Question Generation for the Portuguese Language

Rafael Alexandre da Encarnação Galhoz

Thesis to obtain the Master of Science Degree in

Computer Science and Engineering

Supervisors: Prof. Bruno Emanuel Da Graça Martins
Prof. Pedro Alexandre Simões dos Santos

Examination Committee

Chairperson: Prof. Nuno Miguel Carvalho dos Santos
Supervisor: Prof. Bruno Emanuel Da Graça Martins
Member of the Committee: Prof. Maria Luísa Torres Ribeiro Marques da Silva Coheur

October 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa

Acknowledgments

This dissertation marks the end of a very important and memorable time in my life. Although the last five years I spent studying at the Instituto Superior Técnico had their ups and downs, they are a time that I will cherish and hold close to my heart and will always cherish.

I would like to thank my family for being there for me throughout the years, for their friendship, support, and love, without whom this dissertation would not be possible.

I would also like to acknowledge my dissertation supervisors Professor Bruno Emanuel Da Graça Martins, Professor Pedro Alexandre Simões dos Santos, and Professor João Dias, that made this thesis possible by sharing their insight, support, and knowledge.

Last but not least, I want to thank all of my friends and colleagues who have supported me through both good and bad periods in my life and have helped me develop as a person. I'm grateful.

Thank you.

Abstract

Question generation is an important task in the effort to automatically process natural language data. It can be used as a component in the context of many significant problems such as automatic tutoring systems, improving the performance of passage retrieval or question answering models, and enabling chatbots to lead a conversation. Recent approaches leverage sequence-to-sequence models based on Transformers to achieve state-of-art results. However, most of these advances are still within the English language.

With this in mind this work focuses on the study and development different models based on the Transformer T5 architecture using both supervised and self-critical sequence training over a Portuguese translated version of the SQuAD 1.1 dataset. We compare the results obtained with English baselines, using automatic evaluation metrics. In the end it was possible to observe that the Portuguese models generate questions with lower quality and poorer syntax, although with automatic evaluation results comparable to the ones obtained in the English language models.

Keywords

Question Generation; Transformer; Self-Critical Sequence Training; Deep Learning; Natural Language Processing; Portuguese Language;

Resumo

A geração de perguntas é uma tarefa importante no esforço de automatizar o processamento de linguagem natural, e pode ser usado em várias tarefas relevantes, tais como sistemas de tutoria automáticos, melhorar a performance de modelos capazes de extrair excertos relevantes e responder a perguntas e também permitir a chatbots conduzir uma conversa. Desenvolvimentos recentes utilizam modelos sequencia para sequencia baseados na arquitetura de transformadores capazes de adquirir resultados do estado da arte, contudo estes progressos foram feitos maioritariamente na linguagem inglesa.

Com isto em mente, nós desenvolvemos diferentes modelos baseados na arquitetura de transformadores T5, usando treino supervisionado e treino sequencial autocrítico, utilizando uma versão portuguesa do conjunto de dados SQuAD v1.1 traduzida automaticamente. Comparamos os resultados obtidos com modelos linha de base na língua inglesa utilizando avaliação automática. No final é possível observar que os modelos portugueses geram questões com qualidade inferior e pior sintaxe, contudo com resultados automáticos comparáveis aos obtidos pelos modelos ingleses.

Palavras Chave

Geração de Perguntas; Transformador; Treino Sequencial Autocrítico; Aprendizagem Profunda; Processamento de Linguagem Natural; Linguagem Portuguesa;

Contents

List of Figures

List of Tables

1

Introduction

Contents

The goal of question generation is to generate valid and fluent questions according to a given textual paragraph. This is a crucial aspect of the effort to automatically process natural language data, and it can be used in many scenarios such as developing automatic tutoring systems (Shah et al., 2017), improving the performance of passage retrieval (Nogueira et al., 2019) or question answering models (Riabi et al., 2020), and enabling chatbots to lead a conversation.

Recent approaches to question generation have used sequence-to-sequence models based on Transformers, being often trained to generate a plausible question conditioned on an input document and a candidate answer span within that document (Lopez et al., 2021). Still, most of these approaches have been used only in the context of small experiments with English datasets.

This study advances over previous neural models for question generation in several directions at the same time focusing in the Portuguese language. This includes fine-tuning Transformer models such as T5 with the combined use of supervised and reinforcement learning for model training (i.e., combining the standard teacher forcing approach for maximum likelihood training, with policy gradient techniques to maximize rewards that estimate question quality and answerability) (Hosking and Riedel, 2019; Zhu and Hauff, 2021), and exploring decoding and/or initialization methods that promote diverse generations (Liu et al., 2020; Yue et al., 2021).

In terms of the experimental evaluation, it should be stressed that even for the English language there are currently no dedicated question generation datasets, and many authors have used the context-question-answer triples available in datasets such as SQuAD (Rajpurkar et al., 2016) and MS MARCO (Nguyen et al., 2016). The main focus of this work will be on question generation for the Portuguese language, resorting to the use of machine translation to convert context-question-answer triples datasets into Portuguese so that the resulting data can be used to inform model training, evaluate these models trained over the machine-translated data, and assess the quality of the questions generated by the model compared to other models trained over English data.

1.1 Contributions

This thesis is based around fine-tuning a state-of-art Transformer T5 model in Portuguese question generation, using a machine translated version of SQuAD v1.1, and evaluating the quality of the generated questions when compared to other English question generation models. We fine-tune our models using the teacher forcing approach for maximum likelihood training using the cross-entropy loss function, and also use self-critical sequence training to fine-tune an already trained question generation model using using three different model-based rewards.

In the end, we compare the results of our fine-tuned Portuguese question generation models with both English baseline models and state-of-art approaches developed by other authors. We observed that the trained Portuguese question generation models obtain scores in the automatic evaluation metrics similar to early English question generation models. We make our work available at https://github.com/VivaRafael/Question-Generation-for-the-Portuguese-Language.

1.2 Organization of the Document

The remainder of the thesis is structured as follows: Chapter 2 discusses the fundamental concepts for the understanding of the thesis. In Chapter 3 we present some question generation models made by other authors with interesting results, and also studies advancing question generation using reinforcement leaning. The Chapter 4 introduces the T5 architecture, the self-critical sequence training. and describes the experimental setup used to train the models. The Chapter 5 presents the obtained results and compares them to other state of the art models trained over English data. We conclude the document in Chapter 6 with an overview on the main achievements and a discussion on possibilities for future work.

2

Background

Contents

In this chapter, the necessary concepts needed to understand the thesis will be introduced, including the concept of machine learning with deep neural networks, the transformer architecture and similarity-based metrics to evaluate natural language generation.

2.1 Machine Learning with Deep Neural Networks

The basis of deep learning and neural networks is the perceptron, it is the simplest neural network with only one node. It corresponds to a linear model that can be described by a given input x, a weight vector w with the same size as x, and a bias term b, as shown next:

$$y = x \cdot w + b = \sum_{i=0}^{+N} (x_i w_i) + b, \tag{2.1}$$

where every input variable is multiplied by a certain weight determined during the learning process, summed, and then added a bias term. However, since the expression above strictly comprises a linear combination of variables it is only able to estimate a linear function. To fix this we can extend the model to be capable of approximating a non-linear function by incorporating multiple layers of perceptrons and adding an intermediate activation function so that the output of a perceptron is the application of the activation function on the linear combination y:

$$o = \Phi(y) = \Phi(x \cdot w + b). \tag{2.2}$$

In the first definitions of neuron models, the considered activation function Φ was the step function. Modern examples of activation functions used are ReLU ($\Phi(x) = max(0, x)$) and sigmoid ($\Phi(x) = 1/(1+\epsilon^{-x})$).

A neural network is composed of multiple layers of neurons, where each layer is connected to others. In the case of a fully connected feed-forward neural network, each neuron on one layer is connected to others on previous and subsequent layers, where the connection only allows the information to pass forward. An example of this type of neural network can be seen in Figure 2.1.

As a learning algorithm, the goal of neural networks is to reach a function that predicts a value as accurately as possible over a particular data. While training, the concept of loss is used to compare the disparity between the predicted label and the correct label, the loss is a function that measures the cost of a wrong prediction. For this reason, the goal of the training process revolves around minimizing the average loss over the training data, and to accomplish this we can avail of gradient calculus.

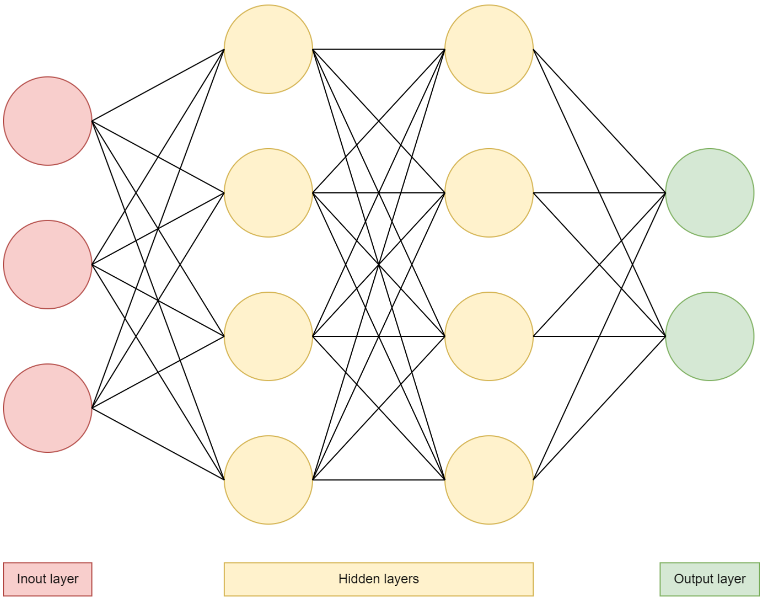

Figure 2.1: Example of a fully connected Feed-Forward Network.

Gradient descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. It repeatedly uses the derivative of the loss function on a point to calculate another point closer to a local minimum of the function until it reaches a local or global minimum of the function. The closer we get to a minimum the lower the disparity between the predicted and correct values.

To propagate the error to the weights from the output layer to other layers, so as to update the weights, we can use the back-propagation algorithm. This consists of executing a forward pass by processing the input, sending it to all layers until reaching the predicted output, and computing the prediction error using the selected loss function. This predicted error is then sent back through the neural network while each node updates its parameters with the gradient descent update rule, described formally by

$$\theta = \theta - \eta \nabla_\theta L(\theta), \tag{2.3}$$

where θ are the parameters of the model, and η is the learning rate, responsible for the size of the learning step, where a small value will lead to slow convergence and a value too big will lead to oscillations that won't allow the model to converge.

It is possible to distinguish three main variants of gradient descent, that differ in the amount of data used to calculate the gradient of the loss function, each one has its own benefits and drawbacks. The default variant is known as the batch gradient descent, where the gradient of the loss function is calculated once in every step, using the entire training set as a whole. This approach guarantees convergence to the global minimum (but only if the loss function is convex, otherwise it can converge to a local minimum).

Stochastic gradient descent calculates the gradient descent update using only one training example. This makes this update much more unstable and can possibly affect the convergence of the model. However, the update is much faster to compute.

Mini-batch gradient descent joins both approaches and calculates the gradient using small batches of the training set, previously divided, and performing the update similarly to the default variant. This allows the training procedure to maintain stabler gradients than the stochastic version.

2.2 Deep Learning for NLP and the Transformer Architecture

Deep learning is a crucial subset of machine learning and due to its power and flexibility, numerous neural network architectures were developed, each tailored to handle specific tasks such as object detection, speech recognition, and language understanding or generation.

In the field of natural language processing, recurrent neural networks were popular for dealing explicitly sequential data (in this case phrases and words) where each part of the sequence is dependent on the other. These networks are capable of achieving great results (Karpathy, 2015). However, by being only able to process information sequentially, they could not learn a high amount of long-term information, because of the vanishing and exploding gradient problems (Bengio et al., 1994), and have a big lack of performance both during training and inference due to being recursive. These networks became much less used upon the introduction of the Transformer architecture (Vaswani et al., 2017).

The Transformer architecture is a neural model based on an encoder-decoder structure where the encoder is designed to map an input sequence of symbols $(x_1, x_2, ..., x_n)$ to a sequence of continuous values $(z_1, z_2, ..., z_n)$. The decoder then picks this representation and uses it to generate an output sequence of symbols $(y_1, y_2, ..., y_n)$, one element at a time. The Transformer model follows the structure shown in Figure 2.2, using fully connected layers for both the encoder and the decoder while relying in self-attention.

To prepare both the input processed by the encoder module and the target output processed by the decoder module, we modify their representation using word embeddings, transforming each token of the sequence to a vector on an embedding space where tokens with similar meanings are close to each other, and tokens with disparate meaning are further. The original architecture, uses a pre-learned embedding model and the vector dimension is fixed to $d_{model} = 512$. Since the model does not contain recurrence or convolution, to utilize the order of each element sequence there is the need to add some information about the relative or absolute position of each token. To accomplish this result a position embedding is added to the input and output word embeddings. These encodings maintain the same

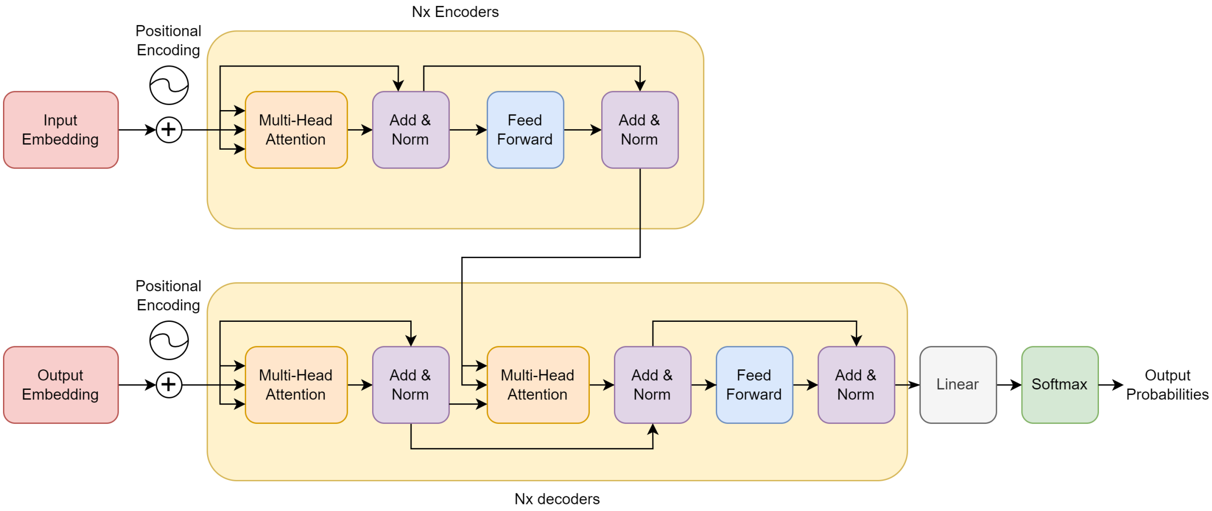

Figure 2.2: The Transformer model architecture.

dimension d_{model}, so they can be summed to the original embeddings and can be expressed as

$$\begin{aligned} \mathrm{PE}(pos, 2i) &= \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \\ \mathrm{PE}(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right), \end{aligned} \tag{2.4}$$

where pos represents the position and i represents the dimension.

A transformer encoder is composed of a stack of N identical layers (the original proposal uses $N = 6$), where each layer is subdivided into two sub-layers with different purposes. The first sub-layer is a multi-head self-attention mechanism that focuses on how surrounding positions affect the current one, and the second is a simple, position-wise fully connected feed-forward network. The model also uses a residual connection around the sub-layers, proceeded by normalization of each layer's result. This can be described by $\mathrm{LayerNorm}(x + \mathrm{SubLayer}(x))$.

A decoder module works in a similar way, but in addition to the two sub-layers in the encoder layers, it has another sub-layer responsible for performing multi-head attention over the output of the encoder stack. And also has a modification of the first sub-layer, adding a mask to the multi-head attention so that the different positions can only attend previous positions. This ensures that predictions can only depend on the known outputs.

The self-attention mechanism in the Transformer architecture allows it to understand context, producing a method that allows other relevant positions to influence the one we are currently processing, it can be described as mapping a query and set of key-value pairs to an output, where the arguments and output are vectors. The attention used in the transformer is called scaled dot-product attention and, to calculate it, we need to create a query vector, a key vector, and a value vector for each word embedding.

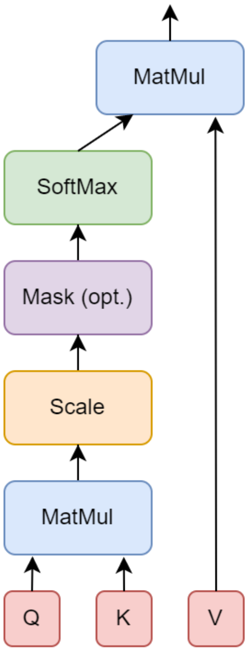

Figure 2.3: Scaled Dot-Product Attention.

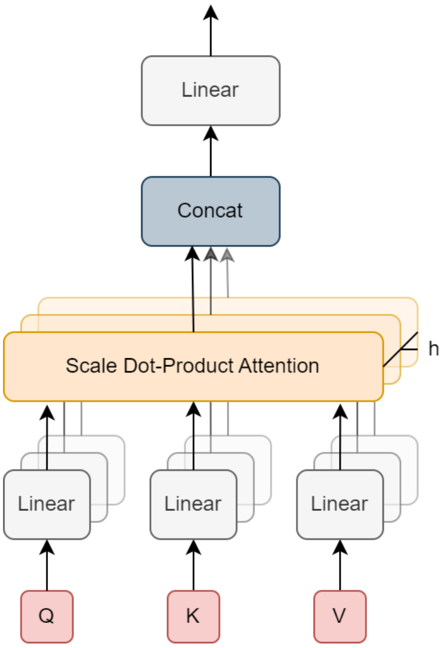

Figure 2.4: Multi-Head Attention.

These values are obtained by multiplying the token's embedding by weight matrices learned during the training process.

The scaled dot-product attention, described in Figure 2.3, can be calculated by multiplying the values with the weights that are computed by the dot product of the query with all the keys, dividing it by $\sqrt{d_k}$ and applying a softmax function. This can be expressed as

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2.5}$$

The multi-head attention, also described in Figure 2.4, extends the scaled dot-product attention by allowing the model to jointly attend the information from different representation subspaces at different positions, using h parallel attention layers. A multi-head can be described by:

$$\begin{aligned} \mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(head_1, head_2, ..., head_h)W^O, \text{with} \\ head_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V), \end{aligned} \tag{2.6}$$

where the projections used in each attention head are parameter matrices $W_i^Q, W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

2.3 Evaluating Natural Language Generation

The first evaluation metrics for natural language generation were based on n-gram similarity computation between the generated and some ground-truth sentences. The most common n-gram metrics to evaluate automatic question generation systems are BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE (Lin, 2004).

BLEU is a precision-based metric that considers exact n-gram matches. It is computed as the fraction of n-grams in the generated text matching the ones in the target text, where n is usually a number varying between 1 to 4.

METEOR uses both precision and recall to compute the n-gram matches. Unlike BLEU it also considers matches with stemmed words and synonyms. The final value is the harmonic mean of the precision and recall of the four types of matches.

ROUGE is a set of evaluation metrics. The most used is ROUGE-L, which can be characterized as the F-measure on the longest common subsequence between the generated and original sentence.

However recent studies (Hosking and Riedel, 2019; Nema and Khapra, 2018) have concluded that text similarity based only on n-grams metrics does not correlate well with human evaluation. For this reason, different and more useful metrics were created. Some examples of state-of-art metrics are CIDEr (Vedantam et al., 2014), BERTScore (Zhang et al., 2020), and BLEURT (Sellam et al., 2020).

CIDEr measures the similarity of the generated sentence against the ground truth sentence by matching n-grams and scaling them with TF-IDF (i.e. term frequency inverse document frequency is a technique to compute a score signifying the importance of a word in a set of documents). Generated sentences with more relevant words have higher scores.

BERTScore uses pre-trained contextual embeddings from a BERT model to match words in predicted and reference sentences using cosine similarity (i.e., the similarity between two vectors of an inner product space. It is computed by using the cosine of the angle between two vectors, determining the variation of direction between the vectors).

BLEURT is another BERT-based evaluation metric, capable of capturing non-trivial semantic similarities between sentences. The model is pre-trained using synthetic data with a language modeling objective, followed by pre-training again on natural language evaluation objectives, then finishing the learning process by fine-tuning it on high-quality human annotated evaluation data.

3

Related Work

Contents

This chapter explores recent works in the area, presenting some question generation models made by other authors with interesting results, and studies advancing question generation using reinforcement leaning.

3.1 Question Generation Models

Question generation is an important task of natural language processing and, when implemented properly, it allows a model to generate a valid, fluent, and relevant question according to a given textual paragraph. This section explores some recent works in the area, ranging from just fine-tuning a transformer language model on paragraph-level questions to using original decoding strategies to increase the diversity of the generated questions.

3.1.1 Fine-tuning GPT-2 on Paragraph-level Question Generation

Many question generation models and techniques employ complex model architectures and additional mechanisms to boost the model performance. However, Lopez et al. (2021) observed that fine-tuning a transformer-based architecture could produce results considerable better than model based on recurrent neural networks, and perform at a close level against other more complex sequence-to-sequence models with answer-awareness and additional distinct performance booster mechanisms.

Lopez et al. (2021) also showed how input data formatting can affect the performance of question generation models, by fine-tuning six GPT-2 (small) language models over different data preparations of the SQuAD dataset. These models were divided into two different approaches: (i) All Questions Per Line (AQPL) where a single training example consisted of a context paragraph followed by its questions, and (ii) One Question Per Line (OQPL) where a training entry had its context paragraph and only one question, this would result on longer training time since contexts are be duplicated. The two approaches there would be created three different models for each delimiter used to separate the context from the question, the ARTIFICIAL delimiter separated the components using an artificial separator designed as "[SEP]", the NATURAL-QUESTION delimiter used "Question:", and the NATURAL-NUMBER delimiter used "i." where i is the index of the question on the training example.

In terms of results, the OQPL model with the NATURAL-NUMBER delimiter achieved the highest score on almost all of the automatic metrics. It was also possible to observe that AQPL models performed consistently worse than their corresponding OQPL models. The best OQPL model, when compared to other state-of-art architectures, had worse performance in BLEU, METEOR, and ROUGE-L. However, due to the considerable less complexity, faster computing, and lower dimension of the model, it can be an interesting alternative for some particular cases.

Other compelling conclusions found after analyzing the generated questions are that 91.67% of the model generated questions are comprised of identification type questions (who/what/when/where), explained by the SQuAD dataset being composed of 88.26% of this type of questions. This frequency leads to the model learning more context-copying than other generation styles. The authors also evaluated the impact of the context length on question quality, reaching the conclusion that contexts closer to 10 sentences performed better and any increase in the number of sentences impacted negatively the generation due to making the subject of the context less apparent to the model. In this case, using answer-awareness (i.e. instead of using just the context to generate questions, we also use a previously chosen answer) was not able to help solve this problem, even making the results worse.

3.1.2 Answer-Clue-Style-aware Question Generation Model

Existing question generation models are not very successful at generating large amounts of high-quality question-answer pairs from certain contexts because of the large action space that allows a large amount of possible generated questions. To solve this problem Liu et al. (2020) proposed Answer-Clue-Style-aware (ACS-aware) question generation with the goal of generating high-quality and diverse question-answer pairs by simulating the way humans ask questions.

To establish the context for ACS-aware question generation, the author introduced a context composed of passage, answer, clue and style. A passage is a sequence of words considered in the question generation. An answer is a span in the input passage p. A clue denotes a chunk of the input p that will need to be included semantically in the question to be generated. A style represents the style of the future generated question. The nine question styles considered by the authors are “who”, “where”, “when”, “why”, “which”, “what”, “how”, “yes-no”, and “other”. The ACS-aware question generation process is then formalized by

$$\begin{aligned} \mathrm{P}(q,a|p) &= \sum_{c,s} \mathrm{P}(a,c,s|p)\,\mathrm{P}(q|a,c,s,p) \\ &= \sum_{c,s} \mathrm{P}(a|p)\,\mathrm{P}(s|a,p)\,\mathrm{P}(c|s,a,p)\,\mathrm{P}(q|c,s,a,p). \end{aligned} \tag{3.1}$$

Two different ACS-aware question generation models were tested: one based on a sequence-to-sequence recurrent neural network with attention and copy mechanism, where the answer, clue, and style are incorporated in the encoder and decoder (S2S-VR-ACS), and the second model based on fine-tuning a GPT-2 language model (GPT2-ACS). To guarantee a high quality of question-answer pairs generation, all generated pairs are passed through a filter, consisting of a BERT question answering model able to predict the answer span and compute the discrepancy with the input answer, and a classifier capable of measuring the entailment between the original sentence and the question-answer. The full model is then composed by the concatenation of the input sample (responsible for generating the answer-clue-style triplets for a given context entry), the ACS-aware question generator, and the data filter. The full model architecture can be seen in Figure 3.1.

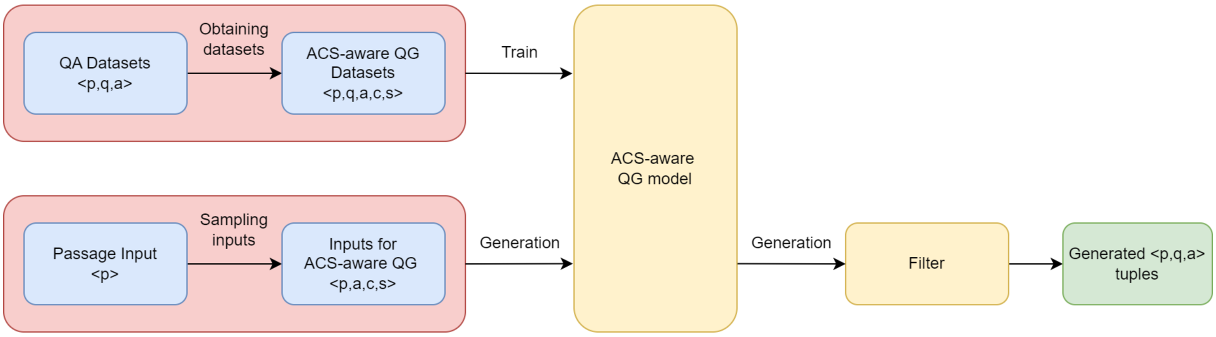

Figure 3.1: ACS-aware system architecture, containing an information sampler, ACS-aware question generation model, and a data filter.

Most existing question generation and question answering datasets are composed of context-question-answer triples and, for this reason, there is the need to use automatic strategies to extract effectively the clue and style from the respective entries. The process of extracting a clue from an input sample starts with parsing and chunking an input context to get all the candidate chunks. The context and question are then tokenized, stemmed, and filtered to only contain content words. Then for each candidate chunk, the total similarity score is computed by adding the score of three different similarities and adding them together, the number of overlapping context tokens, context stems and semantically coherent with each other (e.g. “age” and “old”) context tokens between the candidate clue and the question. In the end, it is chosen the candidate chunk with the highest total score as the clue c.

The algorithm proposed to classify a question regarding its style starts by verifying if the question contains any of the style words {who, where, when, why, which, what, how}. If it does, the question is classified as the corresponding type. For a yes-no type question the authors observe if the first word belongs to a set of features {am, is, was, were, are, does, do, did, have, had, has, could, can, shall, should, will, would, may, might} and if it does the questions is classified as the yes-no type, and otherwise labeled as other. With the extracted clue and style it is possible to create an ACS-aware dataset containing the context, question, answer, clue, and style to train the ACS-aware question generator models.

To generate a question based on a text corpus using an ACS-aware question generation model there is the need to sample an answer, a clue, and a style. To describe the sampling process, the following assumptions are made: (i) the probability of a chunk being selected as an answer only depends on its part-of-speech (POS) tags, named entity recognition (NER) tags, and the length of the chunk; (ii) the style of a question only depends on the POS and NER tags of the answer. (iii) the probability of a chunk being selected as the clue depends on the POS and NER tags and the distance between the chunk and the answer. Using these assumptions we have:

$$\mathrm{P}(a|p) = P(a|\,\mathrm{POS}(a), \mathrm{NER}(a), \mathrm{length}(a)), \tag{3.2}$$

$$\mathrm{P}(s|a,p) = P(s|\,\mathrm{POS}(a), \mathrm{NER}(a)), \tag{3.3}$$

$$\mathrm{P}(c|s,a,p) = P(c|\,\mathrm{POS}(c), \mathrm{NER}(c), \mathrm{distance}(c,a)), \tag{3.4}$$

where the distance between clue c and answer a is represented by the distance between their first tokens. It is possible to learn the conditional probabilistic distributions for an existing ACS-aware converted dataset and with this, it is possible to sample multiple answers, clues, and styles for a single passage. Since multiple questions will be generated for each context there is the need to ensure the quality of these questions, and that none is meaningless and unreasonable. To accomplish this goal, after the generation the authors apply a filter composed of two BERT models. The first is an entailment model capable of verifying if a question-answer pair matches with the associated context while the other confirms if a question is answerable. Using this filter, a particular sample question-answer will be accepted if the BERT-based entailment model classifies it as positive and if the F1 similarity score between the answer span and the answer span predicted by the BERT-based QA model is above a certain threshold.

Both the S2S-VR-ACS and GPT2-ACS models were then compared against other recent and state-of-art answer-aware question generation models, achieving great results on automatic metrics such as BLEU, ROUGE-L, and METEOR. The S2S-VR-ACS model achieved a considerable increase in BLEU, ROUGE-L, and METEOR metrics compared to other state-of-art models, having the best BLEU and ROUGE-L scores, while the GPT2-ACS achieved the best METEOR score, while also performing better than every other model (excluding S2S-VR-ACS) on BLEU.

When comparing GPT2-ACS to CS2S-VR-ACS, it is possible to observe that GPT2-ACS has a higher METEOR score, whereas CS2S-VR-ACS has higher BLEU and ROUGE-L scores. This happens because CS2S-VR-ACS employs vocabulary reduction, making the resulting words less versatile, which increase n-grams similarity metrics. These models were also evaluated manually with three criteria: (i) if the Question is well-formed, checking if it is both grammatical and meaningful; (ii) if the question is relevant to the associated context; (iii) if the answer is valid to the generated question. A sample is only assessed by the next criteria if the answer to the current criteria is positive. When comparing the results on these metrics the GPT2-ACS had significantly better results compared to the CS2S-VR-ACS model on the first and last criteria, having a slightly inferior scores on the relevant question criteria due to to the number of well-formed questions. GPT2-ACS model produced 74.5%, 19.5%, and 6% of well-formed, understandable, and not well-formed questions, 88.3% and 11.7% of relevant and irrelevant questions, and 81.3%, 15.1%, 3.6% of a correct, partial and incorrect answer.

3.1.3 Strategies to Generate Diverse Questions

Yue et al. (2021) proposed a new framework named CliniQG4QA with the goal of generating high-quality question-answer pairs to further train question answering models in a clinical environment where large-scale question-answer pairs are not readily available.

CliniQG4QA aims to enhance clinical question-answer pairs in previously unknown clinical contexts by automatically synthesizing new high-quality question-answer pairs for the new contexts. To accomplish this, the framework is divided in 3 parts, an answer evidence extractor responsible for extracting a relevant text span to be used as an answer, a question phrase prediction module able to predict a collection of question phrases, that reflect the types of questions humans ask associated with a particular answer evidence, and the question generation model that will complete the rest of the question given an answer and the question phrase. The complete model architecture can be seen in Figure 3.2.

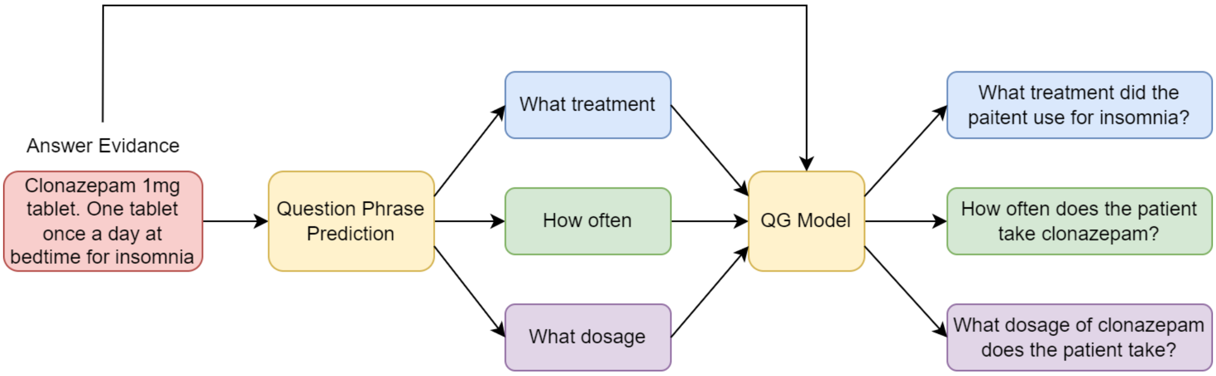
Answer Evidance
Clonazepam 1mg tablet. One tablet once a day at bedtime for insomnia
Question Phrase Prediction
What treatment
How often
What dosage
QG Model
What treatment did the paitent use for insomnia?
How often does the patient take clonazepam?
What dosage of clonazepam does the patient take?

Figure 3.2: CliniQG4QA architecture, containing an answer evidence extractor, question phrase predictor, and the question generation model.

The answer evidence extractor is used to extract possible answer spans from an input document. To select an answer from a document context C as a sequence labeling task, it is used the BIO (Ramshaw and Marcus, 1999) tagging to label answer evidences. Using the ClinicalBERT model (Alsentzer et al., 2019) (a clinical pre-trained BERT model) to encode the context:

$$\mathrm{U} = \mathrm{ClinicalBERT}(C). \tag{3.5}$$

And then applying on top of the hidden state's output of the ClinicalBERT a softmax layer to classify

$$\Pr(a_j|p_i) = \mathrm{softmax}(\mathrm{U} \cdot W + b), \forall c_i \in C, \tag{3.6}$$

where a_j is the estimated BIO tag.

Following the prediction, it is possible to see that some extracted answers are broken phrases, due to the naturally noisy nature and the use of acronyms in clinical texts. To get better and more useful extracted answers it is then computed, for each extracted text, a heuristic responsible for keeping the candidate answers if the length of the candidate surpasses a threshold η, or merging them with their closest candidate (in this case if the result of the merge is smaller than another threshold γ, this candidate is dropped). This heuristic rule is important because in the clinical domains longer answer spans are usually required due to them containing more useful information than a single named entity.

To generate more diverse types of questions, CliniQG4QA employs a question phrase prediction. First, it starts by collecting the first n-grams (proposing n=2) of the questions of the training data as V, then using a sequence to sequence model to map an answer evidence a to a vector representation of possible question phrases $y = (y_1, y_2, ..., y_L) \in \{0, 1\}^L$, where $y_i = 1$ indicates predicting $s_i \in V$ as a valid question phrase for the answer a. Allowing the model to be able to diversify and generate different questions for the same answer and context.

To evaluate the ability of the model proposed of diversifying question generation and improve the effectiveness of question answering models on new contexts, the authors train different question generation models and use them to fine-tune two different pre-trained question answering models DocReader (Chen et al., 2017) and ClinicalBERT (Alsentzer et al., 2019), to test their performance on the dataset used to train the question generation models. This dataset was created by the authors using human-generated pairs and machine-generated human-verified pairs, over the clinical contexts dataset MIMIC-III (Johnson et al., 2016). The question answering models were not exposed directly to this dataset during training and only to the question-answer pairs generated by CliniQG4QA.

The experimental evaluation consisted in comparing the results of question phrase prediction against other vastly used decoding strategies that are proven to increase results of a question generation model like beam search (Ippolito et al., 2019; Sultan et al., 2020), top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020), and verify if exposing the question answering models to the generated pairs could help the different answering models improve their performance on the new and unseen context.

It was possible to observe that more advanced any decoding strategies on the question generation models to train the question answering models, improved their performance and that using the question phrase prediction strategy to train the answering models made them outperform in exact match and F1-Scores the question answering models only pre-trained, and the ones fine-tuned using the base question generation model, beam search, top-k sampling, and nucleus sampling generated pairs. In the end, the authors also experimented with joining both Top-k and Nucleus sampling with question phrase prediction. However even if they saw an increase in the results of the sampling method alone, the results were worse than using only question phrase prediction.

3.2 Fine-tuning Question Generation Models using Reinforcement Learning

In a supervised learning environment, while training a question generation model there is often only one ground-truth question for each context-question-answer triple. Nevertheless, for each context provided, there are possibly multiple relevant facts that can be used to generate a question, with several syntactically correct compositions. Moreover in teacher-forcing learning, using the ground truth as the input, instead of the model output of a prior time step as an input when training, creates mismatches in generating the next action during training and testing. For these reason, likelihood-based training sufferers from the exposure bias problem (Ranzato et al., 2016).

Reinforcement learning can be used as solving these problems by optimizing the question generation model while training, using specific rewards and metrics that evaluate directly the quality of a specific question. Fine-tuning models based on policy gradient methods with rewards less correlated to the training data can counter the problems of teacher-forcing training. However, when applying it to real-life models, even with a steady increase in the metrics score, it results in poorer performance of the model when evaluated manually by humans. This is explained by Hosking and Riedel (2019) and Nema and Khapra (2018) by a lack of significant correlation between the automatic metrics used and human judgment, with the n-grams metrics like BLEU, ROUGE-L or METEOR having low to statistically insignificant values with $\rho < 0.10$.

Because of this lack of correlation Nema and Khapra (2018) proposed an improvement to existing n-gram metrics by making them account for answerability, noting that answerability depends mainly on the presence of relevant content words, named entities, question types and function words, defining $\mathrm{c}(S_r), \mathrm{c}(S_n), \mathrm{c}(S_q)$ and $\mathrm{c}(S_f)$ as the number of relevant words, named entities, question words, and function words on sentence S, respectively. They computed the weighted average of the precision and recall of each of these elements with:

$$\begin{aligned} \mathrm{P}_{\mathrm{avg}} &= \sum_i w_i \frac{\mathrm{c}(S_i)}{|l_i|} \\ \mathrm{R}_{\mathrm{avg}} &= \sum_i w_i \frac{\mathrm{c}(S_i)}{|r_i|} \end{aligned} \tag{3.7}$$

where $i \in r, n, q, f$, the sum of $w = 1$ and $|l_i|, |r_i|$ are the number of words belonging to the i^{th} type of element. With this, the authors combined the existing n-gram metrics (e.g. BLEU2) to create a new metric to evaluate questions as:

$$\text{Q-METRIC} = 2\delta \frac{\mathrm{P}_{\mathrm{avg}}\,\mathrm{R}_{\mathrm{avg}}}{\mathrm{P}_{\mathrm{avg}} + \mathrm{R}_{\mathrm{avg}}} + (1-\delta)\,\mathrm{METRIC}, \tag{3.8}$$

with $\delta \in]0,1[$ and METRIC as a given similarity-based metric. After tuning the weights w_i and δ using human-annotated data, there was a significant increase in correlation between the new metrics and human judgments, with Q-BLEU, Q-ROUGE-L and Q-METEOR having ρ values between 0.16 and 0.26, resulting in an increase of almost the double of the default automatic metrics values.

As learned word embeddings and contextual embeddings have been shown to provide better representations of lexical and semantic knowledge, several authors proposed using learned embeddings together with model-based metrics to optimize the correlation with human judgment while also maintaining focus on answerability and relevance to the context and ground-truth. Some examples of metrics worth mentioning are QPP and QAP (Zhang and Bansal, 2019), as well as other question-specific-rewards (Ling et al., 2020).

Here our focus will be related to recent model-based metrics with state-of-art results discussed and compared by Zhu and Hauff (2021). To compare these metrics the authors used a sequence-to-sequence model with a maxout pointer mechanism and gated self-attention network, similar to the one implemented by Zhao et al. (2018) using the Self-Critical Sequence Training (SCST) reinforcement learning algorithm (Rennie et al., 2016) to maximize the score of the question generated over a multitude of rewards.

The rewards evaluated by Zhu and Hauff (2021) are divided into four different categories. Fluency indicates if the generated question is valid according to the language model, similarity evaluates the similarity between the ground-truth and the generated question, answerabilty computes whether the generated question can be answered with the provided context and relevance indicates the relevance of the generated question over the context, answer, and ground-truth.

On the fluency category, the authors adopt the LM-based fluency rewards proposed by Xie et al. (2020). This reward value is calculated with the equation:

$$\mathrm{R}_{\mathrm{flu}} = -\exp\left(-\frac{1}{|Q|}\sum_{i=1}^{|Q|}\log \mathrm{M}_{\mathrm{flu}}(Q_i|Q_{<i})\right), \tag{3.9}$$

with $\mathrm{M}_{\mathrm{flu}}$ representing the fluency language model.

On the similarity category, due to the lack of quality of n-grams metrics on reinforcement learning the authors propose the use of two semantic-based similarity rewards, a BERTScore (Zhang et al., 2020) based reward and a QPP (Zhang and Bansal, 2019) based reward. BERTScore evaluates the similarity between the ground-truth and the generated question token by token using a similarity score.

To compute the values of precision, recall, and F1 score for BERTScore, we can use

$$\begin{aligned}
\mathrm{P}_{\mathrm{BERT}} &= \frac{1}{\hat{Q}} \sum_{\hat{y}_i \in \hat{Q}} \max_{y_j \in Q} \hat{\mathrm{y}}_i^T \, \mathrm{y}_j, \\
\mathrm{R}_{\mathrm{BERT}} &= \frac{1}{Q} \sum_{y_i \in Q} \max_{\hat{y}_j \in \hat{Q}} \mathrm{y}_i^T \, \hat{\mathrm{y}}_j, \\
\mathrm{F}_{\mathrm{BERT}} &= 2 \frac{\mathrm{P}_{\mathrm{BERT}} \, \mathrm{R}_{\mathrm{BERT}}}{\mathrm{P}_{\mathrm{BERT}} + \mathrm{R}_{\mathrm{BERT}}},
\end{aligned} \tag{3.10}$$

where $\hat{Q}$ represents the generated question, Q the ground-truth question, y_i is the i^{th} token in the question sequence and y_i is the pre-normalized contextual vector generated with BERT. The score used for this similarity reward is the F1 measure. For the QPP (o Question Paraphrasing Probability) reward (Zhang and Bansal, 2019), the authors proposed a BERT-based question paraphrasing classifier to provide the probability of the generated questions and the ground-truth being paraphrased. The score of the reward is the estimated probability multiplied by one hundred.

In the answerability category, the authors evaluated whether the generated question can be answered with the provided context. Two BERT question-answer based answerability rewards are proposed, namely one based on question-answer loss, BERT-QA-loss (Zhang and Bansal, 2019), and the other based on the geometric average of the question-answer probability, designed as BERT-QA-geo (Xie et al., 2020). The BERT-QA model outputs two probability distributions, namely the probability of each token of the C being the start of the answer, $\mathrm{P}^s_{ans} = \mathrm{P}(A^s|C, \hat{Q})$, and the probability of each token of C being the end $\mathrm{P}^e_{ans} = \mathrm{P}(A^e|C, \hat{Q})$. With that, it is possible to compute the cross-entropy loss of the predictions with

$$\mathrm{loss}(\mathrm{P}^s_{ans}, \mathrm{P}^e_{ans}, A) = \mathrm{CE}(\mathrm{P}^s_{ans}, A^s) + \mathrm{CE}(\mathrm{P}^e_{ans}, A^e), \tag{3.11}$$

and the reward for the BERT-QA-loss can be defined as

$$\mathrm{R}_{ans}(C, Q, A) = e^{-\,\mathrm{loss}}. \tag{3.12}$$

If a particular question is answerable based on a given context, the model should be able to find a start and end of a possible answer which would imply that those distributions would peak at a certain token. This means that if those distributions are somewhat constant, the model cannot find a valid possible answer. Because of this, we can use the geometric average of these distributions as a valid answerability heuristic reward:

$$\mathrm{R}_{ans}(C, Q) = \max_{1 \le i \le j \le T, j-1 \le i} \sqrt{\mathrm{P}^s_{ans}(i|C, Q) \, \mathrm{P}^e_{ans}(j|C, Q)}, \tag{3.13}$$

, where l is the maximum length of the answer.

Regarding the relevance category, there are multiple important components of question generation, the generated question, the context, the answer, and the ground-truth. For this reason, Zhu and Hauff (2021) proposed three different binary classifier-based discriminators capable of evaluating if a generated question is relevant to a fixed different combination of the essential components involved in question generation. The C-Rel reward evaluates if a generated question is relevant uniquely to the context. To accomplish this, the authors proposed a BERT binary classifier inspired by Xie et al. (2020). This model takes the context C and the generated question $\hat{Q}$ and computes the probability that $\hat{Q}$ is relevant to C. The CA-Rel reward adds the answer A to the previous reward. This classifier takes (C, A) and produces the probability of relevance between the $\hat{Q}$ and the pair formed by C and A. The CAQ-Rel reward evaluates if a generated question is relevant to both the context, answer and the ground-truth Q_g. The proposed CAQ-Rel classifier takes C, A, Q_g and $\hat{Q}$ and produces the probability of the generated question being relevant to these components.

In terms of automatic evaluation, Zhu and Hauff (2021) explored the performance of the different question generations models using n-gram similarity (BLEU, METEOR, and ROUGE-L) and all the proposed rewards as metrics. With this, it was possible to verify that all the models trained with reinforcement learning had better effectiveness over all the automatic metrics (except for the model trained with fluency rewards on the METEOR metric). It was also possible to observe that a model optimized over one reward also improved the scores on all the other rewards of the same type which implied some sort of correlation between rewards. Upon an extra investigation over the correlations of the rewards, it was possible to see that BERTScore and QPP were strongly correlated, that strangely enough the BERT-QA-loss reward and the BERT-QA-geo reward were almost independent, and that the BERT-QA-loss reward and the fluency reward are also not correlated to any other rewards.

In terms of human evaluation, the criteria used to evaluate the different models were syntax, relevance (to the context), and answerability. With this evaluation, it was possible to observe that the baseline outperformed in all aspects all the relevance based rewards even when these metrics lead to an improvement over the automatic rewards and that a model trained over the METEOR reward achieved better performance than the baseline over the three criteria which implies that it can be used as a computation efficient reward for reinforcement learning. It was also possible to verify that the BERT-QA-loss outperformed every other reward in the relevance and answerability criteria, achieving the second best score on the syntax metric. When comparing the correlation between the rewards scores and the human evaluation criteria it was observed that there is a lack of significant correlation between the two showing that these rewards functions behave in a vastly different way from the human rating.

4

Implementation

Contents

This thesis is based on the ideia of fine-tuning a state-of-art Transformer T5 model in Portuguese question generation, using a machine-translated version of SQuAD, and evaluating the quality of the generated questions when compared to other English question generation models. The model can be fine-tuned in two different ways: (i) using the teacher forcing approach for maximum likelihood training using the cross-entropy loss function, (ii) using self-critical sequence training to fine-tune an already trained question generation model using three different model-based metrics.

This chapter explains in some detail how the T5 Architecture works. It also introduces two different pre-trained variants of the T5 (Raffel et al., 2020) model (PTT5 (Carmo et al., 2020) and mT5 (Xue et al., 2021)) that we fine-tuned for Portuguese question generation, and introduce self-critical sequence training with the REINFORCE algorithm (Keneshloo et al., 2020). Additionally, we also describe the experimental setup used to train the question generation model.

4.1 The T5 Architecture

The exceptional performance of the transformer architecture when first introduced on machine translation (Vaswani et al., 2017) made it quickly noticed. Some authors realized that this architecture could be improved and applied to a wider range of tasks. With this several different architectures emerged such as BERT (Devlin et al., 2019), composed only by the encoder blocks of the transformer, GPT (Radford et al., 2018), which has only decoder blocks, and T5 (Raffel et al., 2020), that follows the traditional encoder-decoder transformer architecture.

T5, also known as Text-to-Text-Transfer-Transformer has the goal of unifying all-natural language tasks into a common text-to-text format, taking the text as input and outputting the new resulting text. The T5 model was pre-trained in the language modeling task on the C4 dataset. This task consisted in masking certain words in a paragraph with a masking token and sending them to the model with the goal of predicting what were the original words that were masked by the masked tokens. An example of this language modeling task can be seen in Figure 4.1. After the pre-training process, the T5 model is fine-tuned in various different tasks that include summarization, question answering, and text classification.

4.1.1 PTT5 and mT5

PTT5 is a T5 monolingual model pre-trained in the Brazilian Portuguese language. The training process starts from the original pre-trained T5 checkpoints and follows the same unsupervised pre-training implemented by T5, but with the Brazilian Portuguese BrWac (Wagner Filho et al., 2018) dataset, containing a large corpus of web pages in Brazilian Portuguese.

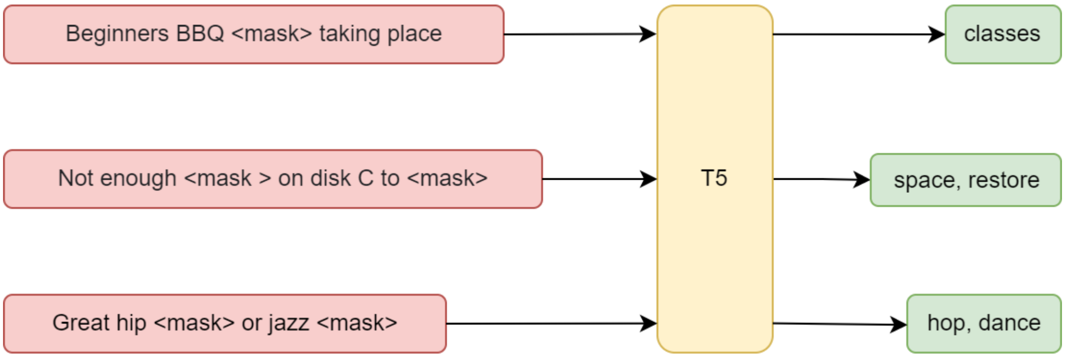
Beginners BBQ <mask> taking place
Not enough <mask > on disk C to <mask>
Great hip <mask> or jazz <mask>
T5
classes
space, restore
hop, dance

Figure 4.1: The unsupervised training process of T5.

The PTT5 model was also fine-tuned on some specific tasks. The first two tasks use the ASSIN 2 dataset (Real et al., 2020), composed of short Brazilian Portuguese sentence pairs and their respective semantic similarity and entailment relations, to allow the model to predict the semantic similarity and entailment between two different sentences. The last task uses the HAREM dataset (Santos et al., 2006), containing a collection of Portuguese-named entities, to allow the model to, given a Portuguese sentence, recognize Portuguese-named entities and their corresponding classes.

In contrast, mT5 (Xue et al., 2021) is a multilingual model, pre-trained with massive amounts of data from 101 languages, including Portuguese. It is only pre-trained through unsupervised language modeling using a massive multilingual version of the original C4 dataset, the mC4 corpus. In contrast to T5 (and PTT5), it is not fine-tuned on any specific tasks, and therefore needs to be fine-tuned before being ready for any specific multi-language task.

4.1.2 Fine-Tuning T5 for Question Generation

To fine-tune these models in the question generation task, we optimize the model parameters with the cross-entropy loss, used extensively in sequence-to-sequence models, with the goal of maximizing the log-likelihood over the training data.

During the question generation training, the model receives a tokenized input consisting of the answer and context to generate the predicted question tokens with a higher likelihood that will be used to compute the loss against the ground-truth. We apply the standard teacher-forcing strategy consisting of using the ground-truth (instead of the output of the previous sequence) when predicting the next sequence. An example of the question generation task can be seen in Figure 4.2.

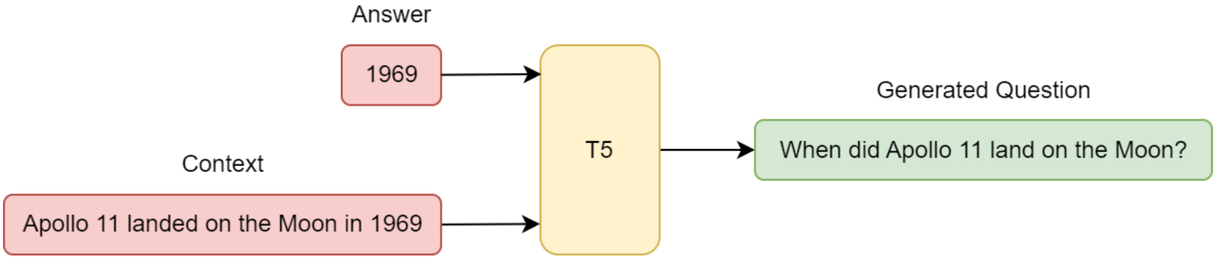
Answer
1969
Context
Apollo 11 landed on the Moon in 1969
T5
Generated Question
When did Apollo 11 land on the Moon?

Figure 4.2: An example of the question generation task using the T5 transformer.

4.2 Self-Critical Sequence Training

Transformer based models and other sequence-to-sequence models when using the standard teacher forcing algorithm (Bengio et al., 2015) (i.e. using the ground truth as the input, instead of the model output of a prior time step as an input when training) create mismatches in generating the next action during training and testing. This happens due to the model not having access to the ground truth data during the prediction.

This problem is regarded as the exposure bias problem (Ranzato et al., 2016) and leads to an error accumulation during prediction, conditioning the generated words to the ground truth instead of the previously generated words. An example of the difference between the training and testing process during teacher forcing can be seen in Figure 4.3.

Multiples proposals were suggested to solve this problem including scheduled sampling (Ranzato et al., 2016) and the use of adversarial generative models (Che et al., 2017; Guo et al., 2018; Su et al., 2018), however, our focus will revolve around the use of reinforcement learning on sequence to sequence models (Keneshloo et al., 2020; Paulus et al., 2017). For this reason, we resort to mixed strategy training where, after using teacher forcing, we train the model again using reinforcement learning via policy gradient, exposing the model to its own predictions and making the ground-truth available only for the reward calculation. There are multiple approaches to achieve this on sequence-to-sequence models. The one we considered relies on a policy-based reinforcement learning method using the REINFORCE algorithm (Keneshloo et al., 2020) to solve the training/testing evaluation mismatch problem.

In reinforcement learning, an agent chooses an action based on a specific policy π. On sequence to sequence models a parameterized policy π_θ can be represented as $\pi_\theta(y_t|\hat{y}_{t-1}, s_t, c_{t-1})$, where s_t represents the decoder and c_t the context at a time step t. When choosing its actions regarding a current policy the model observes the rewards only at the end when comparing the sequence of predicted actions $\hat{y}_t$ (using the current policy) against the ground-truth actions y_t using an evaluation metric. With this, the goal of the training process consists in finding the right parameters of the agent capable of

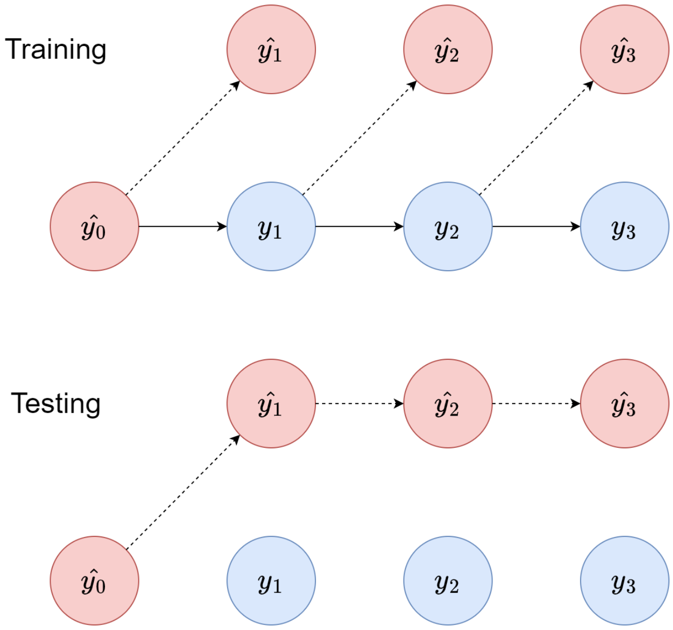

Figure 4.3: The difference between training and prediction during teacher forcing.

maximizing the expected reward and we can define this loss as the negative expected reward:

$$L_\theta = -\mathbb{E}_{\hat{y}1,...,\hat{y}T\sim\pi_\theta(\hat{y}_1,...,\hat{y}_T)}[r(\hat{y}_1, ..., \hat{y}_T)], \tag{4.1}$$

where $r(\hat{y}_1, ..., \hat{y}_T)$ represents the reward associated. The derivative of this loss can be calculated using a single sample from the distribution of actions of the sequence-to-sequence model as:

$$\nabla_\theta L_\theta = -\mathbb{E}_{\hat{y}1,...,T\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(\hat{y}_1...T) r(\hat{y}_1...T)], \tag{4.2}$$

or using the chain rule:

$$\nabla_\theta L_\theta = \frac{\partial L_\theta}{\partial \theta} = \sum_t \frac{\partial L_\theta}{\partial o_t} \frac{\partial o_t}{\partial \theta}, \tag{4.3}$$

$$\frac{\partial L_\theta}{\partial o_t} = (\pi_\theta(\hat{y}_t|\hat{y}_{t-1}, s_t, c_{t-1}) - \mathbf{1}(\hat{y}_t))(r(\hat{y}_1, ..., \hat{y}_T) - r_b), \tag{4.4}$$

where $\mathbf{1}(\hat{y}_t)$ constitutes a 1-of-$|A|$ representation of the ground truth output and r_b the baseline reward, with the goal of forcing the model to select actions that result in a higher reward than the baseline reward. One possible way to compute the baseline reward is to use greedy decoding and compute the reward obtained with the result, assuring that the reward obtained by our current model only gives positive rewards if the sample is better than the current output.

This represents the REINFORCE algorithm (Williams, 1992), i.e. a policy gradient algorithm that can be used on sequence-to-sequence problems. However one of the major problems of this algorithm is that it suffers from high variance, since it is calculated every time a new sample is used for training. To minimize this, it is possible to sample a batch of N sequences of actions at the same time in order to update the gradient, computing the average of these actions with:

$$L_\theta = \frac{1}{N}\sum_{i=1}^{N}\sum_{t}\log\pi_\theta(\hat{y}_{i,t}|\hat{y}_{i,t-1}, s_{i,t}, c_{i,t-1}) \times (r(\hat{y}_{i,1}, ..., \hat{y}_{i,T}) - r_b), \tag{4.5}$$

where r_b represents the baseline reward.

4.3 Experimental Setup

This section presents the baselines and the proposed question generation models, providing the implementation details behind the training process of the different models. We also describe the datasets used for both English and Portuguese language evaluation and the metrics used to assess the models.

4.3.1 Datasets

To train the different models proposed we leveraged the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) dataset, containing almost 100,000 diverse and high-quality question-answer pairs generated by human crowdworkers on more than 500 high-quality Wikipedia articles of different topics. The dataset is mainly composed of questions starting with “What” (≈57k), “Who” (≈10k), “Which” (≈7k), “When” and “How many” (≈6k), requiring responses that are in the form of places, objects, events, persons, dates, or numbers.

In the case of the English baseline models we used the SQuAD v1.1 original dataset. Due due to the lack of dedicated Portuguese question and answering datasets, and also the lack of resources to manually translate one of those datasets, we resorted to machine translation to train the models in Portuguese. For this, we used an already existing Portuguese machine-translated version of SQuAD v1.1, containing the translated context-question-answer triples that can be used for question generation.

We evaluate the performance of each model on the respective SQuAD datasets, by splitting each dataset into a training set with 80%, the validation set with 10%, and the test set with 10% of the data, making sure that an article is only in a set and that there are no different passages extracted from the same article in multiple sets. Also, we ensure that the distribution of the articles used in the splits is the same between both languages. An example extracted from the training split in both languages can be seen in Section 4.3.1.

Context: Independence was unilaterally declared on 24 September 1973. Recognition became universal following the 25 April 1974 socialist-inspired military coup in Portugal, which overthrew Lisbon's Estado Novo regime.
Question: Who was overthrown in the coup?
Answer: Lisbon's Estado Novo regime

Context: A independência foi declarada unilateralmente em 24 de setembro de 1973. O reconhecimento tornou-se universal após o golpe militar de inspiração socialista de 25 de abril de 1974 em Portugal, que derrubou o Regime Estado Novo de Lisboa.
Question: Quem foi derrubado no golpe?
Answer: Regime Estado Novo de Lisboa

Figure 4.4: An entry of the SQuAD v1.1 dataset in English and Portuguese.

4.3.2 Baseline Models

Our approach uses different baselines for each language. For the English language we utilize the original pre-trained T5 model architecture and mT5 (Xue et al., 2021) trained over the original SQuAD v1.1 dataset. For the Portuguese language we make use of two different pre-trained T5 models, namely PTT5 (Carmo et al., 2020) and mT5 (Xue et al., 2021) trained over the Portuguese translated SQuAD v1.1 dataset. For both T5, mT5, and the PPT5 models, we use the pre-trained base models available on the Hugging Face platform due to hardware constraints that would limit the use of larger models. We also compare our results with other models developed by previous authors.

4.3.3 Implementation Details

All the models are fine-tuned over the *base* versions of the respective pre-trained models using the same parameters, with the exception of the reinforcement learning models that use a smaller learning rate. For all the models we use a maximum input sequence length of 512 (the answer and the context after tokenization) and a maximum output sequence length of 96 (that corresponds to the question). We use a batch size of 16 over a maximum of 10 epochs, and we choose the model checkpoint that achieves the lowest validation cross-entropy error over the validation set. Regarding the optimization, we use the AdamW optimizer with Adam's epsilon of 1×10^{-6}, a learning rate of 1×10^{-4} for the cross-entropy training, and a learning rate of 1×10^{-7} for the reinforcement learning training. The decoding length during the inference process is equal to the maximum output token size during the training process and for the automatic evaluation results decoding we use the beam search strategy with 5 beams.

To train the *T5-base*, *PTT5* and *mT5-base* models (for both languages) the tokenization is done by sending the answer and context to the tokenizer, while for the *T5-base-tokens* and *PTT5-base-tokens* the tokenization of the input is done by first concatenating two extra tokens added to the tokenizer (<*answer*>, <*context*>) with the answer text and context text, i.e. "<*answer*> ..answer text.. <*context*> ..context text.." and then sending it to the tokenizer.

All the reinforcement learning models are trained over the *ptt5-base-tokens* changing the loss function from the cross-entropy to the one represented in Equation 4.5., where the reward of the sampled action is calculated by computing the reward over the question generated by multinomial sampling and the baseline reward is the reward of the question generated by greedy decoding. The *PTT5-base-bertscore* uses the BERTScore metric as the reward, the *PTT5-base-bleurt* uses the BLEURT metric as the reward, while the *PTT5-base-qa-loss* uses the BERT-qa-loss metric as the reward.

4.3.4 Metrics and Evaluation

To evaluate the models we use the standard automatic evaluation metrics of question generation models used in Liu et al. (2020); Xie et al. (2020); Zhu and Hauff (2021) and many others studies, in particular BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005). This allows us to compare our results to previous studies.

Due to the low correlation between these n-gram based metrics and human judgment described (Hosking and Riedel, 2019; Nema and Khapra, 2018) we also leverage some model base similarity metrics, specifically BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020). For BERTScore we use the default models for each language provided by Zhang et al. (2020), while for BLEURT metric we opted to use the BLEURT-20 for both languages.

Additionally, we also developed a metric based on answerability QAP (Question Answering Probability) (Zhang and Bansal, 2019) consisting of using a BERT-based question answering model able to predict the probability of a certain token being the start or end of the answer and computing the cross-entropy loss between the predicted and true answer by computing the sum of the cross-entropy for the start and end positions of the answers. The value of this metric is then given by $e^{-loss} \times 100$, where $loss$ is the loss returned by the BERT question-answering model. For the Portuguese language, we selected the BERT-based question answering model trained by Guillou (2021), and for the English language, we used the model trained by Sanh et al. (2019).

5

Results and Discussion

Contents

The automatic evaluation results on the test sets for the English language are given in Table 5.1, while the Portuguese results are given in Table 5.2. Both tables contain the results of some of the prior work on question generation developed by other authors and our trained question generation models.

In terms of general results for both languages, the ACS-aware question generation model developed by Liu et al. (2020) obtains the best results in BLEU1, BLEU2, BLEU3, and ROUGE-L. The noise-aware question generation model proposed by Xiao et al. (2020) achieves the highest score in the BLEU4 and METEOR metrics.

5.1 English Question Generation Results

Our fine-tuned English baseline models performed slightly worse on the different automatic evaluation metrics when compared against the previous state-of-art models. The baseline model with the best results is **T5-base-tokens** and has scores of 50.50 in BLEU1, 35.57 in BLEU2, 27.19 in BLEU3, 21.43 in BLEU4, 50.47 in ROUGE-L, 24.36 in METEOR, 91.97 in BERTScore, 54.72 in BLEURT and 69.64 in the QA-loss metric.

When comparing the **T5-base-tokens** results with the best ones in each metric, it is possible to observe a decrease of 1.80 in BLEU1, 1.13 in BLEU2, 0.81 in BLEU3, 3.97 in BLEU4, 2.77 in ROUGE-L and 2.56 in METEOR.

Table 5.1: Automatic evaluation of the English question generation models on the English test set. "BL" represents BLEU, "RL" represents ROUGE-L, "MTR" represents METEOR, "BERTS" represents BERTScore, "BLRT" represents BLUERT and QA-loss represents the Question Answerability metric. The values of prior works are removed directly from the original papers. Metric values not reported are displayed by "–". The best value of each metric is shown in bold.

Model	BL1	BL2	BL3	BL4	RL	MTR	BERTS	BLRT	QA-loss
Du et al. (2017)	43.09	25.96	17.50	12.28	39.75	16.62	—	—	—
Zhao et al. (2018)	43.47	28.23	20.40	15.32	43.91	19.29	—	—	—
Li et al. (2019)	45.66	30.21	21.82	16.27	44.35	20.36	—	—	—
Chan and Fan (2019)	49.73	34.60	26.13	20.33	48.23	23.88	—	—	—
Dong et al. (2019)	—	—	—	22.12	51.07	25.06	—	—	—
Xiao et al. (2020)	—	—	—	**25.40**	52.84	**26.92**	—	—	—
Qi et al. (2020)	—	—	—	25.01	52.57	26.83	—	—	—
Liu et al. (2020)	**52.30**	**36.70**	**28.00**	22.05	**53.25**	25.11	—	—	—
Zhu and Hauff (2021)	—	—	25.88	20.13	47.51	22.93	—	—	—
Leite and Lopes Cardoso (2022)	48.88	34.37	26.18	20.55	49.56	24.29	—	—	—
mT5-base	48.45	33.67	25.51	19.97	48.94	23.24	91.64	52.83	67.16
T5-base	50.01	35.32	27.01	21.26	50.35	24.23	91.95	54.54	68.73
T5-base-tokens	50.50	35.57	27.19	21.43	50.47	24.36	**91.97**	**54.72**	**69.64**

5.2 Portuguese Question Generation Models Results

Our Portuguese baseline models perform slightly better than the PTT5 model trained by Leite and Lopes Cardoso (2022), that followed the same training strategy and hyper-parameters (except the batch size which is 32). Our equivalent **PTT5-base** model records an increase of 1.36 in BLEU1, 1.63 in BLEU2, 1.57 in BLEU3, 1.34 in BLEU4 and 1.77 in ROUGE-L. This minor increase in all the metrics can be explained by the use of a different Portuguese translation of the SQuAD v1.1 dataset, where some issues created by the machine translation were fixed.

The Portuguese model with the best results in all the n-gram similarity metrics is **PTT5-base-tokens** with the scores of 45.70 in BLEU1, 32.23 in BLEU2, 24.61 in BLEU3, 19.29 in BLEU4, 45.50 in ROUGE-L, 32.80 in METEOR, 82.07 in BERTScore, 44.80 in BLEURT and 64.58 in the QA-loss metric. In contrast, the **PTT5-base-qa-loss** model performed the best in all the model-based metrics, including both model-based similarity metrics and the answerability metric, having scores of 45.51 in BLEU1, 32.07 in BLEU2, 24.48 in BLEU3, 19.18 in BLEU4, 45.41 in ROUGE-L, 32.66 in METEOR, 82.12 in BERTSCORE, 44.90 in BLEURT and 65.59 in the QA-loss metric.

When analyzing the Portuguese models and our English baseline models results, as excepted, it is possible to observe a considerable decrease in the results on all metrics. When comparing the Portuguese **PTT5-base-tokens** with the equivalent English **PTT5-base-tokens** models there is a difference of 4.8 in BLEU1, 3.34 in BLEU2, 2.58 in BLEU3, 2.14 in BLEU4 and 4.98 in ROUGE-L. This discrepancy is even higher when compared to the models with state-of-art results. This disparity between the English and Portuguese results can be explained due to the low quality of machine-translated data, where a brief analysis of the Portuguese-translated dataset can reveal multiple syntax errors, grammatical errors, and mismatches between the answer translation and the answer span in the context.

Table 5.2: Automatic evaluation of the Portuguese question generation models on the Portuguese test set. The values of prior works are removed directly from the original papers. Metric values not reported are displayed by "–". The best value of each metric is shown in bold.

Model	BL1	BL2	BL3	BL4	RL	MTR	BERTS	BLRT	QA-loss
Leite and Lopes Cardoso (2022)	43.61	30.04	22.58	17.54	43.64	—	—	—	—
mT5-base	44.78	31.33	23.82	18.60	44.71	31.94	81.80	43.56	63.52
PTT5-base	44.97	31.67	24.15	18.88	45.41	32.42	82.08	44.74	64.03
PTT5-base-tokens	**45.70**	**32.23**	**24.61**	**19.29**	**45.50**	32.80	82.07	44.80	64.58
PTT5-base-qa-loss	45.51	32.07	24.48	19.18	45.41	32.66	**82.12**	**44.90**	**65.59**
PTT5-base-bertscore	45.36	31.99	24.41	19.12	45.40	32.60	82.07	44.77	64.42
PTT5-base-bleurt	45.38	31.98	24.39	19.11	45.42	32.59	82.07	44.79	64.49

Overall, we observe that the results of the Portuguese generation models in the automatic evaluation metrics are comparable to the results of some earlier works developed in English question generation. However, when compared to English state-of-art models we can see a substantial decrease in performance in all the metrics. It is also possible to observe that trained unilingual models in Portuguese and English perform better than their corresponding multilingual models trained in the target language. Further analysis with a high-quality Portuguese question-answer dataset is necessary.

Regarding the self-critical sequence training results, in overall, the three models that were trained using this strategy performed slightly worse after the training process, decreasing the scores in almost all the automatic evaluation metrics. The only model that improved in some way was the **PTT5-base-qa-loss** model, which after training registered an improvement in all the model-based metrics, including its own metric used as a reward, the QA-loss. However, this improvement in the model-based metrics came at the cost of the scores in the other n-gram similarity metrics.

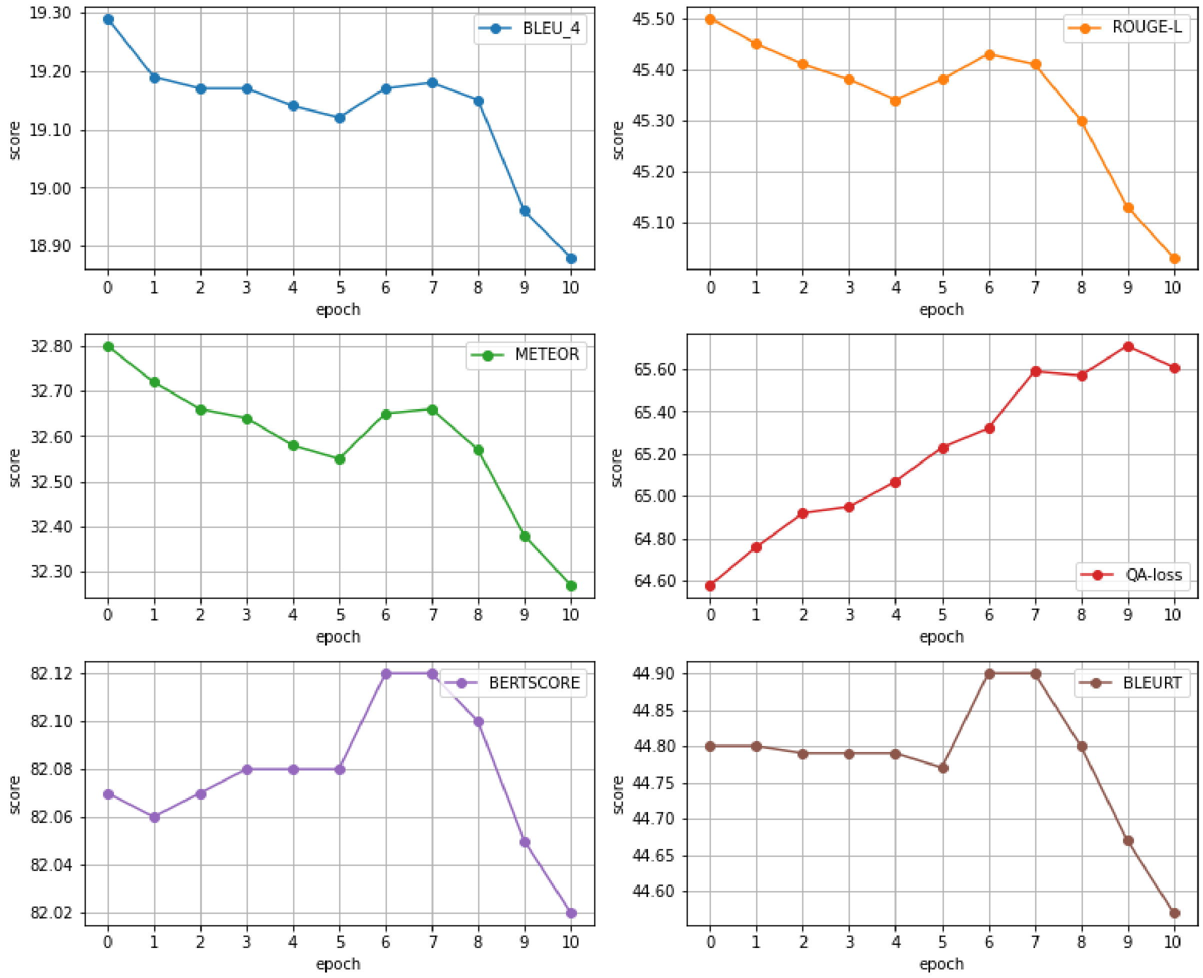

Figure 5.1: Automatic evaluation metrics scores of **PTT5-base-qa-loss** over the test set after a set number of epochs. The epoch number zero represents the initial state of the model before the self-critical sequence training.

Figure 5.1 represents the automatic evaluation results of the **PTT5-base-qa-loss** model over the test set after each epoch. In there it is possible to observe a slight decrease of the n-gram similarity metrics, in the beginning, stabilizing between epochs five and eight with a small increase, and then decreasing once again more abruptly. It is possible to observe a local maximum in epoch seven on the BLEU4 and METEOR, and in epoch six for ROUGE-L.The QA-loss metric improved roughly constant until epoch seven where it reached a local maximum, it decreased in epoch eight, reached the absolute maximum in epoch nine, and decreased again in the last epoch. The BERTScore and BLEURT metrics behaved very alike during training, maintaining the scores more or less stable until epoch five, where they increased to the highest values during epochs six and seven, and then decreased constantly until the end of the training.

We consider epoch seven the one with the best results of self-critical sequence training using the QA-loss reward, with scores of 45.51 in BLEU1, 32.07 in BLEU2, 24.48 in BLEU3, 19.18 in BLEU4, 45.41 in ROUGE-L, 32.66 in METEOR, 82.12 in BERTScore, 44.90 in BLEURT and 65.59 in QA-loss. This results in a decrease of 0.09 to 0.19 in the n-gram similarity metrics and an increase of 0.05, 0.10, and 1.01 in BERTScore, BLEURT, and QA-loss respectively

The other two models trained using self-critical sequence training, **PTT5-base-bertscore** and **PTT5-base-bleurt** performed poorly, and any training using the BERTScore or BLEURT as a reward resulted in a constant decrease in all the automatic evaluation metrics, including the metrics used as a reward For this reason, the values registered in Table 5.2 were recorded after only the first epoch during the training process.

5.3 Questions Generated by the Portuguese Models

In this section, we show some examples of the question generated by our model together with their results in the automatic evaluation metrics. To simplify we show only questions generated by the model with the best performance in the automatic evaluation, **PTT5-base-tokens**. More examples of generated questions by all the different Portuguese question generation models can be seen in Table A.1.

5.3.1 Questions Correctly Generated

Some examples of questions properly generated by the Portuguese model can be seen in Figure 5.2 with the respective evaluation metrics. We consider a question correctly generated if it follows three criteria: (i) it respects the syntax of the Portuguese language; (ii) it is relevant (i.e. a question which refers to a specific objective fact present in the context and does not allow any rationalization to find the answer); and (iii) it is answerable.

The first example of a generated question in the Figure 5.2 is almost equal to the ground-truth, changing only part of the predicate more specifically the words ”foi iniciada” by ”começou”. For this reason, this example has a high score in all the reported metrics.

The second generated question is also very similar in meaning to the original question however with more words, adding more constraints on the phrase such as ”família Bell” and ”Washington DC” instead of only ”Bell” and ”DC”, and also the verb used is different ”comprou” instead of ”adquiriu”. This makes most of the scores considerable lower than in the first example, reporting a BLEU4 score of 0.01.

The last example is the trickiest one because while the generated question is fluent, relevant, and answerable it is completely different from the ground-truth and therefore has almost the lowest possible scores in all the similarity metrics scores. Interestingly enough, since the question is well-formed and the answer provided is correct, the QA-loss score is very high, which helps confirm the value and quality of the generated question. This example helps reinforce that there is a disparity between similarity metrics scores and the quality of a generated question and that one can not fully assess the quality of a question generation model by only observing the automatic evaluation results.

Context: Segundo uma tradição relatada pela primeira vez por Sulcard em cerca de 1080, uma igreja foi fundada no local (então conhecida como Thorn Ey (Ilha Thorn)) no século VII, na época de Mellitus, um bispo de Londres. A construção da igreja atual começou em 1245, por ordem do rei Henrique III.
Answer: 1245
Ground-Truth: Quando foi iniciada a construção da igreja atual?
Generated: Quando começou a construção da igreja atual?
BLEU4: 62.40 **ROUGE-L**: 81.49 **METEOR**: 70.31 **BERTScore**: 97.49 **BLEURT**: 83.73 **QA-loss**: 98.28

Context: A casa da família Bell estava em Cambridge, Massachusetts, até 1880, quando o sogro de Bell comprou uma casa em Washington, DC, e mais tarde em 1882 comprou uma casa na mesma cidade para a família de Bell, para que pudessem ficar com ele enquanto ele participou de inúmeros processos judiciais envolvendo disputas de patentes.
Answer: 1882
Ground-Truth: Em que ano Bell adquiriu uma casa em DC?
Generated: Em que ano a família Bell comprou uma casa em Washington DC?
BLEU4: 0.01 **ROUGE-L**: 80.15 **METEOR**: 64.89 **BERTScore**: 86.61 **BLEURT**: 73.59 **QA-loss**: 99.75

Context: Em 1858, o imperador francês Napoleão III obteve com sucesso a posse, em nome do governo francês, da Longwood House e das terras ao seu redor, última residência de Napoleão I (que morreu ali em 1821). Ainda é propriedade francesa, administrada por um representante francês e sob a autoridade do Ministério de Relações Exteriores da França.
Answer: Napoleão I
Ground-Truth: Quem foi o último morador da casa de Longwood antes de Napoleão III assumir o controle?
Generated: Quem morreu em 1821?
BLEU4: 0.0 **ROUGE-L**: 16.55 **METEOR**: 4.29 **BERTScore**: 73.38 **BLEURT**: 15.59 **QA-loss**: 92.80

Figure 5.2: Example of correctly generated questions by the **PTT5-base-tokens** model.

5.3.2 Questions Generated with Semantic Errors

Some examples of questions generated by the Portuguese model with semantic errors that make the questions unanswerable or with an incorrect answer can be seen in Figure 5.3 with the respective evaluation metrics. In this case, we consider a question unanswerable if it can not be answered with the given context. A question with an incorrect answer is a question that can not be answered correctly with the answer given during the generation but can be answered based on the context.

The first example of a generated question in the Figure 5.3 is completely different from the ground-truth. However, the issue regarding this question is that the correct answer to ”A quem pertence o Museu Presidencial?” is not ”Abraham Lincoln” but ”estado de Illinois” or ”Illinois”. For this reason, the QA-loss score is low, because it was calculated using the wrong answer. Also, the results of the similarity metrics are very low due to the lack of similarity between the generated and original questions.

The second generated question represents a question that does not make sense and can not be answered based on the context provided. This makes this question unanswerable and consequently the answer incorrect, which makes the QA-loss almost 0. However, due to having some of the words contained in the ground-truth it has higher scores in the similarity metrics than the first example.

Context: Bibliotecas e museus foram estabelecidos para outros presidentes, mas eles não fazem parte do sistema de bibliotecas presidenciais da NARA e são operados por fundações privadas, sociedades históricas ou governos estaduais, incluindo Abraham Lincoln, Rutherford B. Hayes, William McKinley, Woodrow Bibliotecas Wilson e Calvin Coolidge. Por exemplo, a Biblioteca e Museu Presidencial Abraham Lincoln pertence e é operada pelo estado de Illinois.
Answer: Abraham Lincoln
Ground-Truth: Qual é o primeiro presidente que a NARA não possui registros em seu sistema de bibliotecas presidenciais?
Generated: A quem pertence o Museu Presidencial?
BLEU4: 0.0 **ROUGE-L**: 14.82 **METEOR**: 5.90 **BERTScore**: 74.48 **BLEURT**: 7.37 **QA-loss**: 4.30

Context:A corrupção também pode afetar os vários componentes das atividades de Esportes (árbitros, jogadores, equipe médica e de laboratório envolvidos nos controles antidoping, membros da federação esportiva nacional e comitês internacionais que decidem sobre a alocação de contratos e locais de competição).
Answer: Esportes
Ground-Truth: A corrupção pode determinar certas coisas em que tipo de atividades?
Generated: O que pode afetar a corrupção?
BLEU4: 0.0 **ROUGE-L**: 30.15 **METEOR**: 22.69 **BERTScore**: 81.1 **BLEURT**: 47.03 **QA-loss**: 0.04

Figure 5.3: Example of generated questions by the **PTT5-base-tokens** model with semantic errors.

5.3.3 Questions Generated with Syntax Errors

Some examples of questions generated by the Portuguese model with syntax errors can be seen in Figure 5.4 with the respective evaluation metrics. A question with syntax errors is a question that is not well-formed and has some sort of grammatical or lexical issue. A large number of questions generated by the model have syntax problems, so in these examples, we only show questions where the issue was so significant that the question could not be understood by looking at either the context, answer, or the respective ground-truth.

In the first example of the Figure 5.4, the question generated does not follow the correct syntax of the Portuguese language, making it confusing and almost impossible to answer it correctly. Nonetheless by looking at the beginning of the context and the given answer we can somewhat understand what the model tried to achieve. Due to the low similarity between the generated and ground-truth questions, the similarity metrics results are very low. Also, even with the syntax problems of the question, the QA-loss metric result was decent for this example.

Regarding the second example, the question generated also does not follow the correct syntax of the Portuguese language, and it is practically impossible to understand and answer correctly. Despite this, the QA-loss metric score of this question is very high, and also low scores on all the similarity metrics.

Context: Varejistas, fabricantes de artigos esportivos e outras empresas se beneficiam da luz solar adicional da tarde, pois induz os clientes a comprar e a participar de esportes ao ar livre da tarde. Em 1984, a revista Fortune estimou que uma extensão de sete semanas do horário de verão renderia US $ 30 milhões adicionais para as lojas 7-Eleven, e a National Golf Foundation estimou que a extensão aumentaria as receitas da indústria de golfe de US $ 200 milhões para US $ 300 milhões. Um estudo de 1999 estimou que o horário de verão aumenta a receita do setor de lazer da União Europeia em cerca de 3%.
Answer: artigos esportivos
Ground-Truth: Que categoria de produtos usados em atividades ao ar livre se beneficia da hora extra de luz do dia no horário de verão?
Generated: O que os fabricantes de?
BLEU4: 0.0 **ROUGE-L**: 18.05 **METEOR**: 4.61 **BERTScore**: 66.40 **BLEURT**: 3.60 **QA-loss**: 72.68

Context: Os dois países planejaram uma missão conjunta para atracar a última nave Apollo dos EUA com uma Soyuz, conhecida como Projeto de Teste Apollo-Soyuz (ASTP). Para se preparar, os EUA projetaram um módulo de ancoragem para o Apollo compatível com o sistema de ancoragem soviético, que permitia que qualquer de suas embarcações atracasse com qualquer outro (por exemplo, Soyuz / Soyuz e Soyuz / Salyut). O módulo também era necessário como uma câmara de ar para permitir que os homens visitassem a nave um do outro, que possuía atmosferas incompatíveis na cabine. A URSS usou a missão Soyuz 16 em dezembro de 1974 para se preparar para o ASTP.
Answer: Projeto de Teste Apollo-Soyuz
Ground-Truth: ASTP significa o que?
Generated: ASTP de quê?
BLEU4: 0.0 **ROUGE-L**: 21.79 **METEOR**: 12.63 **BERTScore**: 75.48 **BLEURT**: 48.35 **QA-loss**: 96.46

Figure 5.4: Example of generated questions by the **PTT5-base-tokens** model with syntax errors.

6

Conclusion

Contents

6.1 Summary and Contributions

The goal of this study was to address the current state-of-art of question generation while exploring how to implement these methods for question generation in a different language without a decent amount of natural language processing corpora. We explored fine-tuning a pre-trained state-of-art language model in Portuguese question generation using a machine-translated dataset. We achieved this by developing multiple baselines and models using both supervised and reinforcement learning and evaluating them using automatic evaluation metrics.

In the end, we conclude that the quantitative results obtained on the Portuguese models were comparable with earlier works made by other authors using dedicated high-quality question-answering datasets. Nonetheless, it was possible to observe issues regarding the use of a machine-translated dataset with poor quality to fine-tune a question generation model. For this reason, we emphasize the importance for the natural language processing community of the creation of dedicated Portuguese and other language question-answer datasets. Also, we want to reinforce that the automatic evaluation metrics based on n-grams used in this study have been proven to correlate poorly with human judgment, so a further human evaluation would provide a more accurate understanding of the quality of the generated questions.

6.2 Future Work

Although the automatic evaluation results fell under our initial expectations of using a machine-translated dataset, with results comparable to early English question generation models, there are still multiple ways capable of improving the results of question generation in the Portuguese language.

The most straightforward method to achieve this is to implement state-of-art strategies that improved the results of English question generation models. We detected two different approaches that generated great results in English and have the possibility of improving the results in Portuguese: (i) the ACS-aware question generation model proposed by Liu et al. (2020) that promotes the generation of more diverse question and (ii) the noise-aware question generation model proposed by Xiao et al. (2020) that implements a noise-aware generation method that strives to reduce the discrepancy between training and inference and therefore minimize the exposure bias problem. Additionally, it could be proven interesting to explore other pre-trained sequence-to-sequence models such as PEGASUS (Zhang et al., 2019) and BART (Lewis et al., 2019).

Another idea is to experiment using self-critical sequence training with other model-based text generation metrics or n-gram similarity metrics such as the ones introduced by Nema and Khapra (2018) that were not used in this work. Also, the use of other methods to solve the exposure bias problem such as scheduled sampling (Ranzato et al., 2016), the use of adversarial generative models (Che et al., 2017; Guo et al., 2018; Su et al., 2018) or the noise generation method Xiao et al. (2020) can be more effective than the use of self-critical sequence training.

Other possible and more reliable path to improve Portuguese question generation further is to improve the quality of Portuguese corpora for question generation and question answering, either by developing a rule-based system to clean up the data or manually by hiring human crowdworkers. This would significantly improve the results by reducing the number of errors on the translated dataset.

Bibliography

Alsentzer, E., Murphy, J., Boag, W., Weng, W.-H., Jindi, D., Naumann, T., and McDermott, M. (2019). Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1171–1179, Cambridge, MA, USA. MIT Press.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Carmo, D., Piau, M., Campiotti, I., Nogueira, R., and Lotufo, R. (2020). Ptt5: Pretraining and validating the t5 model on brazilian portuguese data.

Chan, Y.-H. and Fan, Y.-C. (2019). A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

Che, T., Li, Y., Zhang, R., Hjelm, R. D., Li, W., Song, Y., and Bengio, Y. (2017). Maximum-likelihood augmented discrete generative adversarial networks. *ArXiv*, abs/1702.07983.

Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., and Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation.

Du, X., Shao, J., and Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension.

Fan, A., Lewis, M., and Dauphin, Y. (2018). Hierarchical neural story generation.

Guillou, P. (2021). Portuguese bert base cased qa (question answering), finetuned on squad v1.1.

Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., and Wang, J. (2018). Long text generation via adversarial training with leaked information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The curious case of neural text degeneration.

Hosking, T. and Riedel, S. (2019). Evaluating rewards for question generation models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2278–2283, Minneapolis, Minnesota. Association for Computational Linguistics.

Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., and Callison-Burch, C. (2019). Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Sci Data*, 3:160035.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks.

Keneshloo, Y., Shi, T., Ramakrishnan, N., and Reddy, C. K. (2020). Deep reinforcement learning for sequence-to-sequence models. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2469–2489.

Leite, B. and Lopes Cardoso, H. (2022). Neural question generation for the portuguese language: A preliminary study. In Marreiros, G., Martins, B., Paiva, A., Ribeiro, B., and Sardinha, A., editors, *Progress in Artificial Intelligence*, pages 780–793, Cham. Springer International Publishing.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Li, J., Gao, Y., Bing, L., King, I., and Lyu, M. R. (2019). Improving question generation with to the point context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3216–3226, Hong Kong, China. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Ling, Y., Cai, F., Chen, H., and de Rijke, M. (2020). Leveraging context for neural question generation in open-domain dialogue systems. In *Proceedings of The Web Conference 2020*, WWW '20, page 2486–2492, New York, NY, USA. Association for Computing Machinery.

Liu, B., Wei, H., Niu, D., Chen, H., and He, Y. (2020). Asking questions the human way: Scalable question-answer generation from text corpus. *Proceedings of The Web Conference 2020*.

Lopez, L. E., Cruz, D. K., Cruz, J. C. B., and Cheng, C. (2021). Simplifying paragraph-level question generation via transformer language models. In Pham, D. N., Theeramunkong, T., Governatori, G., and Liu, F., editors, *PRICAI 2021: Trends in Artificial Intelligence*, pages 323–334, Cham. Springer International Publishing.

Nema, P. and Khapra, M. M. (2018). Towards a better metric for evaluating question generation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3950–3959, Brussels, Belgium. Association for Computational Linguistics.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Nogueira, R., Yang, W., Lin, J., and Cho, K. (2019). Document expansion by query prediction. *CoRR*, abs/1904.08375.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization.

Qi, W., Yan, Y., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R., and Zhou, M. (2020). ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for*

Computational Linguistics: EMNLP 2020, pages 2401–2410, Online. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ramshaw, L. A. and Marcus, M. P. (1999). Text chunking using transformation-based learning.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2016). Sequence level training with recurrent neural networks.

Real, L., Fonseca, E., and Oliveira, H. G. (2020). The assin 2 shared task: a quick overview. In *International Conference on Computational Processing of the Portuguese Language*, pages 406–412. Springer.

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., and Goel, V. (2016). Self-critical sequence training for image captioning. *CoRR*, abs/1612.00563.

Riabi, A., Scialom, T., Keraron, R., Sagot, B., Seddah, D., and Staiano, J. (2020). Synthetic data augmentation for zero-shot cross-lingual question answering. *CoRR*, abs/2010.12643.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC*$^2Workshop.$

Santos, D., Seco, N., Cardoso, N., and Vilela, R. (2006). Harem: An advanced ner evaluation contest for portuguese. In *quot; In Nicoletta Calzolari; Khalid Choukri; Aldo Gangemi; Bente Maegaard; Joseph Mariani; Jan Odjik; Daniel Tapias (ed) Proceedings of the 5 th International Conference on Language Resources and Evaluation (LREC'2006)(Genoa Italy 22-28 May 2006)*.

Sellam, T., Das, D., and Parikh, A. P. (2020). BLEURT: learning robust metrics for text generation. *CoRR*, abs/2004.04696.

Shah, R., Shah, D., and Kurup, L. (2017). Automatic question generation for intelligent tutoring systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, pages 127–132.

Su, J., Xu, J., Qiu, X., and Huang, X. (2018). Incorporating discriminator in sentence generation: a gibbs sampling method.

Sultan, M. A., Chandel, S., Fernandez Astudillo, R., and Castelli, V. (2020). On the importance of diversity in question generation for QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5651–5656, Online. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Vedantam, R., Zitnick, C. L., and Parikh, D. (2014). Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726.

Wagner Filho, J. A., Wilkens, R., Idiart, M., and Villavicencio, A. (2018). The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning.

Xiao, D., Zhang, H., Li, Y., Sun, Y., Tian, H., Wu, H., and Wang, H. (2020). Ernie-gen: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3997–4003. International Joint Conferences on Artificial Intelligence Organization. Main track.

Xie, Y., Pan, L., Wang, D., Kan, M., and Feng, Y. (2020). Exploring question-specific rewards for generating deep questions. *CoRR*, abs/2011.01102.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yue, X., Zhang, X. F., Yao, Z., Lin, S., and Sun, H. (2021). Cliniqg4qa: Generating diverse questions for domain adaptation of clinical question answering.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Zhang, S. and Bansal, M. (2019). Addressing semantic drift in question generation for semi-supervised question answering. *CoRR*, abs/1909.06356.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). Bertscore: Evaluating text generation with bert.

Zhao, Y., Ni, X., Ding, Y., and Ke, Q. (2018). Paragraph-level neural question generation with max-out pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Zhu, P. and Hauff, C. (2021). Evaluating bert-based rewards for question generation with reinforcement learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '21, page 261–270, New York, NY, USA. Association for Computing Machinery.

A

Appendix

Table A.1: Examples of generated questions using the developed Portuguese question generation models. The contexts, ground-truths and answers are all extracted from the Portuguese test set.

Example 1							
Context	O primeiro satélite do sistema de segunda geração, o Compass-M1, foi lançado em 2007. Seguiu-se mais nove satélites durante o período 2009-2011, alcançando uma cobertura regional funcional. Um total de 16 satélites foram lançados durante esta fase.						
Answer	16	**BL4**	**RL**	**MTR**	**BERTS**	**BLRT**	**QA-loss**
Ground-truth	Quantos satélites foram lançados desde 2007?	—	—	—	—	—	0.74
mT5-base	Quantos satélites foram lançados durante o período 2009-2011?	**31.56**	**63.94**	**53.73**	**92.86**	**66.11**	**1.02**
PTT5-base-base	Quantos satélites foram lançados durante a segunda geração?	**31.56**	**63.94**	**53.73**	91.09	52.51	0.25
PTT5-base-tokens	Quantos satélites foram lançados durante a segunda geração?	**31.56**	**63.94**	**53.73**	91.09	52.51	0.25
PTT5-base-qa-loss	Quantos satélites foram lançados durante a segunda geração?	**31.56**	**63.94**	**53.73**	91.09	52.51	0.25
PTT5-base-bertscore	Quantos satélites foram lançados durante a segunda geração?	**31.56**	**63.94**	**53.73**	91.09	52.51	0.25
PTT5-base-bleurt	Quantos satélites foram lançados durante a segunda geração?	**31.56**	**63.94**	**53.73**	91.09	52.51	0.25
Answer	2007	**BL4**	**RL**	**MTR**	**BERTS**	**BLRT**	**QA-loss**
Ground-truth	Quando foi lançado o satélite Compass-M1?	—	—	—	—	—	**76.04**
mT5-base	Quando o Compass-M1 foi lançado?	0.0	60.7	40.68	87.68	**79.36**	66.66
PTT5-base	Quando o Compass-M1 foi lançado?	0.0	60.7	40.68	87.68	**79.36**	66.66
PTT5-base-tokens	Quando o satélite experimental do Compass-M1 foi lançado?	0.0	**63.94**	**52.54**	**88.98**	74.18	52.42
PTT5-base-qa-loss	Quando o satélite experimental do Compass-M1 foi lançado?	0.0	**63.94**	**52.54**	**88.98**	74.18	52.42
PTT5-base-bertscore	Quando o satélite experimental do Compass-M1 foi lançado?	0.0	**63.94**	**52.54**	**88.98**	74.18	52.42
PTT5-base-bleurt	Quando o satélite experimental do Compass-M1 foi lançado?	0.0	**63.94**	**52.54**	**88.98**	74.18	52.42
Answer	cobertura regional funcional	**BL4**	**RL**	**MTR**	**BERTS**	**BLRT**	**QA-loss**
Ground-truth	O que foi alcançado com o lançamento de 9 satélites adicionais entre 2009 e 2011?	—	—	—	—	—	**83.95**
mT5-base	O que o Compass-M1 alcançou durante o período 2009-2011?	0.0	29.54	15.14	**72.54**	**31.66**	82.53
PTT5-base	Qual foi a cobertura do Compass-M1?	0.0	16.25	4.36	65.61	11.33	28.53
PTT5-base-tokens	O que o Compass-M1 alcançou?	0.0	**33.61**	**16.26**	66.11	17.17	82.21
PTT5-base-qa-loss	O que o satélite Compass-M1 alcançou?	0.0	24.37	8.73	67.06	22.06	80.07
PTT5-base-bertscore	Que tipo de cobertura o Compass-M1 alcançou?	0.0	23.58	9.64	62.04	18.48	32.31
PTT5-base-bleurt	Que tipo de cobertura o Compass-M1 alcançou?	0.0	23.58	9.64	62.04	18.48	32.31
Example 2							
Context	BBC Television é um serviço da British Broadcasting Corporation. A corporação, que opera no Reino Unido sob os termos de uma carta real desde 1927, produz programas de televisão por conta própria desde 1932, embora o início de seu serviço regular de transmissão de televisão seja datado de 2 de novembro de 1936.						
Answer	Reino Unido	**BL4**	**RL**	**MTR**	**BERTS**	**BLRT**	**QA-loss**
Ground-truth	Em que país a BBC está sediada?	—	—	—	—	—	98.97
mT5-base	Onde a BBC Television opera desde 1927?	0.0	37.5	22.62	73.89	42.24	75.29
PTT5-base	Em que país a BBC opera?	**55.78**	**79.05**	**65.78**	**91.25**	**67.64**	**99.14**
PTT5-base-tokens	Em que país a BBC opera?	**55.78**	**79.05**	**65.78**	**91.25**	**67.64**	**99.14**
PTT5-base-qa-loss	Em que país a BBC opera?	**55.78**	**79.05**	**65.78**	**91.25**	**67.64**	**99.14**
PTT5-base-bertscore	Em que país a BBC opera?	**55.78**	**79.05**	**65.78**	**91.25**	**67.64**	**99.14**
PTT5-base-bleurt	Em que país a BBC opera?	**55.78**	**79.05**	**65.78**	**91.25**	**67.64**	**99.14**
Answer	2 de novembro de 1936	**BL4**	**RL**	**MTR**	**BERTS**	**BLRT**	**QA-loss**
Ground-truth	Em que data a BBC transmitia regularmente pela TV?	—	—	—	—	—	98.65
mT5-base	Qual é o início do serviço regular de transmissão de televisão da BBC?	0.0	17.18	5.46	75.77	55.22	98.69
PTT5-base	Quando foi o início do serviço regular de transmissão de televisão?	0.0	9.24	2.86	75.28	55.2	**99.15**
PTT5-base-tokens	Quando a BBC começou a transmitir programas de televisão?	0.0	**30.0**	**18.62**	**78.14**	**68.49**	28.96
PTT5-base-qa-loss	Quando foi o início do serviço regular de transmissão de televisão da BBC?	0.0	17.18	5.46	76.97	63.78	98.15
PTT5-base-bertscore	Quando foi o início do serviço regular de transmissão de televisão da BBC?	0.0	17.18	5.46	76.97	63.78	98.15
PTT5-base-bleurt	Quando foi o início do serviço regular de transmissão de televisão da BBC?	0.0	17.18	5.46	76.97	63.78	98.15