ and ‘when’ for computer generation, whereas ‘why’ and ‘how’ are seen as extensions of greater difficulty [6, 7], which intuitively pose greater challenges both in writing questions and assessing answers.

After the form has been decided, the question has to then be generated. There are a range of current approaches, from complex NLP pipelines to recurrent neural networks [8]. Benefits have been reported from overgenerating questions and later ranking them as a method of presenting the best questions to the final users [8] [9]. Suitable shortlists of questions may be ranked in order of difficulty [1] and lastly, for some situations there may be manual selection of the best questions from the generated shortlist.

Many challenges lie within the field of question generation. If the difficult requirements are not adequately met then the questions are likely to be irrelevant, stilted or nonsensical. However, as automatic question generation potentially offers such substantial benefits for education, research in this field is likely to produce significant future benefits to both students and educators.

1.3 Objectives

The central aim of this project is the generation of questions. These will come directly and solely from the teaching material, and not through the augmentation of existing questions or through a precomputed database. Answers will be generated

alongside the questions, but the assessment of how closely a student's answer matches the generated answer lies outside the scope of this work.

An initial task within the larger aim of generating questions is the separation of complex sentences into shorter factoid style sentences, offering limited and unambiguous information. Following this separation, the main focus will be the subsequent conversion of the sentences into questions. In the past this has often been approached using complex pipelines of rules, though here this process will rely on modern neural networks. Additional aims include the ranking of generated questions, as seen for measuring the linguistic quality of hints [10].

1.4 Key deliverables

1. Code for sentence preprocessing and question generation
2. Code for identification of interesting answers within contexts
3. Code to rank generated questions

1.5 Overview of chapters

This chapter provides information on the reasons why this project was undertaken, as well as the aims and also the challenges.

The literature review forms Chapter 2. Further and more specific information on motivations are detailed, before descriptions of conventional approaches to question generation. Following this are new approaches and their relevance here. Lastly the details of the datasets and software used in this research are provided.

In Chapter 3, the requirements of the task are described, beginning with what would be necessary for a complete software package and ending with which parts were selected and the specification of this project.

The design of the experiments is described in Chapter 4, broken down into three sections. The experiments within each section, along with their results, form the next three chapters.

Chapter 5 describes the seven experiments performed on the central theme of this work, question generation. These are listed in the order they were performed and this is the order in which improvements were expected to be made.

For Chapter 6, an experiment on identifying interesting answers is covered, and is a modification to the structured prompt format previously used.

Chapter 7 details methods used to rank the generated questions, focussing on the training and usage of an LSTM.

Chapter 8 concludes the document, revisiting the potential benefits of the work, briefly outlining the experimental methods and summarising the results.

Chapter 2

Literature and Technology Survey

2.1 Overview of Question Generation (QG)

Asking students to answer questions and work through examples is common in education [11], and so there is a strong motivation to develop tools or techniques that can help with question generation. The questions can be asked of students to help them with comprehension or assessment, or they may also be asked to generate the questions themselves, as this has been found to help their own understanding [4]. This presents another use of QG, as the student's questions could be compared to the generated ones.

In many STEM subjects, templates can be used by a teacher to manually generate large numbers of questions for students to practice on. A slightly more advanced approach is to use simple scripts to automatically create worksheets, and vast amounts of material can be made in short time using this approach [1].

However, this approach still requires time consuming creation of templates, and furthermore, it is not suitable for many subjects. An ideal solution would be more efficient and useful by directly synthesising questions from a wide range of source materials such as textbooks, novels, or Wikipedia [6]. It would be able to generate these questions on topics ranging from ethics to operation of robotic arms [4, 12].

There are different types of questions that may be generated, such as deep or shallow ones. Deep questions may be of the why, how or what types [13], and can involve assessment of an entire article to ascertain which topics are of central importance. An example may be "How have attitudes to nuclear power changed over the last two decades?". One type is definitional, and one way to ask this type of question is to identify key phrases from a document and prefix them with a wh-word and "is" [4]. For example, "What is *Escherichia coli*?" could be generated from a document on gut microbiota. Deep questions are likely to take longer to answer and are harder for a computer to provide feedback for [13]. These questions may also be more difficult for a computer to write, and for lower grade students, there appears to be little difference in rates of learning between deep and shallow questions [14]. Lastly, as the *E. coli* example illustrates, there may be many ways to answer them, ranging from the trivial ("A bacterium") to the sophisticated ("An important model species, used to study topics ranging from genetics to nanoscale biological motors").

Instead, this work will focus on shallow questions, which are typically made from individual sentences, or a small number of concatenated sentences. Questions such as "What is the average annual rainfall in Wales?" are shallow ones. These could allow teachers more time to carefully design deep questions, as well as being useful for diagnosing any underlying lack of understanding [4]. Somewhat deeper

questions were also permitted but limited to starts such as ‘How much’ and ‘Why did’, and this is discussed in Chapter 5.6.

2.2 Target concept identification

For QG to develop into a sophisticated tool, it is important that it does not simply apply syntactic transforms to input sentences. Instead, the key concepts from a document should be identified so that students focus on the most important material. The type of question (where, why, how) is key and this will vary based on the subject and context. Choosing an appropriate question difficulty is desirable as this allows for increasingly difficult questions to be asked, as well as allowing QG to be used for fields where the questions are not deliberately challenging, such as medical diagnoses [5].

Approaches for identifying key concepts can be found in work on summarization. An example is the extension of the MEAD summarizer into the COGENT system, which uses Term Frequency Inverse Document Frequency (TFIDF) and a search engine, to score each sentence [15]. The COGENT system can be used to produce sentences which summarise each concept. These are then classified by an SVM to determine how linked the concepts are and build a map of them [5]. From the map, decisions can be made about which topics to focus on, or how to introduce them sequentially.

An overview of other approaches, including decision trees, MLPs and PageRank inspired methods is given in [16]. In this work, a neural network model is described that utilises a Long Short-Term Memory (LSTM) decoder to learn and predict locations of human selected sections.

2.3 Conventional approaches

It proved possible to generate questions through specific templates [17], and this technique was investigated for some time [18], but was limited in applicability.

Moving beyond this in sophistication, the approach of Heilmann and Smith was influential and has been widely cited [2, 4, 6, 8, 19, 20]. In their work, a wide range of NLP operations were used to produce a complex and linear pipeline from material to questions [8, 9]. To generate questions, some piece of information must be removed from the sentence and this is a basic exercise for a human, who can comfortably manipulate, add to, or subtract from the material for clarity. For a computer though, extensive manipulation and addition of words or context were challenging, however, so Heilmann et al. restricted themselves to minor modification, subtraction and introduction of question words.

Some spans are islands, which means that moving a word from the island to outside of it results in an ungrammatical sentence. For example, “My meeting with Jane this morning was pleasant” cannot be converted to a question by simply extracting the

name as this would give “Who was my meeting with this morning was pleasant?”. This stands in contrast to sentences such as “Einstein developed SR in Switzerland in 1905”, which could become “Who developed SR in Switzerland in 1905?” or “What did Einstein develop in Switzerland in 1905?”. The first stage then, in the work of Heilmann et al., was marking these sections as unsuitable for the removal of information for QG. The next stage was labelling words using entity identifiers, removing words from unmarked sections, and inserting wh-words appropriately.

Next, the main verb was identified and where necessary, split into its base form and the relevant form of ‘do’ (do, does, did, done). Subject-auxiliary inversion was then performed. An example of this would be converting ‘John has eaten’ to ‘Has John eaten?’. This was followed by the final formation of the questions through insertion into copies of the original syntactic tree.

Some simple filtering of possibilities then occurred, before the final stage where the questions were ranked. Instead of attempting to create a process where every generated question would become an output, Heilmann et al. filtered out poorly defined questions involving pronouns and implemented a ranking system. This featured a number of hand written metrics such as noun phrase count, or whether the answer span was found in the source sentence. A model was trained to assess the significance of each metric, and then used to calculate an overall score and ranking [9]. Given that a computer may produce a vast number of questions rapidly, and that some problems (such as negation) are easier to identify in the final question than in the source material, the overgenerate and rank approach appears to be a useful one.

2.4 Neural networks

QG using conventional rules is capable of producing questions with a low to acceptable quality [7]. Each stage is well understood and can be explained. However, the creation of these complex pipelines requires significant understanding of linguistics and they become increasingly challenging to maintain and add to as their sophistication grows.

A contrasting approach is the use of neural networks for question generation (NQG), which has attracted increasing interest since 2017. For this, it is not necessary to separate out the decisions of what to ask about and of how to ask about it. Instead the entire section of text is provided as an input and the question is directly generated as an output.

The earlier NQG techniques used Seq2Seq models [21]. These are a development on the prior technique in neural translation of using an encoder to convert a sentence into a vector (of constant length), before using a decoder to produce the translated output. With this system, the decoder is trained using a recurrent neural network (RNN) to learn what the next word will be, given the input vector and the words that it has already generated [22]. Unfortunately, as the encoder has to capture all the information contained within the sentence, the accuracy drops as the sentence length increases [23]. To overcome this, Seq2Seq uses an attention

mechanism, trained as another neural network, to provide the decoder with information about how well inputs at one position are aligned with outputs at another position. This allows the decoder to focus on one part of the input more than another, and provides another way to use information from the input without having to solely rely on the encoder (with its associated issues with long inputs) [24].

One issue with computational QG is the frequent presence of the answer in the question, and this can be avoided through masking the answer and training another neural net to provide information about it [25]. For example, using a masked input of "<a> lived in New Zealand before 1445" avoids the answer being in the question and means it is unlikely the question will be a 'how', 'where' or 'when' question. However, to determine whether it is a 'who' or 'what' question, further information about the answer '<a>' is needed and that is provided by the neural net. There are a range of techniques for this, as the large majority of models from this era used this method to increase specificity of generated questions [21].

A recent introduction is that of the Transformer, which exclusively uses attention mechanisms to perform tasks such as natural language translation with state-of-the-art accuracy [26]. This has also been used for question generation without the need for additional features such as answer encoding [27]. In the work of Lopez et al., a pre-trained model of GPT-2 was further trained on the SQuAD dataset following reformatting operations. Using various evaluation metrics, this approach was found to have higher performance than previous Seq2Seq models. A significantly more complex approach using two Transformers and answer awareness forms the UniLM approach, offering even higher performance [28]. When benchmarked on the same inputs, using metrics of BLEU_4, METEOR and ROUGE_L, the work of Lopez et al. scored 8.3, 21.2 and 44.4, whereas the work of Dong et al. scored 22.1, 25.1 and 51.1 [27]. However, the relative simplicity of the work by Lopez et al. [27] means that their approach is of particular interest here.

One evaluation metric was Bilingual Evaluation Understudy Score (BLEU), which is a measure of the overlap of n-grams between the input and output [29]. BLEU was first used to assess machine translations and accepts multiple reference (human written) translations for one input. Calculation of it is a multistage process with the first step being finding the modified n-gram precision. If *Count* is the number of times a unigram appears in the machine translation and *MaxRefCount* is the maximum number of times that it appears in the reference translation, the clipped count is given by:

$$Count_{clip} = \min(Count, MaxRefCount)$$

This is then used to calculate the modified precision score for values of *n* between 1 and 4. If *Candidates* is the output set of sentences from the machine translation, then the modified precision score is:

$$P_n = \frac{\sum_{C \,\epsilon\, \{Candidates\}} \sum_{n-gram \,\epsilon\, C} Count_{clip}(n-gram)}{\sum_{C' \,\epsilon\, \{Candidates\}} \sum_{n-gram' \,\epsilon\, C'} Count(n-gram')}$$

In words, this means to sum the clipped n-gram counts for the machine translated sentences and divide by the number of n-grams in the machine translated sentences. This measures how closely the machine and human translated sentences match, and it can be seen that the precision will be low if the machine generates much longer sentences than humans did. However, it is important to avoid giving erroneously high scores to generated sentences that have good overlap but are far too short. If *c* is the total length of the candidate (machine translated) sentences and *r* the total length of the reference (human translated) sentences that is closest to *c*, then the brevity penalty is:

$$BP = \begin{cases} 1 & c > r \\ e^{(1-r/c)} & c \leq r \end{cases}$$

Taking the cutoff length for n-grams as 4 and applying equal weights gives a BLEU score of:

$$BLEU = BP \times e^{\log p_1 + \log p_2 + \log p_3 + \log p_4}$$

BLEU is flexible and it is possible to use different weights or cutoffs. However, the above definition was the baseline in the original paper [29], and this is what is referred to as a BLEU score throughout this work.

2.5 Datasets

A comparative overview of datasets used for QG is given by Pan et al. [21]. In the Shallow cognitive level category, the first dataset is SQuAD, which was originally created to help train models to answer questions [30]. However, it can also be used for QG, as the crowdsourced questions can become the output that the model aims to produce. The next item in this category is the NewsQA dataset, which shares many similarities with SQuAD but appears to be more challenging to answer [31].

Following this are the Medium cognitive level datasets, with Microsoft MAchine Reading COmprehension dataset (MS MARCO) being one example [32]. This was derived from 1 million searches on Bing, each with a human answer, and 8 million passages from the internet containing the knowledge required to answer the questions. As this was derived from searches, it is not known which sentences in the passages contain the information, which would increase the difficulty of the task here, when compared to SQuAD. ReAding Comprehension Dataset From Examinations (RACE) is the other entry in this category and originates from 100,000 exam questions and 28,000 passages [33]. Neither the questions, nor the answers are spans from the context, which again would make this dataset more challenging to use than SQuAD. Lastly, the Deep cognitive level datasets questions are less applicable here, as shallow questions are focussed on due to the reasons discussed in Section 2.1.

2.6 Software and pre-trained models

There are a number of languages, libraries and software packages that are likely to be used for NLP manipulation as part of this project, in particular those used by Heilmann and Smith [9]. Tregex and Tsurgeon are Java based tools for the extraction, visualisation and modification of trees, and are used widely for QG [6] [4, 20].

It is important to know what type of object is being discussed in the material (person, place, event etc.) or generating questions about it will be challenging. For this, a named entity recogniser is used, with one example used for QG being the Stanford NER [25]. Other options, such as the BBN IdentiFinder Text Suite are detailed here [34], but the Stanford NER currently appears to be the most popular. It is important to note that if contemporary events are being discussed, these systems will rapidly become less usable if not updated. As for syntactic trees, SpaCy provides a NER [35], which was used here.

As used in the work on Transformer based QG [27], the company Hugging Face have made available pre-trained GPT-2 models [36]. Creating one from scratch is a task that would have been far outside the scope of a project such as this, so these models may prove vital here. Although GPT-2 is often used for extended prose creation, one technique that could be useful is prediction from a structured prompt, such as sentiment analysis on Twitter [37]. Instead of fine tuning using tweets and sentiments, it may be possible for GPT-2 to learn how to generate questions from contexts.

Lastly, for evaluating the similarity between human generated answers and automatic QG, the BLEU algorithm is commonly used [22, 28, 38].

Chapter 3

Requirements

To be widely used, an automatic question generation system needs a number of components. These can include:

- Assessment of key topics within text
- Identification of sentences covering these topics
- **Identification of a potential answer within each sentence**
- **Generation of questions from the sentence and answer pairs**
- **Selection of questions**
- An answering system with either:
    - alternative multiple choice answers or
    - a way of measuring whether a typed answer is sufficiently close to the correct answer
- A user interface

This project is experimental and does not attempt to meet all of these requirements or produce a finished software package. Instead it focussed on the middle sections, in bold above. The concept is that this project could provide core machine learning components which later could be combined with existing techniques such as TextRank [39] for a standalone piece of software or a feature within an intelligent tutoring system.

3.1 Identification of a potential answers

It is possible to generate a question without an answer and this is particularly suited to essay style questions. However, for shorter factual questions, it significantly diminishes the utility of the software if the answer cannot be automatically assessed. For this reason, answers were selected for each question here. As for Question Answering (QA) work [30] the answers will be a span from the source material.

3.2 Question generation

The core part of this project is the transformation of context-answer pairs into questions. These sentences should be grammatically correct, readily understood, relevant, specific, and should not give away the answer. Due to the many idiosyncrasies of language, it is very difficult to formulate a set of rules by hand to

meet these requirements, and so advanced machine learning language models were investigated instead.

3.3 Question ranking

Due to the challenging nature of question generation, it is unlikely that a given process will produce only high quality results. One solution is to abandon any context/answer pair which yields an unsuitable answer. Although this may be necessary at times, it is wasteful and only possible for long articles. Instead, multiple questions may be generated and then ranked.

3.4 Requirements specification

- **Identification of a potential answer within each sentence (Ch. 6)**
  - Answer must be a span from the source sentence.
  - Span selection must correlate with selections from human authors.
- **Generation of questions from sentence and answer pairs (Ch. 5)**
  - Questions must be generable from any reasonable context-answer pair
  - Questions must resemble human written ones and be trained on them.
  - Multiple possible questions must be generated.
- **Selection of questions (Ch. 7)**
  - Generated questions must be filtered to avoid basic errors such as the answer in the question.
  - Questions must then be ranked according to likelihood that they are valid sentences.

3.5 Overview of experiments by chapter

Chapter 5.1 discusses the preprocessing of SQuAD, the structured prompts used to train GPT-2 and the results from this.

Chapter 5.2 shows the improvements made from reducing the original context down to single sentences, but also the problems this creates.

Chapter 5.3 addresses these problems through coreference resolution, yielding further improvement in similarity between original and generated questions.

Chapter 5.4 describes the reduction of source material further through splitting of the sentences.

Chapter 5.5 details attempts made to reduce the vocabulary of the material in order to make the learning of context and question patterns more readily apparent.

Chapter 5.6 covers the reduction of source material by limiting the number of

question starts to avoid distraction by lesser used question types.

Chapter 5.7 describes improvements made through hyperparameter optimization.

Chapter 6 covers the reshaping of the prompt training structure so that interesting answers may be selected.

Chapter 7 looks at how generated questions may be ranked, both through simple rules and an LSTM model that assesses the likelihood of the sentence.

Chapter 4

Design of the experiments

In Chapter 3, three different tasks were described (identification of an answer, generation of questions, selection of a question) and a number of different experiments were conducted for each of these tasks.

Experiments were performed first for the generation of questions. This allowed techniques developed to be used in later work on identification of answers. As the answers were selected from the context, whereas the questions were generated, the converse (answer identification experiments informing question generation) would not necessarily hold.

Next followed the identification of answers experiments, and lastly the work on ranking of generated questions is described.

4.1 Generation of questions

One way to train a language model is to take a large collection of human written material and here the Stanford Question Answering Dataset (SQuAD) was chosen. The SQuAD training set consists of groups of contexts, questions, answers, beginning positions and ending positions of the answer span. One example is shown below:

- In some countries, YouTube is completely blocked, either through a long term standing ban or for more limited periods of time such as during periods of unrest, the run-up to an election, or in response to upcoming political anniversaries. In other countries access to the website as a whole remains open, but access to specific videos is blocked. In cases where the entire site is banned due to one particular video, YouTube will often agree to remove or limit access to that video in order to restore service.
- youtube is completely what in some places?
- blocked
- 338, 345

Typically SQuAD is used for QA by training models to predict the start and end of the answer span. Given that the context is finite (and that answers are constrained by sentences), this is a multiple choice problem with many options.

Using SQuAD for Question Generation (QG) in the same manner is not possible because the question is not contained within the context and so text must be

generated. Furthermore, not even all the words for the question will be present in the context. Information about a person who is “an author of” may be provided, yet the question may begin with “Who wrote”. This necessitates an advanced language model, but it is vital to monitor and control hallucination, which is insertion of new or spurious information in the question that is not present in the context. It also requires a model that can be fine-tuned for the specific task of question generation.

The model chosen was Open-AI’s Generative Pre-trained Transformer 2 (GPT-2) [40]. This has received substantial attention in the press [41, 42] and is mainly known for its ability to create short passages of realistic synthetic text [43, 44]. An atypical use of GPT-2 is sentiment analysis on Twitter [37], as discussed in Chapter 2. As GPT-2 can be fine-tuned in that way, it was considered worth investigating here whether it could be fine-tuned to create questions by using the context and the answer as a prompt.

As discussed in Chapter 2, a key theme of this work is the exploration of whether preprocessing can significantly help models to produce high quality results. It is in this section that the preprocessing of the context sections is also studied.

4.2 Identification of answers

Previously, the input was context & answer, and the output was the question. As GPT-2 can be fine-tuned on any text, it is possible to modify this so that the input is the context and the output is the answer. This may initially seem absurd; it is as if the answer is sought without the question even being known. However, the goal here is not to find the original answer but to find which spans are typically of interest to the human authors of SQuAD.

For example, if the context is:

> The Protestant movement began to diverge into several distinct branches in the mid-to-late 16th century.

Then valid answers may include:

> The Protestant movement
>
> several distinct branches
>
> mid-to-late 16th century
>
> 16th century

Unhelpful answers could be:

> movement began
>
> in the

It is unreasonable to expect any model to have a high accuracy in this way, but ideally it will have at least a moderate accuracy and a similar spread of word types and answer lengths.

4.3 Selection of questions

GPT-2 can be configured to produce multiple possibilities, which allows for the possibility of ranking and selecting the best question. Several rules for this were considered in the experiments here, such as repetition, hallucination measurement and avoidance of giving away the answer. Following this, language models (LSTM) were trained and applied, with their results compared to the rules.

Chapter 5

Experimental results - Question Generation

Seven experiments were performed on question generation and their results are described here. The experiments are described in the order they were performed, and this sequence was chosen to allow incremental improvements to be made. For example, the sentence splitting work (5.4) incorporated coreference resolution (5.3) and training on single sentences (5.2).

5.1 Training on entire context

The SQuAD dataset “Training Set v2.0 (40 MB)” was processed into context-answer-question groups. These were then split into a training set (94%), validation set (3%) and testing set (3%). The latter sets were small because GPT-2 is relatively slow to generate text.

Contexts with sentences longer than 256 words were removed, as these were often technical items such as diagrams or chemical reactions. Entries with an answer position of 0 or -1 were also removed to ensure there was an answer for each set.

The model was fine-tuned using Max Woolf’s GPT-2 Simple[1] system, with an example input being:

> [Context]: An independent test in 2009 uploaded multiple versions of the same song to YouTube, and concluded that while the system was "surprisingly resilient" in finding copyright violations in the audio tracks of videos, it was not infallible. The use of Content ID to remove material automatically has led to controversy in some cases, as the videos have not been checked by a human for fair use. If a YouTube user disagrees with a decision by Content ID, it is possible to fill in a form disputing the decision. YouTube has cited the effectiveness of Content ID as one of the reasons why the site's rules were modified in December 2010 to allow some users to upload videos of unlimited length.
>
> [Answer]: were modified
>
> [Question]: What happened to the sites rules in Dec. 2010?
>
> <|endoftext|>

[1] https://github.com/minimaxir/aitextgen

The relevant sentence is highlighted in green, the required information in blue, and the answer in red. The last line is a specified token that breaks the training set into examples instead of a single long stretch of text. For generating results, the text after “[Question]: “ for each entry in the validation set was removed. Several generated questions are shown below:

- What can Content ID cause the site to say?
- What was the reason YouTube's rules were modified in December 2010 to allow some users to upload videos of unlimited length?
- What did YouTube announce about the content of videos that they removed?
- What is one possible reason YouTube would modify its rules?
- How is it possible for a user to upload a video of which length?

The generated questions aligned poorly with the ground truth (“What happened to the sites rules in Dec. 2010?”), and the average BLEU score from this experiment was 0.32. Some did not make sense, others contained the answer, and not all corresponded to a specific section of the context. However, GPT-2 reliably grasped the input pattern and consistently produced reasonable sounding and generally grammatically correct questions. This gave reason to continue with this approach.

Many of the suggested questions are focussed on the wrong sentences but (unlike in QA) selecting the correct sentence is not part of the challenge, as it will always be known from which sentence the answer will be found. The next experiment therefore followed the same pattern but using only the relevant sentence as the context.

5.2 Training on single sentences

For each input set, the context was reduced to the sentence containing the answer. This removed the possibility of a generated question referring to a different part of the context than the answer was from. It was also believed that it may assist training through shortening the inputs. In all other respects, the model was trained in the same way as in Section 5.1.

This change yielded an increase in BLEU scores from 0.32 to 0.36. An example item in the validation dataset that showed an improvement is shown below:

> [Context]: In the 1960 election to choose his successor, Eisenhower endorsed his own Vice President, Republican Richard Nixon against Democrat John F. Kennedy.
>
> [Answer]: Richard Nixon
>
> [Question]: Who did Eisenhower endorse for president in 1960?

Comparing the first generated question here with the first for the last approach gives:

> Previous approach:
>
> > Who lost the election to whom?
>
> Selected sentence approach:
>
> > Who was the Vice President that Eisenhower endorsed?

This example shows how this approach improved on the previous one by asking about the relevant sentence in the context.

A disadvantage of this approach is that this reduction of the context can mean that information is lost:

> [Context]: He also ordained that the tax-payers were free, having paid his local tax, to choose their own church.

Without knowing who ‘He’ was, it is likely any generated question will be vague and unsatisfactory. To address this, the context text needs to be automatically edited, both for training and generation.

5.3 Coreference resolution

Written text frequently included ambiguities that need to be resolved by the reader. One example is:

> John lives with James in London, and there are many things he likes about the city.

Two ambiguities are present, ‘he’ and ‘the city’. Rewriting to remove them yields:

> John lives with James in London, and there are many things John likes about London.

This process is called coreference resolution [45] and is a challenging problem for NLP [46]. The entity may be named before or after a reference (anaphora or cataphora) [47]. Multiple entities may be referred to with a single word (split antecedents) [48] or multiple words may be used both for the entities and the reference (noun phrases) [45]. Resolving these ambiguities reduces the risk of the model asking vague questions and makes it more likely that all necessary information is contained within the selected sentence.

Two code packages were investigated for this task, both of which use neural networks. The first was NeuroSYS’s ‘coreference-resolution’[2] and this leveraged ‘NeuralCoref’[3] from HuggingFace and ‘coreference_resolution’[4] from AllenNLP, together with written rules to select which produced the best output. The results were excellent, though installation was challenging due to specific dependencies and outdated documentation. The second package was ‘coreferee’[5] from Explosion AI, which also yielded high quality results.

[2] https://github.com/NeuroSYS-pl/coreference-resolution
[3] https://github.com/huggingface/neuralcoref
[4] http://docs.allennlp.org/v0.9.0/api/allennlp.models.coreference_resolution.html
[5] https://github.com/explosion/coreferee

A representative example is below. Original:

> In June 1776, Benjamin Franklin was appointed a member of the Committee of Five that drafted the Declaration of Independence. Although he was temporarily disabled by gout and unable to attend most meetings of the Committee, Franklin made several small changes to the draft sent to him by Thomas Jefferson.

NeuralCoref:

> … Although he was temporarily disabled by gout and unable to attend most meetings of the Committee of Five that drafted the Declaration of Independence, Benjamin Franklin made several small changes to the draft sent to Benjamin Franklin by Thomas Jefferson.

Coreferee:

> … Although Franklin was temporarily disabled by gout and unable to attend most meetings of the Committee, Franklin made several small changes to the draft sent to Franklin by Thomas Jefferson.

The two resolved concepts are highlighted, and the different approaches can be seen. NeuralCoref typically produced a more literal resolution, forcing each entity to be labelled with the full expression, which was useful for containing the maximum information within one sentence. Coreferee was more flexible and less stilted, and here made one substitution that NeuralCoref missed. A key reason why Coreferee was selected was that it generated results at over an order of magnitude faster than NeuralCoref, which meant that it was practical for use on the large SQuAD dataset.

This preprocessing yielded a small increase in BLEU scores from 0.36 to 0.37. An example is shown below:

Previous approach:

> [Context]: His body arrived on April 2, and was interred later that day in a small chapel on the grounds of the Eisenhower Presidential Library.
>
> [Generated question]: When was his body delivered?

Coreference resolution approach:

> [Context]: Eisenhower body arrived on April 2, and was interred later that day in a small chapel on the grounds of the Eisenhower Presidential Library.
>
> [Generated question]: When did Eisenhower's body arrive at the presidential library?

5.4 Sentence splitting

A sentence can contain multiple pieces of information, presenting a challenge for question generation. The generated question may focus on the wrong part or it may contain unnecessary information from it. The resolved example in Section 5.3 was:

> John lives with James in London, and there are many things John likes about London.

Some of the possible grammatical constructions are:

> John lives with James
>
> John lives in London
>
> James lives London
>
> There are many things John likes about London

It was found in Section 5.2 that selecting the relevant sentence gave an improvement in the questions generated. From this, it was considered possible that selecting the relevant grammatical construction might yield further improvements.

As with coreference resolution, this is not a simple task and requires addition or subtraction of words as well as rearrangements. Inflection of verbs is also desirable for a grammatically correct answer. The code used for this task was 'ClauCy'[6] by Emmanouil Chourdakis, which in turn was based on 'ClausIE'[7].

An example is shown below, with the original question being "Who was the Chief of Staff who promoted Eisenhower?":

Previous approach:

> [Context]: Next, Eisenhower was appointed Assistant Chief of Staff in charge of the new Operations Division (which replaced WPD) under Chief of Staff General George C. Marshall, who spotted talent and promoted accordingly.
>
> [Generated question]: Who replaced WPD?

Sentence splitting approach:

> [Context]: Eisenhower was appointed under Chief of Staff General George C. Marshall, who spotted talent and promoted accordingly.
>
> [Generated question]: Who was the Chief of Staff General Eisenhower?

The generated question is ungrammatical, but the words chosen are closer than the previous approach, which was asking about irrelevant information.

However, this technique resulted in a small drop of the BLEU score from 0.37 to 0.36. Qualitative analysis of examples suggests this may be due to valid but incorrect questions, or hallucinating information. These may be due to this technique reducing the lengths of the contexts and thus the volume of text to fine-tune on.

[6] https://github.com/mmxgn/spacy-clausie
[7] https://gate.d5.mpi-inf.mpg.de/ClausIEGate/ClausIEGate/

5.5 Vocabulary reduction

There are often multiple ways to express similar concepts in text and the connection of these can be a challenge for models. For example, a context might state that “X was the author of Y” and the question may be “Who wrote Y?”. The link between the noun ‘author’ and the verb ‘wrote’ may be obvious to a human but may not be recognised by a machine. Preprocessing the context so that ‘author’ is converted to ‘writer’ may make this link clearer. Ideally, the context would be rewritten to ‘X wrote Y’ but this may not always be possible.

As well as in each context-answer-question group, it could be useful to reduce the vocabulary across the whole training set. For example, many dates are present and are often linked to ‘When’ or ‘In what year’ questions. As it is not the intention of the training set to teach GPT-2 when historical events happened but to associate years with time questions, all years could be replaced with one year. This repetition may then help strengthen the link. It would be possible to use tags, such as converting ‘1999’ into [YEAR], but then the GPT-2 model would only be able to learn about the tags from the training data and not from the extensive pretraining.

To do this, all unmodified contexts were tokenized to words using Spacy, then vectorized[8] using a Word2Vec model trained on the ‘glove-wiki-gigaword-300' corpus. Each word was classed by Spacy[9] into fine-grained part of speech tags, such as RBR for comparative adverb or RBS for superlative adverb. For each tag group, the word vectors were split into 10 groups by k-means clustering. The word closest to the centre vector of each group was then used as the representative word for that group.

Once the representative word for each group was found, all words in that group were replaced by it. When multiple words in a context were part of the same group, they would be replaced by the next most representative word and so forth. Each time a substitution was made, a key-value pair was recorded in a dictionary, allowing words in the generated question to be converted back to their originals.

Unfortunately, words a human may believe to be similar were not always in the same group, as the below example illustrates.

Original:

> Chopin went in 1835 to Carlsbad, where Chopin spent time with Chopin’s parents.

Reduced vocabulary:

> Chopin thought in 1725 to Carlsbad, where Chopin came fact with Chopin’s others.

The date has been modified, as intended. However, ‘spent time’ is going to produce different results to ‘came fact’, as GPT-2 would have seen few examples of the latter. Most examples on the internet appear to be ‘came fact-to-face’, which is an

[8] https://radimrehurek.com/gensim/auto_examples/howtos/run_downloader_api.html
[9] https://spacy.io/usage/linguistic-features#pos-tagging

error. Despite both being a verb then a noun, the combined meaning is completely different, and it is not reasonable to expect good results from GPT-2 when the inputs are nonsensical. Due to this conversion process not yielding the desired results, it was concluded that this process may have potential but was beyond the scope of this project and taken no further.

5.6 Question starts

The GPT-2 model is unlikely to generate the correct question if it asks the wrong type of question, such as ‘What’ instead of ‘Who’. On occasion, the generated question does retain the correct meaning, but it then still scores poorly on BLEU.

The first few words often determine the type of question, and are referred to as question starts here. Here, the first two words were investigated as this allowed a balance between how narrow the type of question was, and how many starts would be needed for a high volume of training data. It was found that there were 11,904 different starts to the questions (in this preprocessed version) of the SQuAD database. Many, such as ‘Who worshipped’ or ‘What wars’ appeared only once, and it is typically possible to rewrite these questions so they use a more common start. Therefore, the effects of reducing the dataset to only questions with common starts were investigated. The criteria were that it needed to preserve ample training volume, coverage of different question types, and variety so that students would not tire rapidly of the same style of sentences.

It was found that 50% of questions could be covered with 35 starts, which are shown in Table 1 below. Reducing the corpus in this manner resulted in a noticeable increase in the quality of questions generated, with the BLEU score going from 0.37 to 0.41.

First words	Count		First words	Count
How does	200		Where is	503
How is	202		Who did	507
What country	206		What do	561
When were	214		Where did	698
What has	219		What year	754
Where does	230		How much	768
What can	270		What does	1073
Why did	303		What type	1094
In which	336		What are	1147
On what	368		Who was	1454
What were	389		When was	1476
What percentage	404		What did	1814
How did	406		In what	2085
How long	456		When did	2192
Who is	460		How many	3411
What kind	464		What was	3653
Along with	480		What is	5846
Where was	482			

Table 1. The top 35 most common two word question starts in the SQuAD database.

5.7 GPT-2 hyperparameters

For this work a pre-trained GPT-2 model was chosen and then fine-tuned on the SQuAD dataset. This left three ways to affect the results, with the first being selection of the pre-trained model. Moving from the 124M parameter model to the 355M parameter model slightly raised the BLEU score from 0.41 to 0.42, but at the cost of an approximately 2.5x increase in training times. This increase would have made subsequent optimisation tasks impractical, so the smaller model was kept.

Balance is needed between keeping the language fluency of the original model and learning the structured prompt pattern used here. Although loss when training continued to decrease with training, overtraining was a concern. It can be seen in Figure 1 that performance on the validation set suffers after 3000 training cycles, likely due to overtraining. Following this result, all subsequent models used 3000 cycles.

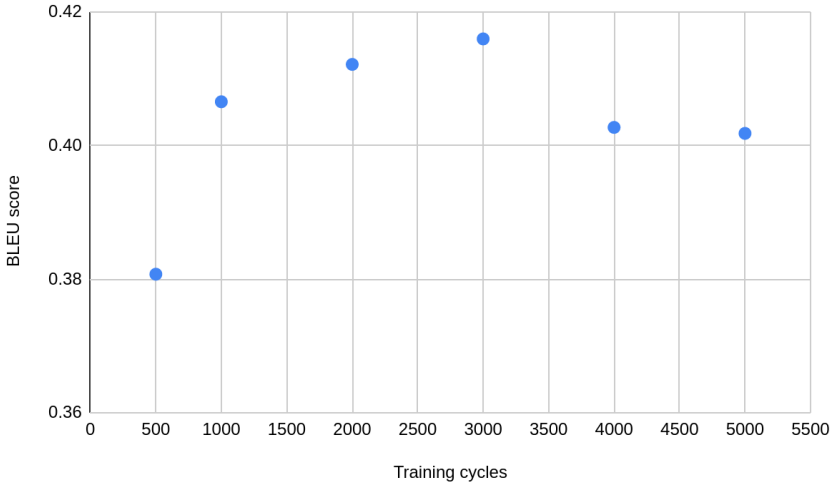

Figure 1. BLEU scores shown against number of training cycles for a GPT-2 model

The last hyperparameter was the temperature of the GPT-2 model, which controls the variety of results produced through an element of randomness in the sampling. In Figure 2, it can be seen that the average BLEU score for the generated questions (for the validation set) decreases with an increasing temperature. However, low temperatures increased repetition, reducing the chance for the ideal question to be present. For this experiment, 10 questions were generated each time and it can be seen that a higher temperature maximises the quality of the best question in the set.

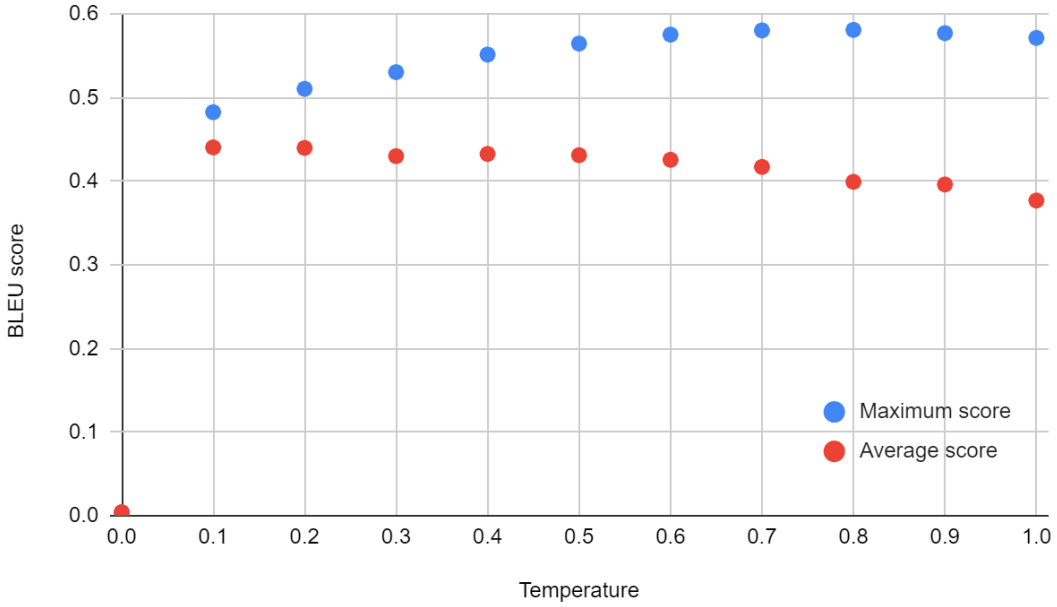

Figure 2. BLEU scores shown against temperature hyperparameter for the fine-tuned GPT-2 model.

In summary, if one question is to be selected without any filtering, the temperature should be set to 0.1, whereas if one has a perfect overgenerate and rank system, the temperature should be 0.8. The challenges involved in working towards this goal are discussed in Chapter 7.

5.8 Question generation results

Section	Name	BLEU score
5.1	Training on entire context	0.32
5.2	Training on single sentences	0.36
5.3	Coreference resolution	0.37
5.4	Sentence splitting	0.36
5.5	Vocabulary reduction	N/A
5.6	Question starts	0.41
5.7	GPT-2 hyperparameters	0.42

Table 2. BLEU scores for question generation using GPT-2 and preprocessing techniques.

Chapter 6

Experimental results - Answer selection

Questions were generated in Chapter 5 through prompts to GPT-2 consisting of a context tag, a context, an answer tag, an answer, and a question tag. The context comes from the original article and may be a sentence chosen using a method such as TextRank [39]. The answer, however, still needed to be selected automatically for the software to independently generate questions.

6.1 GPT-2 prompt modification

The approach chosen was to change the original prompts to skip the answer text and question tag. One example would be:

> [Context]: In other countries access to the website as a whole remains open, but access to specific videos is blocked.
>
> [Answer]:

This may appear absurd; it is difficult to computationally answer when the question is known, and seemingly impossible when it is not. It is a valid concern, but the aim here is not to find the correct answer but to select spans that a human might ask questions about. For the above example, the correct answer was “specific videos is blocked”, and the top options generated by the fine-tuned GPT-2 model were:

- as a whole
- remains open, but access to specific videos is blocked
- Access to specific videos is blocked
- in other countries
- as a whole remains open, but access to specific videos is blocked.

Three options (“still open”, “does not”, “ does not blocked”) were removed from the above list because they were not in the context. This is a downside of this approach, but enough possible options remain. These options appear reasonable and questions can generally be readily written for them such as “What is the condition of the website in other countries?” for the last entry.

Quantifying the results gives:

- 25% of suggestions were the exact answer

- 34% of suggestions had the answer as a substring
- 42% of sets, with 10 suggestions generated for each context, contained one suggestion that was the exact answer

It would appear that the authors of SQuAD were fairly predictable in their choices, resulting in a full quarter of answers being correctly guessed without even seeing the questions. Proper nouns and dates were some of the most reliably chosen.

To further investigate the similarity between human and GPT-2 selected answers, these were tagged using Spacy's POS tagger. For example, proper nouns (PROPN) were most frequently selected by both GPT-2 and the human authors. For the former, Spacy identified 17,474 proper nouns out of a total number of 56,556 tokens, meaning that 30.9% of all tokens generated by GPT-2 were proper nouns. The results are shown in Table 2, where it can be seen that there is a close match between the parts of speech in the original answers and in the generated answers.

POS	Original	GPT-2
ADJ	9%	8%
VERB	4%	3%
ADP	7%	7%
DET	7%	7%
NOUN	22%	21%
AUX	1%	1%
PROPN	27%	31%
PRON	1%	0%
CCONJ	4%	3%
NUM	9%	11%
PUNCT	6%	5%
ADV	1%	1%
PART	1%	1%
SCONJ	0%	0%
SYM	0%	1%
X	0%	0%

Table 3. Word types used in the original answers from the SQuAD dataset and in answers generated by a fine-tuned GPT-2 model.

Chapter 7

Experimental results - Ranking questions

GPT-2 has been used here to generate ten possible questions for each context-answer prompt. Comparing the BLEU scores of all the first questions (M = 0.418, SD = 0.207) with scores from the tenth questions (M = 0.417, SD = 0.214) using Welch's unequal variances *t*-test, shows there is no significant change with t = -0.16, p = 0.87. This means that the first is not any more likely to be correct than the last, and a method for selection was needed. Two options were considered, writing rules for selection, or training a language model for selection.

7.1 Rules

When the generated list was read through, several rules were immediately apparent, with the first being that the question should not contain the answer. This was achieved by looking at the intersection between the question and answer words, while ignoring any words in the top 1000 most used words.

Second, the question should not include unusual words that are not part of the context, as it is likely that the GPT-2 model has hallucinated them. Unusual was defined as outside of the top 10,000 words.

Lastly, significant repetition should be avoided in the question as this is highly unlikely to be correct. Minor repetition was defined as two words being repeated once. As this is uncommon but possible, it only resulted in a small score deduction. The minimum for major repetition was a two word phrase being repeated twice, or a three word phrase repeated once. The worst generated example featured a two word phrase repeated five times, illustrating the necessity of checking for this.

After these rules were implemented, it appeared that further rules would have to be significantly more advanced to make further progress. Instead the use of a language model was chosen.

7.2 LSTM

The role of the model was to evaluate the generated questions in isolation, with no link to either context or answer. For example, an adjective is more likely to occur

immediately before a noun than immediately after [49], and the presence of the latter may provide a small amount of evidence that the question could be nonsensical or not grammatically correct.

Recurrent neural networks are a well known model for time-series data [50], but suffer when attempting to predict relationships that occur over large distances, such as information in text [51]. One approach to deal with this is to use Long Short-Term Memory (LSTM) units, which provide a form of memory to the network [52].

To better understand their usage here, an LSTM was created using Keras and trained on sets of British place names[10]. These were then generated letter by letter. The results were relatively convincing and included ‘Bringham’, ‘Whitnock’ and ‘Pikecote Hill’, none of which exist. This process illustrated the need to monitor overtraining through looking at divergence in training and validation loss, and the methods of controlling it through dropout rate and model complexity.

For generated question ranking the LSTM model here was trained from scratch on the questions in the SQuAD dataset. This corpus is small compared to the one used to train GPT-2 and so it was possible that relationships between words would be missed through lack of examples. To address this, each word was converted using Spacy’s POS tagger. For example, ‘his’ became [PRP$], which was then mapped to a single character 'Ø'. This process meant that a generated question (‘How do you disagree with a YouTube decision?’) became a string of symbols ('ɜÀ8ţĤèî'). From these strings, the LSTM could be trained in exactly the same way as for the place names.

Once the model was trained, instead of generating new strings, the probabilities for each character were calculated. The geometric mean of these probabilities was then calculated and used to rank each generated question. For example, a question of ‘Why is the sky blue?’ may have individual word probabilities given by the LSTM of 0.4, 0.1, 0.2, 0.06, 0.16. The score would then be:

$$S = \left(\prod_{i=1}^{n} x_i \right)^{1/n} = (0.4 \times 0.1 \times 0.2 \times 0.06 \times 0.16)^{1/5} = 0.15$$

The highest and lowest ranked questions are shown in Table 3. It can be seen that four out of five low ranked questions are unsatisfactory, whereas only one of the top five has an error, which is quite minor.

Plotting the LSTM score against BLEU scores is a way of exploring agreement between them. It can be seen in Figure 3 that the correlation is low, but it is positive. Visible differences exist in average LSTM scores between BLEU scores of 0.0 and 1.0.

Furthermore, the top question selected by BLEU and by LSTM was the same 17.3% of the time, which is a modest but useful increase over the 10% found when choosing randomly.

[10] https://www.data.gov.uk/dataset/4949c88e-89b7-49b5-a0cf-8a3a2a4dac9d/os-open-names

Generated question	LSTM score
What is too many consumers trying to save (or pay down) because of?	0.02
Context: After selecting Truman budget director, Joseph M. Dodge	0.02
What was the highest industrial output percentage of all 28 EU countries>	0.02
What is it called when too many consumers are saving (or paying down) their debt?	0.02
What is too many consumers trying to save (or pay down) to?	0.03
....	….
What is the main language of Armenia?	0.60
What is another name for the President JK Bridge?	0.60
Who is the leader of the minority party?	0.60
What is the primary flag-carrier in Portugal?	0.61
What is another term for Portuguese late Gothic?	0.61

Table 4. Questions generated using a GPT-2 model fine-tuned on the SQuAD dataset, ranked by geometric mean of probabilities from an LSTM model trained on the same dataset.

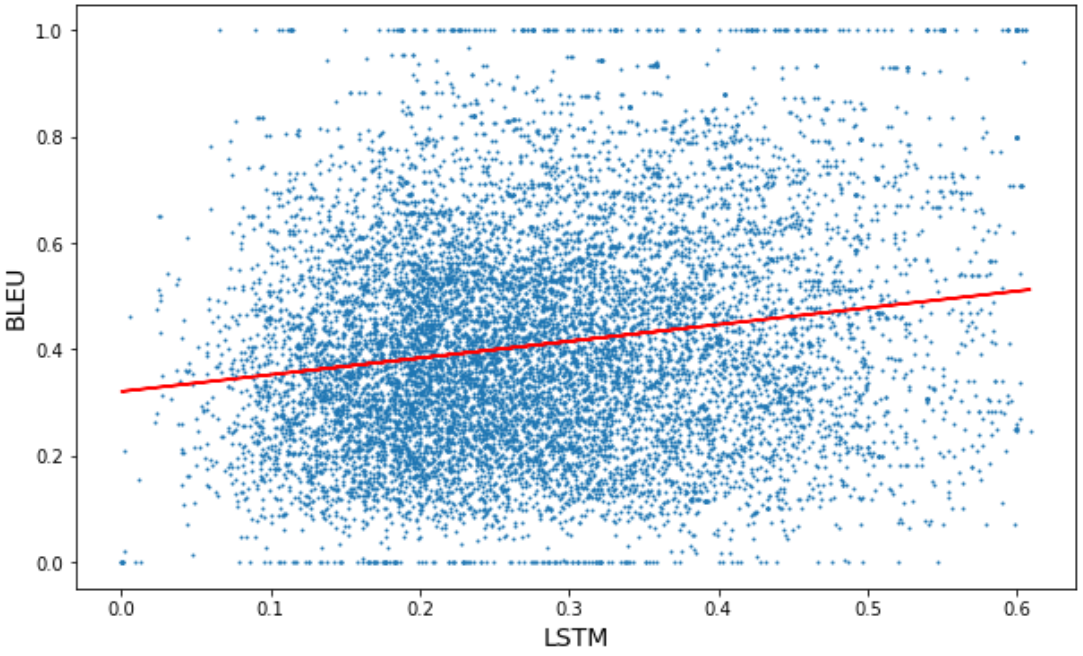

Figure 3. The BLEU score was calculated between original and generated questions. This was compared to the geometric mean of probabilities of each word in the generated question, provided by a trained LSTM RNN. The line of best fit is in red.

Chapter 8

Conclusions

In this experimental work, the central aim was to produce code that could take a context sentence, select a span from it to be the answer, and then generate a suitable question. For this aim, combinations of the SQuAD dataset, GPT-2 language models, and preprocessing techniques were explored. To evaluate the degree of similarity between the original and generated questions, a single BLEU score was calculated, as defined in Section 2.4.

The experiments performed for question generation are listed below::

- An improvement was found by reducing the input context down to the relevant sentence, as this avoided the likelihood of the question asking about the wrong sentence.
- Further improvement came from applying neural network based coreference resolution techniques, which removed ambiguities, particularly when discussing people.
- Reducing the length of the input further, through splitting sentences into grammatical constructions and selecting the relevant one, led to a small decrease in BLEU score. This may have been due to the reduction in fine-tuning training volume.
- Vocabulary reduction was attempted, with the concept being that a group of similar words would be replaced by a single word. Word vectors were clustered, but it could be seen that the process did not adequately group similar words and so this approach was abandoned.
- Although there were numerous two word starts to the questions in the dataset, it was found that half the questions could be retained by selecting only the top 35 most common starts. This greatly reduced the number of patterns encountered by GPT-2 when fine tuning and resulted in a significant BLEU score increase.
- GPT-2 fine-tuning hyperparameters were explored, finding that 3000 training cycles gave the highest performance without overtraining. A temperature of 0.8 was ideal when 10 questions were generated then ranked. For a single question, a temperature of 0.1 gave better results.

A single experiment was tried for answer generation:

- The previous prompt structure of context-answer-question groups was modified to leave context-answer pairs. GPT-2 was fine-tuned on this, and then used to generate answers in a test set where only the context was

provided. It was found that 25% of the generated answers were identical to the original and the types of words used in each were very similar. This suggests that this approach can be relied on to choose important spans, which then can be used for interesting and relevant questions.

Two approaches were explored for ranking the generated questions:

- Basic rules were considered, with a scoring system. It was not uncommon to see the answer in the generated question, so this needed to be avoided. Inclusion of new and unusual words in the question suggests GPT-2 has hallucinated information, which reduced the score. Repetition of multiple word phrases strongly indicated that the question was unlikely to be correct.
- Although those simple rules were useful, producing a comprehensive set would have been very challenging. Instead, for all the questions in the dataset, each word was converted into part-of-speech tags, such as ADJ, or PUNCT, and then into single characters. This was then used to train an LSTM which assessed the likelihood of each generated question. When compared to BLEU scores, the model picked the highest scoring question 17% of the time, a modest increase on the 10% expected by chance.

Taken together, the code from these experiments is capable of producing high quality questions from Wikipedia style input sentences. It could form the core of a question generation and assessment software package, if it were combined with sentence selection code, user answer assessment code, and a user interface.

Future work would involve returning to specific elements and attempting to increase their performance. These include:

- The coreference resolution approach could be combined with other approaches to resolve a greater number of sentences.
- Splitting the sentences into grammatical constructions did not result in the hoped for improvements, but it is likely that potential remains with this technique.
- Vocabulary reduction was an ambitious challenge, but significant improvements may be possible with different methods, such as using hypernyms. For example, mentions of ‘Mars, Saturn’ and ‘Jupiter’ could be replaced by ‘the planet’.
- The LSTM model can be trained further, for example on MS MARCO [32].

Lastly, extensions of this research would also include finding new metrics of similarity between generated questions and the originals. BLEU was used for this here and as it directly compares n-grams, this is inappropriate when synonyms are used. BLEU is also poorly suited to comparing questions that are rearranged or asked in a different way. As it is more important that high quality questions are generated than the exact phrasing of the original, finding a metric that assesses overall meaning would be an improvement. One option would be the BERT score[11] which uses embeddings to assess similarity.

[11] https://huggingface.co/spaces/evaluate-metric/bertscore

Word count

There are 9771 words in the main body of this report. This count excludes values in tables, reference numbers and the lengthy extracts from SQuAD on pages 14 and 16.

Bibliography

[1] L. Zhang and K. VanLehn, “Adaptively selecting biology questions generated from a semantic network,” *Interactive Learning Environments*, vol. 25, no. 7, pp. 828–846, 2017.

[2] Q. Guo, C. Kulkarni, A. Kittur, J. P. Bigham, and E. Brunskill, “Questimator: Generating knowledge assessments for arbitrary topics,” *IJCAI*, vol. January, pp. 3726–3732, 2016.

[3] I. V. Serban *et al.*, “A Large-Scale, Open-Domain, Mixed-Interface Dialogue-Based ITS for STEM,” in *Lecture Notes in Computer Science*, vol. 12164, Bittencourt, I., Cukurova, M., Muldner, K., Luckin, R., Millán, E., Ed. Springer, 2020.

[4] K. E. Boyer and P. Piwek, Eds., “Proceedings of QG2010: The Third Workshop on Question Generation,” Pittsburgh: questiongeneration.org.

[5] L. Becker, R. Nielsen, I. Okoye, T. Sumner, and W. Ward, “What’s next? Target concept identification and sequencing,” *Proceedings of QG2010*, [Online]. Available: http://www.cs.ecu.edu/gudivada/research/papers/Third%20Workshop%20on%20Question%20Generation%20QG2010-Proceedings.pdf#page=40

[6] M. Liu, R. A. Calvo, A. Aditomo, and L. A. Pizzato, “Using Wikipedia and Conceptual Graph Structures to Generate Questions for Academic Writing Support,” *IEEE Transactions on Learning Technologies*, vol. 5, no. 3, pp. 251–263, 2012.

[7] S. N. Heilman M, “Good question! Statistical ranking for question generation,” in *NAACL HLT 2010 - Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational*

Linguistics, pp. 609–617.
[8] X. Du, J. Shao, and C. Cardie, “Learning to ask: Neural question generation for reading comprehension,” presented at the Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, 2017. doi: 10.18653/v1/p17-1123.
[9] M. Heilman and N. A. Smith, “Question Generation via Overgenerating Transformations and Ranking.” 2009. doi: 10.21236/ada531042.
[10] E. Kochmar, D. D. Vu, R. Belfer, V. Gupta, I. V. Serban, and J. Pineau, “Automated personalized feedback improves learning gains in an intelligent tutoring system,” in *Lecture Notes in Computer Science*, Cham: Springer International Publishing, 2020, pp. 140–146.
[11] A. B. H. de Bruin, M. Sibbald, and S. Monteiro, “The science of learning,” in *Understanding Medical Education*, Chichester, UK: John Wiley & Sons, Ltd, 2018, pp. 23–36.
[12] C. Lynch, K. D. Ashley, N. Pinkwart, and V. Aleven, “Adaptive Tutoring Technologies and Ill-Defined Domains,” in *Adaptive Technologies for Training and Education*, P. J. Durlach and A. M. Lesgold, Eds. Cambridge University Press, 2012, pp. 179–203.
[13] P. Zaphiris and A. Ioannou, Eds., *Learning and collaboration technologies: Second international conference, LCT 2015, held as part of HCI international 2015, Los Angeles, CA, USA, august 2-7, 2015, proceedings*, 2015th ed. Cham, Switzerland: Springer International Publishing, 2015.
[14] M. Heilman, “Automatic Factual Question Generation from Text,” *Dissertation*, no. June, p. 203, 2011.
[15] S. De La Chica, F. Ahmad, J. H. Martin, and T. Sumner, “Pedagogically useful extractive summaries for science education,” *Coling 2008 - 22nd International Conference on Computational Linguistics, Proceedings of the Conference*, vol. 1, no. August, pp. 177–184, 2008.
[16] S. Subramanian, T. Wang, X. Yuan, S. Zhang, Y. Bengio, and A. Trischler, “Neural models for key phrase extraction and question generation,” *arXiv*, 2017, doi: 10.18653/v1/w18-2609.
[17] W. Chen, G. Aist, and J. Mostow, “Generating Questions Automatically from Informational Text,” *Workshop Proceedings of the AIED 2009: 14th International Conference on Artificial Intelligence in Education*, no. 1, pp. 17–24, 2009.
[18] E. Sneiders, “Automated question answering using question templates that cover the conceptual model of the database,” *Lect. Notes Comput. Sci.*, vol. 2553, no. June 2002, pp. 235–239, 2002.
[19] A. M. Olney, A. C. Graesser, and N. K. Person, “Question Generation from Concept Maps,” *Dialogue & Discourse*, vol. 3, no. 2, pp. 75–99, 2012.
[20] K. Mazidi, “Infusing Automatic Question Generation With Natural Language Understanding,” *Dissertation University of North Texas*, 2016, [Online]. Available: https://digital.library.unt.edu/ark:/67531/metadc955021/m2/1/high_res_d/MAZIDI-DISSERTATION-2016.pdf
[21] L. Pan, W. Lei, T. S. Chua, and M. Y. Kan, “Recent advances in neural question generation,” *arXiv*, no. 3, 2019, [Online]. Available:

http://arxiv.org/abs/1905.08949
[22] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734, 2014.
[23] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,” *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, 2015.
[24] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.
[25] Y. Kim, H. Lee, J. Shin, and K. Jung, “Improving neural question generation using answer separation,” *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pp. 6602–6609, 2019.
[26] A. Vaswani *et al.*, “Attention is all you need,” *Adv. Neural Inf. Process. Syst.*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
[27] L. E. Lopez, D. K. Cruz, J. C. B. Cruz, and C. Cheng, “Transformer-based end-to-end question generation,” *arXiv*, 2020, [Online]. Available: http://arxiv.org/abs/2005.01107
[28] L. Dong *et al.*, “Unified Language Model Pre-training for Natural Language Understanding and Generation,” *arXiv*, no. NeurIPS, 2019, [Online]. Available: http://arxiv.org/abs/1905.03197
[29] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a Method for Automatic Evaluation of Machine Translation,” *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.
[30] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “SQuAD: 100,000+ Questions for Machine Comprehension of Text,” *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 2383–2392, 2016.
[31] A. Trischler *et al.*, “NewsQA: A Machine Comprehension Dataset,” *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 191–200, 2017.
[32] T. Nguyen *et al.*, “MS MARCO: A human generated MAchine reading COmprehension dataset,” *CEUR Workshop Proc.*, vol. 1773, no. Nips 2016, pp. 1–11, 2016.
[33] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, “RACE: Large-scale ReAding comprehension dataset from examinations,” *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, pp. 785–794, 2017.
[34] S.-M.-R. Beheshti, S. Venugopal, S. H. Ryu, B. Benatallah, and W. Wang, “Big Data and Cross-Document Coreference Resolution: Current State and Future Opportunities,” no. November, 2013, [Online]. Available: http://arxiv.org/abs/1311.3987
[35] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, “A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy,

Gate," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, Oct. 2019, pp. 338–343.
[36] T. Wolf *et al.*, "Transformers: State-of-the-art natural language processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2019, doi: 10.18653/v1/2020.emnlp-demos.6.
[37] M. Mayank, "Guide to fine-tuning Text Generation models: GPT-2, GPT-Neo and T5," *Towards Data Science*, Jul. 11, 2021. https://towardsdatascience.com/guide-to-fine-tuning-text-generation-models-gpt-2-gpt-neo-and-t5-dc5de6b3bc5e (accessed Aug. 04, 2022).
[38] I. V. Serban *et al.*, "Generating factoid questions with recurrent neural networks: The 30M factoid question-answer corpus," *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, vol. 1, pp. 588–598, 2016.
[39] Mihalcea and Tarau, "Textrank: Bringing order into text," *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, [Online]. Available: https://aclanthology.org/W04-3252.pdf
[40] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI*, [Online]. Available: https://life-extension.github.io/2020/05/27/GPT%E6%8A%80%E6%9C%AF%E5%88%9D%E6%8E%A2/language-models.pdf
[41] J. Seabrook, "Can a Machine Learn to Write for The New Yorker?," *The New Yorker*, The New Yorker, Oct. 04, 2019. Accessed: Aug. 04, 2022. [Online]. Available: https://www.newyorker.com/magazine/2019/10/14/can-a-machine-learn-to-write-for-the-new-yorker
[42] A. Hern, "New AI fake text generator may be too dangerous to release, say creators," *The guardian*. [Online]. Available: https://amp.theguardian.com/technology/2019/feb/14/elon-musk-backed-ai-writes-convincing-news-fiction
[43] J. S. Lee and J. Hsiang, "Patent claim generation by fine-tuning OpenAI GPT-2," 2020 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0172219019300766
[44] "Write With Transformer." https://transformer.huggingface.co/doc/gpt2-large (accessed Aug. 04, 2022).
[45] V. Ng and C. Cardie, "Improving Machine Learning Approaches to Coreference Resolution," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Jul. 2002, pp. 104–111.
[46] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end Neural Coreference Resolution," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Jul. 2017, [Online]. Available: http://arxiv.org/abs/1707.07045
[47] R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu, "Anaphora and coreference resolution: A review," *Inf. Fusion*, vol. 59, pp. 139–162, Jul. 2020.
[48] J. Yu, N. S. Moosavi, S. Paun, and M. Poesio, "Stay Together: A System for Single and Split-antecedent Anaphora Resolution," *arXiv [cs.CL]*, Apr. 12, 2021.

[Online]. Available: http://arxiv.org/abs/2104.05320
[49] “Post-positive adjective,” *Academic Dictionaries and Encyclopedias*. https://en-academic.com/dic.nsf/enwiki/7583579 (accessed Aug. 05, 2022).
[50] S. C. Quax, M. D’Asaro, and M. A. J. van Gerven, “Adaptive time scales in recurrent neural networks,” *Sci. Rep.*, vol. 10, no. 1, p. 11360, Jul. 2020.
[51] A. Mujika, F. Meier, and A. Steger, “Fast-slow recurrent neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/e4a93f0332b2519177ed55741ea4e5e7-Abstract.html
[52] Z. Yu, D. S. Moirangthem, and M. Lee, “Continuous Timescale Long-Short Term Memory Neural Network for Human Intent Understanding,” *Front. Neurorobot.*, vol. 11, p. 42, Aug. 2017.