



UNIVERSIDADE D  
COIMBRA

Renato Miguel Francisco de Matos

## **Automatic Generation of Multiple Choice Questions**

Dissertation in the context of the Master in Informatics Engineering,  
specialization in Intelligent Systems, advised by Prof. Hugo Oliveira and Hugo Amaro  
and presented to the Department of Informatics Engineering of the Faculty of  
Sciences and Technology of the University of Coimbra.

September of 2022





FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

DEPARTMENT OF INFORMATICS ENGINEERING

Renato Miguel Francisco de Matos

# Automatic Generation of Multiple Choice Questions

Dissertation in the context of the Master in Informatics Engineering,  
specialization in Intelligent Systems, advised by Prof. Hugo Oliveira and Hugo  
Amaro and presented to the Department of Informatics Engineering of the  
Faculty of Sciences and Technology of the University of Coimbra.

September, 2022





FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
**COIMBRA**

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

Renato Miguel Francisco de Matos

# Geração Automática de Perguntas de Escolha Múltipla

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Sistemas Inteligentes, orientada pelo Professor Doutor Hugo Oliveira e Hugo Amaro e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro, 2022



## Acknowledgements

I would like to thank Prof. Hugo Oliveira and Hugo Amaro for all the help during the development of the project and the writing of the thesis. Always available to clarify doubts and indicate the next steps to follow, their guidance was of vital importance in this work.

I am very thankful to my parents and my sister. They were always there, supporting and encouraging with all their efforts during all these years.

Last but not least, the friends. Not only to the ones that were part of this academic journey but also to those friends who, although on different paths, were always present and a source of motivation, thank you.





## Abstract

With technology taking a more prevalent role in our daily activities, new opportunities and challenges emerge. New technological tools have been successfully used in the educational context for some time now, facilitating teachers, educators and trainers in the transmission of knowledge. However, there are still tasks that can take advantage of these developments, as is the case of the creation of questions. The development of technology as a complementary tool to aid in question generation can decrease the effort related to this task and save valuable time, as well as potentially provide to those who are learning a new tool to learn new contents or revisit old contents.

In this work, we explore multiple Natural Language Processing techniques for the task of Automatic Generation of Multiple Choice Questions. Given that multiple choice questions are composed of more than one part, namely the stem (text of the question) and the distractors (incorrect answers), this involves multiple steps. Guided by a pipeline composed of Pre-processing, Answer Selection, Question Generation and Distractor Selection, we developed various approaches to generate the intended results. Some of the methods used are more conventional, involving linguistic analysis or rules to rearrange sentences, while others, such as the Transformers, are based on available models trained by other researchers for the task of Question Generation. We describe the background of the methods and how they were implemented in this work. To help in the development and in evaluation of the approaches implemented, we resorted to automatic and human evaluation metrics.

The resulting system was able to integrate various methods to perform each of the sub-steps we defined as necessary to generate multiple choice questions. Some of the approaches present positive results, standing as capable of creating questions of good quality and coverage that can be used as a starting point to create tests or questionnaires without the need for major human intervention.

## Keywords

Natural Language Processing, Multiple Choice Questions, Automatic Question Generation, Distractor Selection, Linguistic Analysis, Rule-based Approach, Transformers



## Resumo

Com a tecnologia a assumir um papel cada vez mais predominante nas nossas atividades diárias, surgem novas oportunidades e desafios. As novas ferramentas tecnológicas têm sido utilizadas com sucesso em contexto educacional há já algum tempo, sendo um auxiliar para professores, educadores e formadores na transmissão de conhecimento. No entanto, ainda existem tarefas que podem beneficiar de novos desenvolvimentos, como é o caso da criação de perguntas. O desenvolvimento de uma ferramenta complementar para auxiliar na geração de perguntas poderia diminuir o esforço relacionado a esta tarefa e economizar tempo valioso, além de potencialmente fornecer a quem aprende uma nova maneira de aprender novos conteúdos ou visitar conteúdos antigos.

Neste trabalho, exploramos várias técnicas de Processamento de Linguagem Natural para a tarefa de Geração Automática de Perguntas de Múltipla Escolha. Tendo em conta que perguntas de múltipla escolha são compostas por mais de uma parte, nomeadamente o texto da pergunta e as respostas incorretas, são necessárias várias etapas. Guiando-nos por uma pipeline composta por Pré-processamento, Seleção de Respostas, Geração de Perguntas e Seleção de Respostas Incorretas, desenvolvemos várias abordagens para gerar os resultados pretendidos. Alguns dos métodos utilizados são mais convencionais, envolvendo análise linguística ou regras para reorganizar frases, enquanto outros, como os Transformers, são baseados em modelos treinados e disponibilizados por outros pesquisadores para a tarefa de Geração de Perguntas. Descrevemos a base teórica dos métodos e como eles foram implementados neste trabalho. Para ajudar no desenvolvimento e na avaliação das abordagens implementadas, recorreremos a métricas de avaliação automática e baseadas em análise humana.

O sistema resultante foi capaz de integrar vários métodos para realizar cada uma das subtarefas que definimos como necessárias para gerar perguntas de múltipla escolha. Algumas das abordagens apresentam resultados positivos, sendo capazes de criar perguntas de boa qualidade e abrangência que podem ser usadas como ponto de partida para criar testes ou questionários sem a necessidade de grande intervenção humana.

## Palavras-Chave

Processamento de Linguagem Natural, Perguntas de Escolha Múltipla, Geração Automática de Perguntas, Seleção de Distratores, Análise Linguística, Abordagem Baseada em Regras, Transformers



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Contextualization . . . . .	3
1.3	Goals . . . . .	3
1.4	Proposed Approaches . . . . .	4
1.5	Results and Contributions . . . . .	5
1.6	Structure . . . . .	6
<b>2</b>	<b>Background</b>	<b>7</b>
2.1	Natural Language Processing, Understanding and Generation . . . . .	7
2.2	Question Answering and Question Generation . . . . .	9
2.3	Multiple Choice Questions . . . . .	9
2.4	Linguistic Analysis . . . . .	10
2.4.1	Named Entity Recognition . . . . .	11
2.4.2	Part-of-Speech Tagging . . . . .	11
2.4.3	Shallow Parsing (Chunking) . . . . .	12
2.4.4	Coreference Resolution . . . . .	13
2.4.5	Sentence Structure . . . . .	13
2.5	Artificial Neural Network Architectures . . . . .	15
2.5.1	Recurrent Neural Networks . . . . .	15
2.5.2	LSTM, GRU and Encoder-decoder Architecture . . . . .	16
2.5.3	Transformers . . . . .	17
2.6	Ontologies and Knowledge-Based Systems . . . . .	19
2.7	Word Embeddings . . . . .	20
2.8	TF-IDF . . . . .	21
2.9	Evaluation Metrics . . . . .	21
2.10	Summary . . . . .	24
<b>3</b>	<b>Related Work</b>	<b>25</b>
3.1	Pre-processing . . . . .	27
3.2	Question Generation . . . . .	28
3.2.1	Generation based on rules and templates . . . . .	28
3.2.2	Generation based on Machine Learning . . . . .	30
3.3	Distractor Selection . . . . .	31
3.4	Post-processing . . . . .	33
3.5	Evaluation . . . . .	33
3.6	Summary . . . . .	35
<b>4</b>	<b>A Pipeline for Question Generation</b>	<b>37</b>

4.1	Pre-processing . . . . .	38
4.2	Answer Selection . . . . .	40
4.3	Question Generation . . . . .	41
4.3.1	Rules . . . . .	42
4.3.2	Transformer . . . . .	46
4.4	Distractor Selection . . . . .	47
4.4.1	Named Entities . . . . .	47
4.4.2	GloVe . . . . .	48
4.4.3	WordNet . . . . .	49
4.4.4	DBPedia . . . . .	50
4.4.5	Transformer . . . . .	51
4.5	Summary . . . . .	51
<b>5</b>	<b>Evaluation</b>	<b>53</b>
5.1	Automatic Evaluation . . . . .	54
5.1.1	Automatic Evaluation of Answer Selection Methods . . . . .	54
5.1.2	Automatic Evaluation of Question Generation Methods . . . . .	58
5.2	Human Evaluation . . . . .	59
5.2.1	Human Evaluation of Question Generation Methods . . . . .	60
5.2.2	Human Evaluation of Distractor Selection Methods . . . . .	66
5.3	Main Conclusions . . . . .	68
<b>6</b>	<b>Conclusion</b>	<b>69</b>
	<b>Appendix A Question Generation Rules</b>	<b>77</b>
	<b>Appendix B Evaluation Forms</b>	<b>79</b>
B.1	Coimbra . . . . .	80
B.2	Cristiano Ronaldo . . . . .	82
B.3	Europe . . . . .	86
B.4	Queen . . . . .	88
B.5	Star Wars . . . . .	91
B.6	Distractors . . . . .	94







# List of Figures

2.1	NLP and NLG in the context of QG . . . . .	8
2.2	MCQ example . . . . .	10
2.3	NER example using SpaCy . . . . .	11
2.4	PoS Tagging example . . . . .	12
2.5	Chunking example . . . . .	13
2.6	Simple coreference example [Jurafsky and Martin, 2009] . . . . .	13
2.7	Example of a sentence analyzed via Claucy with with type SVC . . . . .	14
2.8	Example of a sentence analyzed via Claucy with two clauses . . . . .	14
2.9	RNN cell . . . . .	16
2.10	LSTM and GRU architectures . . . . .	17
2.11	Transformer architecture . . . . .	18
2.12	Example of crosses between pairs of mapped unigrams (METEOR)	23
3.1	Workflow suggested by Ch and Saha [2018] . . . . .	27
4.1	System workflow . . . . .	38
4.2	Co-reference resolution example in the Coimbra article from Wikipedia	40
4.3	Example of question generated from a clause of type SVA, with the answer being a named entity of label CARDINAL present in the subject, and a simple verb in present tense . . . . .	44
4.4	Example of question generated from a sentence of type SVA, with the answer being a named entity of label DATE present in the subject, and a verb "TO BE" in present tense . . . . .	45
4.5	Example of question generated from a clause of type SV, with the answer being a noun chunk present in an adverbial without NE label, and a compound verb in past tense . . . . .	45
4.6	Example of question generated from a clause of type SVC, with the answer being a noun chunk present in the subject with NE label ORG, and a verb "TO BE" in past tense . . . . .	46
5.1	Example from SQuAD: Passage (context) from a Wikipedia article, examples of questions about the passage, and potential answers identified in the text (and their location in the text) . . . . .	55
5.2	Answer distribution for the statement: "Questions are of sufficient quality to be included in a questionnaire without major editing required." . . . . .	62
5.3	Answer distribution for the statement: "The question set has good coverage in terms of possible questions to ask about the text." . . . . .	63

5.4	Answer distribution for the statement: "The questions presented are a good starting point for creating a questionnaire within this theme". . . . .	64
B.1	Questions to evaluate the performance of Question Generation approaches . . . . .	80
B.2	Information about the form for the article "Coimbra", including text from which questions were generated . . . . .	80
B.3	Questions generated using NER for Answer Selection and rules for Question Generation for the article "Coimbra" . . . . .	81
B.4	Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Coimbra" . . . . .	81
B.5	Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Coimbra" . . . . .	82
B.6	Questions generated using Transformers for both Answer Selection and Question Generation for the article "Coimbra" . . . . .	82
B.7	Information about the form for the article "Cristiano Ronaldo", including text from which questions were generated . . . . .	82
B.8	Questions generated using NER for Answer Selection and rules for Question Generation for the article "Cristiano Ronaldo" . . . . .	83
B.9	Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Cristiano Ronaldo" . . . . .	84
B.10	Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Cristiano Ronaldo" . . . . .	85
B.11	Questions generated using Transformers for both Answer Selection and Question Generation for the article "Cristiano Ronaldo" . . . . .	85
B.12	Information about the form for the article "Europe", including text from which questions were generated . . . . .	86
B.13	Questions generated using NER for Answer Selection and rules for Question Generation for the article "Europe" . . . . .	86
B.14	Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Europe" . . . . .	87
B.15	Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Europe" . . . . .	87
B.16	Questions generated using Transformers for both Answer Selection and Question Generation for the article "Europe" . . . . .	88
B.17	Information about the form for the article "Queen (band)", including text from which questions were generated . . . . .	88
B.18	Questions generated using NER for Answer Selection and rules for Question Generation for the article "Queen (band)" . . . . .	89
B.19	Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Queen (band)" . . . . .	90
B.20	Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Queen (band)" . . . . .	90
B.21	Questions generated using Transformers for both Answer Selection and Question Generation for the article "Queen (band)" . . . . .	91
B.22	Information about the form for the article "Star Wars (film)", including text from which questions were generated . . . . .	91

B.23	Questions generated using NER for Answer Selection and rules for Question Generation for the article "Star Wars (film)" . . . . .	92
B.24	Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Star Wars (film)" . . .	93
B.25	Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Star Wars (film)" . . . .	93
B.26	Questions generated using Transformers for both Answer Selection and Question Generation for the article "Star Wars (film)" . . . .	94
B.27	Information about the form to evaluate distractors . . . . .	94
B.28	Distractors selected for a question generated from the article "Coimbra". Answer NE label: GPE . . . . .	95
B.29	Distractors selected for a question generated from the article "Coimbra". Answer NE label: QUANTITY . . . . .	96
B.30	Distractors selected for a question generated from the article "Star Wars (film)". Answer NE label: PERSON . . . . .	97
B.31	Distractors selected for a question generated from the article "Star Wars (film)". Answer NE label: ORDINAL . . . . .	98
B.32	Distractors selected for a question generated from the article "Queen (band)". Answer NE label: WORK_OF_ART . . . . .	99
B.33	Distractors selected for a question generated from the article "Queen (band)". Answer NE label: DATE . . . . .	100
B.34	Distractors selected for a question generated from the article "Europe". Answer NE label: LOC . . . . .	101
B.35	Distractors selected for a question generated from the article "Europe". Answer NE label: ORG . . . . .	102
B.36	Distractors selected for a question generated from the article "Cristiano Ronaldo". Answer NE label: NORP . . . . .	103
B.37	Distractors selected for a question generated from the article "Cristiano Ronaldo". Answer NE label: CARDINAL . . . . .	103



# List of Tables

2.1	Question Example from Mindflow . . . . .	10
2.2	SpaCy named entity types . . . . .	11
2.3	SpaCy PoS tags . . . . .	12
3.1	Related Research . . . . .	35
5.1	Comparison of Answer Selection Methods . . . . .	56
5.2	Comparison of Answer Selection Methods (without stop words) . . . . .	57
5.3	Comparison of Answer Selection Methods (transformers) . . . . .	58
5.4	Comparison of Question Generation Methods . . . . .	59
5.5	Number of responses to forms . . . . .	61
5.6	Answer distribution for the statement: "Questions are of sufficient quality to be included in a questionnaire without major editing required." . . . . .	61
5.7	Answer distribution for the statement: "The question set has good coverage in terms of possible questions to ask about the text." . . . . .	62
5.8	Answer distribution for the statement: "The questions presented are a good starting point for creating a questionnaire within this theme". . . . .	62
5.9	Questions generated from the sentence "In 1949, the Council of Europe was founded with the idea of unifying Europe to achieve common goals and prevent future wars" using the approach "ne_transformer" . . . . .	65
5.10	Answer distribution for the statement: "Question well formulated." . . . . .	65
5.11	Answer distribution for the statement: "Question is pertinent." . . . . .	66
5.12	Answer distribution for the statement: "Answer is correct." . . . . .	66
5.13	Distractors' form results . . . . .	67



# Chapter 1

## Introduction

Technology permeates our lives and the contact with new information is constant, becoming an integral part of how we perform our daily activities. In this reality, new opportunities and challenges emerge.

One of the examples of how technology bridged our necessities is how we learn. It has been some time since we started using new tools in the educational context, such as presentations, videos and online platforms to expose content and test acquired knowledge in a more engaging and motivating way. Most of the time, we can say that these innovations were successfully incorporated by teachers, educators and trainers, helping in the transmission of knowledge.

However, there is still room to take advantage of existing opportunities. Teachers, educators and trainers have to assess if their students or trainees are understanding and retaining knowledge successfully. One of the most traditional ways to perform this task is through questionnaires. Although the existence of QLMS (quiz-based learning management systems), like Mindflow<sup>1</sup> and Kahoot<sup>2</sup>, ensures accessibility to the questionnaires in an interesting way, the task of creating the questions is still up to the educators. That may turn out to be a difficult task, especially regarding time constraints. The development of technology as a complementary tool to aid in the acceleration of the process can potentially save valuable time to dedicate to more specialized tasks that need personal interaction to help in the learning process and or that cannot be automatized. For example, having time to better consider the learning process and difficulties of students/trainees in planning the next classes, being available to clear doubts individually or simply reducing some workload.

Even from a student's point of view, having a tool capable of generating questions to study can be useful to learn new topics or revisit old content. It can also provide motivation and ensure that they learn at their rhythm inside and outside the classroom.

In the case of these scenarios, information is mostly produced by humans for humans, and thus available in human language. Approaching the problem with

---

<sup>1</sup><https://mindflow.pt/>

<sup>2</sup><https://kahoot.com/>

Natural Language Processing (hereafter, NLP) [Jurafsky and Martin, 2009] [Eisenstein, 2018] techniques seems a reasonable way to take advantage of this kind of information. Knowing that these are only two very concrete examples, an opportunity arises to combine the need to create questions with the techniques currently available in NLP. Using NLP concepts and techniques to develop solutions to this necessity is referred to as Automatic Question Generation (hereafter, AQG) [Kurdi et al., 2020]. The topic of this thesis is AQQ, specifically, the generation of multiple choice questions (hereafter, MCQ).

In the next sections of this chapter, we motivate and contextualize this work, describe its goals and the proposed approaches to achieve them, as well as share a summary of the results and contributions accomplished.

## 1.1 Motivation

Imagine, for example, a teacher who has had difficulties related to having less available time due to increased work in class preparation, having to teach at home or even to personal reasons. Having a tool capable of generating questionnaires from the programmatic contents would be a valuable resource. Even if some manual work was still required, it could mean less time spent on the creation of the questions and more available time to perform other tasks that could not be automated or accelerated, like personal interaction with the students for better knowing their difficulties. Or, in another case, a business like a call-center where the rotation of employees is usually high. In cases like this, there is a continuous flow of newcomers that need to be trained according to the formative content of the clients. Having a tool capable of accelerating the creation of questions from these contents could streamline the training process, helping in the training and evaluation of new workers.

In the current state of generalized use of technology, expanding the use of technology and new solutions seems to be beneficial to this type of contexts. Traditionally, the process of creating questions on any specific subject relies on manual and intellectual human work and is thus a time-consuming task. By exploring the potential of NLP, developing a system capable of generating questions with little human intervention can accelerate the process and improve productivity, potentially leading to more time available to devote to other activities. Students can also benefit from these new tools. A system capable of generating questions from educational sources like textbooks can be an effective way to complement their study. Having the possibility of creating questions about a certain school subject can also be used to simulate how exams for that subject can be and test how ready they are.

In the recent past, due to the pandemic, teaching methods had to be adapted and the use of technological solutions in different contexts was reinforced. This is one more example of how beneficial it can be to explore the adoption of current technology developments when creating systems capable of being an alternative when new challenges arise.



## 1.2 Contextualization

This thesis is developed under the Masters in Informatics Engineering, at the University of Coimbra. Its work is within the scope of the project SmartEDU, a partnership between the University of Coimbra, Instituto Pedro Nunes and Mindflow, which aims to investigate and develop a solution capable of automatizing the creation of educational and formative content, so that less time is spent on the manual creation of these contents. The system resulting from this work will be incorporated into SmartEDU and will be responsible for feeding a quiz generation system with MCQ about any given subject.

Mindflow is a company based in Coimbra, specialized in Mobile Learning and Gamification. By using together proven scientific principles of gamification, neuro-linguistic programming and positive psychology, it has the goal of identifying and improving learning and motivation processes. The company has been developing a product comprised of a mobile app and a web platform to transform traditional training content into engaging mobile games to support education and training. Such product can optimize training programs, turning the training of employees more cost efficient and increasing their productivity.

SmartEDU (CENTRO-01-0247-FEDER-072620) is co-financed by the European Regional Development Fund (FEDER), through Portugal 2020 (PT2020), and by the Regional Operational Programme Centro 2020.

## 1.3 Goals

With the objective of SmartEDU in mind – accelerating the process of creating tests and quizzes in general, on a given subject – the main goal of this thesis is to leverage new technologies, specifically under the domain of NLP, to study how they can be applied to generate adequate solutions. For this purpose, some of the existing approaches to AQG were studied to better identify those that better suit the purpose of generating MCQs, based on a set of documents on any given subject written in English. We are aware that generating well-written and meaningful questions is not always possible and there may have to be manual adjustments to the output. However, the goal is not necessarily to generate perfect ready-to-use questions, but questions that, even if containing some imperfections, may serve as a starting point for speeding up the process and making the work easier. For example, some adjustments regarding the fluency and grammar of the text of the question, or the adequacy of the generated distractors (alternative incorrect answers), may be necessary. However, by automatically generating a set of questions, these minor adjustments are expected to take less time compared to generating the questions entirely from scratch.

The main objective is then to develop a system comprised of the integration of computational tools for automatically generating MCQs from given written contents. As referred before it should be given the opportunity to edit the automatically generated content. To achieve this objective, there are some sub-steps

the system must be able to perform:

- **Answer Selection:** In this step, expressions (i.e. words or phrases) from the written contents or from external sources are selected as candidates to answers. This step is necessary at the beginning of the process due to the existence of approaches where it is necessary to indicate the answer for generating the question (see section 1.4);
- **Question Generation:** Given a context (e.g. a paragraph or a sentence) and the selected answer, the system must be able to generate questions about the written content.
- **Distractor Selection:** As incorrect answers are needed in multiple-choice questions, the system must be able to indicate some incorrect alternatives to the answer. These distractors can be selected from the written contents themselves or from external sources.

Evaluation of the implemented techniques is also a goal. To validate the approaches, automatic and human evaluations will be performed. To automatically evaluate, we can resort to commonly-used metrics like BLUE and ROUGE. To perform human evaluation we can resort to people related to the project to provide data that indicates the best techniques or approaches to achieve the main goal.

## 1.4 Proposed Approaches

In order to achieve the set goals, two paradigms were explored: one based on rules and another based on transformers. We decided to establish a single workflow common to both paradigms, with the difference being how the sentence is transformed from a statement into a question. This is due to the availability of transformer-based AQG models that we initially identified, which need the answer as input for generation. As the rule-based approach can also use this information to determine what questions should be generated, we concluded that we could have different implementations of the same pipeline. Later we identified answer-agnostic transformers, i.e., that do not need the answer. However, we kept the same workflow as it is compatible with all of the implemented techniques.

Generally speaking, we can divide the workflow into four phases: pre-processing, answer selection, question generation and distractor selection.

In pre-processing, co-reference resolution is applied to prepare the text for the following phases, replacing pronouns with the expressions they refer to. For selecting potential answers, we adopt more conventional approaches, like selecting named entities and noun chunks. We can also resort to a transformer-based model for this.

The major difference between the approaches resides in question generation, as we can opt for rules based on sentence transformations, assisted by the use of

the Claucy library<sup>3</sup>, or transformers. The latter are fine-tuned models (answer-aware or answer-agnostic) for the task of AQG in the English language. As transformers are considered a state-of-the-art architecture in NLP, we decided to use them. Moreover, for the task of AQG, they are commonly trained with the same language comprehension datasets we had available (e.g. SQUAD (Rajpurkar et al. [2016])). Because of that, we resorted to already available fine-tuned models.

To generate MCQs, it is also necessary to obtain distractors. To do so, we experimented with two types of method. In the first type of method, we looked for words or expressions present in the same text from where the answer was taken (named entities, noun chunks, n-grams, simple terms, ...). The other methods were based on external sources, like WordNet [Fellbaum, 1998], DBpedia [Auer et al., 2007], Word2vec [Mikolov et al., 2013a]. We also experimented with a transformer to perform this task.

To validate the approaches, automatic and human evaluations were performed. In automatic evaluation, common text similarity metrics, like BLEU and ROUGE, were used. Human evaluation was performed in two ways. In the first, the author of the thesis analyzed all of the questions resulting from the use of the methods better classified in automatic evaluation, generated from some Wikipedia sections. With this specialized opinion, we assessed whether the questions were well formulated, pertinent and had suitable answers. To have a broader opinion, questionnaires about the quality of the questions and of the distractors were also distributed and analyzed. While for the automatic evaluation the SQUAD (Rajpurkar et al. [2016]) dataset was used, for human evaluation we resorted to a small set of Wikipedia articles about subjects well known to the population in general.

## 1.5 Results and Contributions

With this work, we were able to develop a system based on a pipeline that integrates multiple types of approaches to select answers, generate questions and select distractors. According to human evaluation, some of the approaches present positive results, being able to generate questions with sufficient quality to be included in a questionnaire without major editing required, with good coverage of the source texts used and that can serve as a starting point in the creation of questionnaires.

Methods that resorted to transformers to generate questions always produced better results than the rule-based approach. When distinguishing these same methods based on how they select answers, the results were not that conclusive, with both expressions selected by transformers and named entities standing as good options.

However, there is still future work to be done. Some methods, for both question generation and distractor selection, need further development. Despite the

---

<sup>3</sup><https://spacy.io/universe/project/spacy-clausie>

results of distractor selection not being negative, we were not able to implement a method that would generate differentiated distractors, with some more incorrect than others. We also did not implement methods to deal with validation and ranking of the questions generated.

## 1.6 Structure

This thesis is structured as follows. In Chapter 2, topics and concepts needed to understand this work will be explained. After that, in Chapter 3, we will present related scientific work found during the state-of-the-art study that we consider to be the most relevant for our work and whose developments can be used as an inspiration to achieve our goals. Then, in Chapter 4, we detail the approaches and how we developed the system. In Chapter 5, we describe both automatic and human evaluations performed and analyze their results. Finally, in Chapter 6, we discuss the final considerations of this work.

# Chapter 2

## Background

In this chapter, theoretical concepts related to this work and methods found in the reviewed literature that are relevant to the project are presented. Firstly, we examine more generic concepts such as Natural Language Processing, progressing later to question-related themes like Question Generation. Then, we describe some methods commonly used in NLP, as well as methods used in rule-based and machine learning approaches. Potentially useful knowledge databases (e.g., WordNet) are also introduced. Lastly, some evaluation metrics are also explained.

### 2.1 Natural Language Processing, Understanding and Generation

As referred in Chapter 1, Natural Language Processing (NLP) can be defined as the study and application of computational techniques of how human language, in text or speech form, can be understood and generated to perform a certain task. NLP can be considered a discipline within Artificial Intelligence (AI), with influences from other areas such as Linguistics and Mathematics. Some of its applications are Machine Translation (automatic translation between languages), Summarization (identification of the most important ideas from a certain document), Dialogue Systems (also known as conversational agents, which converse with humans using natural language), Automatic Question Answering and, what interests us the most, Automatic Question Generation.

In simple terms, to process the input, NLP systems need to know how the language operates and what each word represents individually and in the context of a phrase/sentence. This type of processing can be described as Natural Language Understanding (NLU). Jurafsky and Martin [2009] identify the following properties as essential to, as they name it, obtain "knowledge of language":

- morphology: knowledge of the meaningful components of words;
- syntax: knowledge of the structural relationships between words;
- semantics: knowledge of meaning;

- pragmatics: knowledge of the relationship of meaning to the actual goals and intentions of the speaker;
- discourse: knowledge about linguistic units larger than a single utterance (e.g., documents).

Another important aspect is to be able to resolve ambiguity, that is, derive the meaning of a word or phrase when many are possible. Part-of-speech tagging, Word Sense Disambiguation and Probabilistic Parsing are some of the methods that can help in resolving ambiguities.

Analogously, we can describe Natural Language Generation (NLG) as the NLP area responsible for generating text. Gatt and Krahmer [2018] identify two types of NLG: text-to-text generation and data-to-text generation. What mainly distinguishes both of them is the input. While in text-to-text generation the input is exclusively text, in data-to-text there are other forms of input like tables, images or diagrams. Machine Translation, Summarization, Automatic Text Correction and Automatic Question Generation can be considered a text-to-text generation task. Examples of data-to-text generation are Image Captioning, reports (e.g. sports statistics, weather, clinical information) and other applications where useful table and graph information can be translated to text. Despite this division, a great part of the techniques are common to both. To generate text, a NLG system often needs to determine what information to include and in which order, select the right words or phrases to express the intended idea and combine all of them in a grammatically correct way. We will be mainly interested in text-to-text NLG.

AQG combines this two big NLP areas: NLU and NLG. As depicted in Figure 2.1 from Yao et al. [2012], to generate questions, NLU allows to get a representation of the data (text) and NLG is responsible for the generation of the questions from these data.

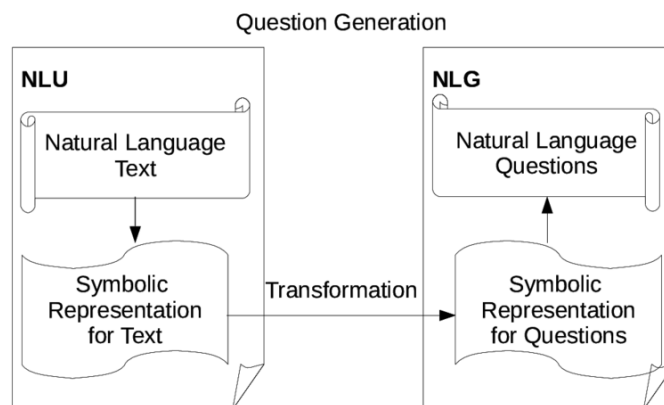


Figure 2.1: NLP and NLG in the context of QG

## 2.2 Question Answering and Question Generation

Question Answering is one of the tasks related to questions and NLP whose automation can benefit from the huge amounts of available information. Automatic Question Answering (AQA) can be described as the way to automate the production of natural language answers to natural language questions. To do so, an AQA system must be able to interpret the question and the documents from which it may be able to retrieve useful information to get the answers from. Very competent search engines, such as Google and Bing, have been around for some time now. Traditionally, these engines return a list of ranked whole documents (web pages) having as input a set of keywords. However, new approaches based on semantic search and knowledge bases have been implemented. Continuing with examples from Google, Google's Knowledge Graph is able to present a "summary" of relevant information to a given query and the search engine can also present commonly asked questions related to what the user is searching. In a way, these examples can be considered a form of QA. But, generally speaking, AQA systems allow users to ask a question in natural language and to get the answer also in natural language. To be capable of that, they might have to synthesize and summarize information from multiple sources and even to perform inference, that is to draw conclusions based on well-known facts. Then, they must be capable of returning that information in the form of a well-written text that correctly presents it.

Since 1999, the Text Retrieval Conference (TREC) has reported the advances in systems that perform this type of task.

However, our objective will be sort of the inverse – creating question-answer pairs given a certain document. Available literature regarding QG can be divided into two distinct methods, distinguishing how this task can be approached: more traditionally, using rules and templates, or by using neural networks (e.g. RNN or Transformers). During this chapter, we will present and detail methods and techniques used to generate questions that are currently used by the scientific community.

## 2.3 Multiple Choice Questions

The main focus of this thesis will be Multiple Choice Questions (MCQ). MCQs can be described as composed by three different parts. The stem is the question itself. Although the stem can be a declarative sentence, with the answer term missing (fill-in-the-blank), in this work we want to generate questions by transforming the original sentences (usually declarative) into interrogative sentences. The latter will be the type of question we will focus on more. Every question has one corresponding answer (or key), and, in the case of MCQs, a set of distractors. The answer is the correct option, derived from the question context, while the distractors are wrong options, present to mislead the answerer.

In Table 2.1 we are presented with one of the reference questions given to us

Table 2.1: Question Example from Mindflow

Question	Correct Answer	Wrong Answer 1	Wrong Answer 2	Wrong Answer 3
What is the most common benefit paid in UK?	plans or schemes	unemployment compensation	Health insurance	Meal allowance

What is the most common benefit paid in UK?

- a) plans or schemes
- b) unemployment compensation
- c) Health insurance
- d) Meal allowance

Figure 2.2: MCQ example

by Mindflow.

Typically, a MCQ has three components: a stem, a correct answer and multiple distractors (wrong answers). Based on the example of Table 2.1, we could present the question as in Figure 2.2. In that case, the components of the MCQ would be:

- a stem: "What is the most common benefit paid in UK?";
- a correct answer: in this case, a);
- several distractors: b), c) and d).

Other types of question that can be approached are: open questions, fill-in-the-blank (complete the sentence), yes/no, true/false.

About MCQs, Ch and Saha [2018] denote some of their advantages and disadvantages. As positive, and comparing with other types of question, MCQs allow to evaluate a considerable amount of knowledge in a short time. They also enable the use of a consistent scoring system, non-dependent of human opinion. Other positive aspect, with high interest to us, is the ease of automating the creation of tests based on this type of questions when compared with others. However, there are also some downsides. Even if answered randomly, there is always the possibility of choosing correctly. By opposition, a wrong answer does not consider partial knowledge that could be valued. It may be also difficult to create questions about more complex topics that need further explanation.

## 2.4 Linguistic Analysis

In this section, we explain some of the Linguistic Analysis tasks of importance to this work, namely Named Entity Recognition (NER), Part-of-Speech (PoS) Tagging, Shallow Parsing (Chunking) and Co-reference Resolution. Whether applying co-reference resolution to pre-process text, or resorting to NER, PoS Tagging or Chunking to identify meaningful expressions about which questions can be asked, among other applications, these tasks are highly relevant to our work.



Table 2.2: SpaCy named entity types

Named Entity Type	Description
CARDINAL	Numerals that do not fall under another type
DATE	Absolute or relative dates or periods
EVENT	Named hurricanes, battles, wars, sports events, etc.
FAC	Buildings, airports, highways, bridges, etc.
GPE	Countries, cities, states
LANGUAGE	Any named language
LAW	Named documents made into laws
LOC	Non-GPE locations, mountain ranges, bodies of water
MONEY	Monetary values, including unit
NORP	Nationalities or religious or political groups
ORDINAL	"first", "second", etc.
ORG	Companies, agencies, institutions, etc.
PERCENT	Percentage, including "%"
PERSON	People, including fictional
PRODUCT	Objects, vehicles, foods, etc. (not services)
QUANTITY	Measurements, as of weight or distance
TIME	Times smaller than a day
WORK_OF_ART	Titles of books, songs, etc.

### 2.4.1 Named Entity Recognition

Named Entity Recognition (NER) refers to the detection and classification of entities from a certain text. These entities can be of various types, like locations, dates or persons (see Table 2.2), and can be composed of single or multiple words. In a simplified way, everything that can be referred to with a proper name or temporal or numerical expressions can usually be associated with an entity type.

SpaCy<sup>1</sup> and NLTK<sup>2</sup> are examples of tools capable of performing NER. Using SpaCy and a pipeline trained for English (e.g., "en\_core\_web\_sm"), we can identify the entity types present in Table 2.2, as illustrated in the example of the Figure 2.3. In the figure, we can see that four named entities were identified: PERSON, GPE (in this case, country), WORK\_OF\_ART and DATE.

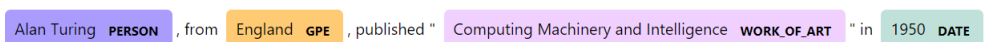


Figure 2.3: NER example using SpaCy

### 2.4.2 Part-of-Speech Tagging

Part-of-speech (PoS) can be understood as the class to which a word belongs considering its syntactic function. In English, some parts-of-speech are noun, pronoun, adjective and verb (see Table 2.3).

PoS tagging is used to assign grammatical information to each word of the sentence. In the case of Figure 2.4, the PoS tags of the sentence constituents were

<sup>1</sup><https://spacy.io/>

<sup>2</sup><https://www.nltk.org/>

Table 2.3: SpaCy PoS tags

PoS Tag	Description
Tag	Description
ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary
CONJ	conjunction
CCONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other
SPACE	space

identified using SpaCy<sup>3</sup> and the pipeline trained for English "en\_core\_web\_sm". The following PoS tags were obtained: PROPN (proper noun), PUNCT (punctuation), ADP (adposition), VERB (verb), CCONJ (coordinating conjunction) and NUM (numeral). In an approach in which we pretended to use some of these words as answer candidates, the ones identified as proper nouns (PROPN) or numerals (NUM) could be used as possible answers.

Alan Turing from England, published "Computing Machinery and Intelligence" in 1950.

PROPN PUNCT ADP VERB CCONJ NUM

Figure 2.4: PoS Tagging example

### 2.4.3 Shallow Parsing (Chunking)

Shallow parsing takes as input PoS tags and groups them in larger units – phrases. The following are examples of phrases: Noun phrase (NP), verb phrase (VP), adjective phrase (ADJP), adverb phrase (ADVP), prepositional phrase (PP) Shallow parsing can be used in entity detection.

In Figure 2.5 we used the same sentence of the PoS tagging example, in this case to identify noun phrase chunks (using spaCy too). We can observe that more meaningful chunks, like "Alan Turing" and "Computer Machinery" are identified, and that can be used to further proceed to entity type recognition.

<sup>3</sup><https://spacy.io/>

Alan Turing, from England, published "Computing Machinery and Intelligence" in 1950.

Figure 2.5: Chunking example

## 2.4.4 Coreference Resolution

Coreference resolution can be described as the task of finding all expressions that refer to the same entity in a text. In the context of our work, it can be useful for replacing ambiguous words (mentions), like pronouns, with the expression they refer to.

There are multiple types of references. For example, an anaphora occurs when there is a reference to an entity that has been previously introduced in the text. A cataphora is the inverse, with a word referring with another still will appear posteriorly.

In Figure 2.6 we have a simple coreference example. "he" and "Woodman" refer to "Tin Woodman" in the previous sentence, "it" refers to "Emerald City" and "Wizard" refers to "Wizard of Oz".

The Tin Woodman went to the Emerald City to see the Wizard of Oz and ask for a heart. After he asked for it, the Woodman waited for the Wizard's response.

Figure 2.6: Simple coreference example [Jurafsky and Martin, 2009]

## 2.4.5 Sentence Structure

Sentence Structure (or Pattern) allows the identification of the basic elements that compose a sentence. As described in Chapter 4, in this work we used the Claucy<sup>4</sup> library, an implementation of ClausIE [Del Corro and Gemulla, 2013] for Python and the Spacy<sup>5</sup> library. By analyzing the structure of a sentence, and resorting to the notations used in Claucy, the basic components that we can identify are:

- Subject (S);
- Verb (V);
- direct object (O);
- indirect object (also O);
- complement (C);
- adverbial(s) (A).

Based on these components, a sentence can be of one of the following types:

---

<sup>4</sup><https://github.com/mmxgn/spacy-clausie>

<sup>5</sup><https://spacy.io/>

- SV (subject + verb);
- SVA (subject + verb + adverbials);
- SVC (subject + verb + complement);
- SVCA (subject + verb + complement + adverbials);
- SVO (subject + verb + direct object);
- SVOA (subject + verb + direct object + adverbials);
- SVOC (subject + verb + direct object + complement);
- SVOCA (subject + verb + direct object + complement + adverbials);
- SVOO (subject + verb + direct object + indirect object);
- SVOOA (subject + verb + direct object + indirect object + adverbials).

In Figure 2.7 we can observe an example of the output achieved with Claucy for the sentence "Coimbra is a city and a municipality in Portugal.". Having into account that the output is presented in the format "<type, S, V, O, O, C, A>", we can say that the sentence is of type SVC, with subject "Coimbra", verb "is" and complement "a city and a municipality in Portugal".

"Coimbra is a city and a municipality in Portugal."  
 ↓  
 <SVC, Coimbra, is, None, None, a city and a municipality in Portugal, []>

Figure 2.7: Example of a sentence analyzed via Claucy with with type SVC

In the example of Figure 2.8 we observe a different case. As the sentence is constituted by two clauses, the analysis decomposes the sentences into clauses. A clause of type SVO is identified, with the other having type SV. To notice that in some cases, as in this SV clause, despite the presence of adverbials ("best" and "for his epic work Os Lusíadas"), the library might not reflect it in the type.

"Luís de Camões wrote a considerable amount of lyrical poetry and drama but is best remembered for his epic work Os Lusíadas"  
 ↓  
 <SVO, Luís de Camões, wrote, None, a considerable amount of lyrical poetry and drama, None, []>  
 <SV, Luís de Camões, remembered, None, None, None, [best, for his epic work Os Lusíadas]>

Figure 2.8: Example of a sentence analyzed via Claucy with two clauses

## 2.5 Artificial Neural Network Architectures

Machine Learning can be referred as the branch of Computer Science composed of a set of methodologies that try to develop computer programs capable of improving their performance with their own experience, that is, learning. For such, they can use as inspiration biological neural networks. In other words, the algorithms replicate the way humans learn by adapting concepts like neurons, dendrites and synapses to machine equivalents. Therefore, these models are named artificial neural networks (ANN), also commonly simply referred as neural networks.

We can divide Machine Learning in three main categories: supervised, unsupervised, or semi-supervised. In supervised learning, the data given to the network is labeled. While training the network with such data, it will learn what makes a certain group of inputs correspond to their label, and other groups to other labels. After training it will hopefully be capable of inferring which label corresponds to newly presented inputs. In opposition, unsupervised learning uses unlabelled data. Instead of mapping inputs to their labels, the inference is achieved by recognizing patterns in data, grouping the data instances based on their similarities. Semi-supervised uses a little of both worlds. For example, a model can be trained based on unsupervised learning and then fine-tuned to a specific task with supervised learning.

Related to ANN architectures, they can be split in shallow and deep neural networks. The architectures referred in this work – Recurrent Neural Networks and Transformers – belong to the deep neural network group.

### 2.5.1 Recurrent Neural Networks

The main feature that differentiates Recurrent Neural Networks (RNNs) from other types of Artificial Neural Networks (ANNs) is the implementation of loops. That is, this type of neural network allows the use of previous outputs as inputs. This aspect justifies the good results RNNs present in problems where the sequence of events or data points is relevant, such as a time-series. Text can also be seen as a sequence of characters or words. That is also the case for many NLP problems, such as AQG. For example, if we want to generate a sentence, it is important to predict the next word based on the previously generated words so that all of them together form a sentence that makes sense. Note that the RNN feedback can be from various steps behind, and not only one.

Analyzing the example from Figure 2.9, we can see that the cell takes  $X_t$  as input, outputs hidden state  $h_t$  and has (internal) state  $c_t$  at step  $t$ . Note that  $h_t$  and  $c_t$  are usually the same. The state is looped back and fed as input in the next step ( $c_{t-1}$ ), together with new input  $X_t$ . This loop mechanism allows this type of NNs to have sort of a memory that makes it possible to have in consideration information from previous steps.

There are also bidirectional variants of RNNs. Back to the example of text

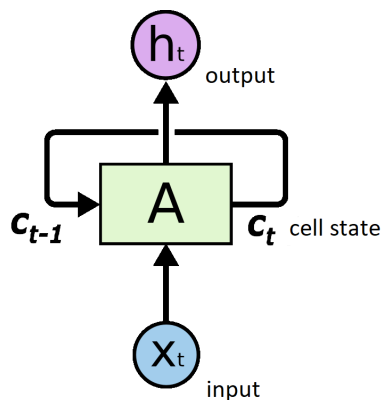


Figure 2.9: RNN cell

generation, in that case the prediction of a word would take into account the states that generate both previous and next words. These bidirectional variants are usually used in the works presented in Chapter 3.

We can differentiate this type of network even more according to the dimension of inputs and outputs: one-to-one, one-to-many, many-to-one, and many-to-many. Each one of them is usually more adequate for specific tasks. For example, to perform music generation one-to-many is used and to get named entity recognition it is more adequate many-to-many. For our task, where we want a sentence with multiple words to generate a question with multiple words, we have to use many-to-many.

## 2.5.2 LSTM, GRU and Encoder-decoder Architecture

Long-Short Term Memory (LSTM) is a type of RNN capable of learning long-term dependencies from sequences. When we talk about short-term dependencies, we are referring to when the gap between previous steps, whose states contain relevant information, and the current step, is small. In opposition, long-term dependencies occur when this gap is considerably larger. RNNs have good performance with short-term dependencies, but lack in long-term dependencies.

To have good performance with long-term dependencies, LSTMs are structured in a way that enables them to remember information needed in the context and forget what is no longer applicable. Briefly, each block contains three gates. An input gate that determines what information should be added or updated in the state, an output state that determines what part should be in the output, and a forget state, that determines what part of the information is no longer relevant. For each gate, the value is between zero and one, where one means "let all go through" and zero "nothing goes through".

GRU (Gated Recurrent Unit) is another type of RNN created with the objective of resolving the problem of long-term dependencies. In this case, there are two gates: reset and update.

An encoder is composed of a stack of RNNs layers (constituted by LSTM or

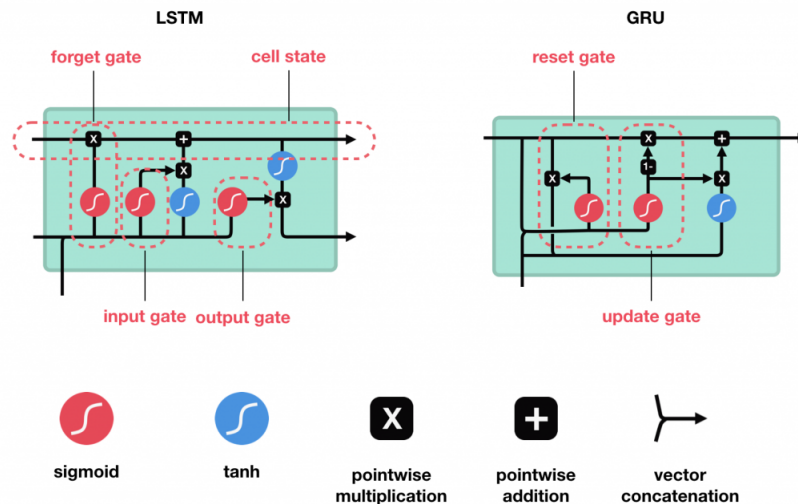


Figure 2.10: LSTM and GRU architectures

GRU cells). Encoding means converting input into a certain format. The encoder converts input into a vector called hidden state, which "summarizes" the input features and whose length depends on the number of cells in the RNN. The hidden state is the output of the encoder and the input of the decoder.

The decoder is also composed of RNN layers and converts the hidden state into an output sequence.

In Figure 2.10 we can observe both LSTM and GRU architectures, with respective gates (input, output and forget to LSTM and update and reset to GRU).

A major drawback of deep networks, like the ones we describe here, is the vanishing gradient problem. During training, weights are adjusted in back-propagation so that outputs are closer to the reference. In the case of deep networks, which have a great number of layers, it can become increasingly difficult to train, due to the values adjusting less than what was supposed. The vanishing gradient problem is then described as the probability of keeping the information diminishing exponentially the further away it gets from a specific word. The result is long-term information tending to be lost by the model, the longer the sequence is.

One of the great advantages of Transformers, the architecture presented in Section 2.5.3, is not having this problem.

### 2.5.3 Transformers

The Transformer [Vaswani et al., 2017] is a more recent artificial neural network architecture. It is heavily based on attention mechanisms and does not contain recurrence or convolutions like other types of artificial neural networks. One of the big advantages of the Transformer is the fact that, unlike architectures that consider data more sequentially, all operations can be parallelized. This makes the process of training big models more efficient.

Words can have different meanings when at the beginning or end of the sentence. One of the transformers' innovations is positional encoding. Instead of the words being sequentially given to the network, it starts by attributing a number to each word and processes the sequence as a whole. Knowing the position of the words allows the model not to be dependent on the order words are given to it. In the long run, this makes it possible to better learning of how word order affects the data without falling into the vanishing problem gradient.

Although the attention mechanism is not exclusive to transformers, it has great importance. By being capable of modeling dependencies without regard to distances in input or output sequences, as well as determining which words should be given more or less attention, it allows the model to look at every single word of a sequence and, because of that, is better at learning how gender, plurality and word order function in a certain language after tons of data.

However, self-attention (or intra-attention) is an innovation. After analyzing tons of data, it begins to learn the internal representation of the language, that is, the underlying meaning. That includes, for example, learning synonyms, rules of grammar, recognizing parts-of-speech and identifying gender or tense.

Describing the architecture in more detail, it is composed of an encoder and a decoder. Figure 2.11 represents the architecture, with the left part being the encoder and at the right the decoder.

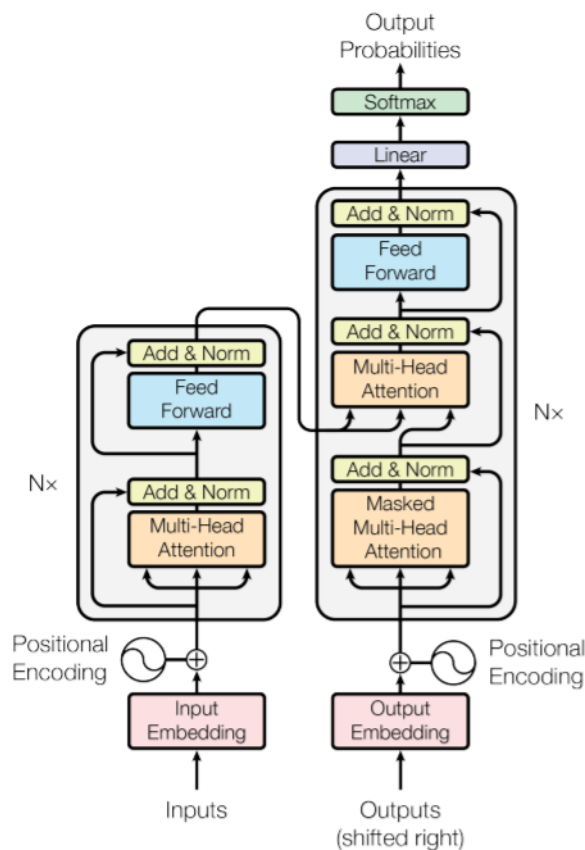


Figure 2.11: Transformer architecture

In the architecture, the first step is to obtain the embedding of the text, with



this being the input given to the encoder. The encoder is a stack of six identical layers, each consisting of two blocks. The first is a multi-head attention mechanism. Multi-head attention consists of multiple attention layers running in parallel, making it possible to attend to information from different representations at different positions, allowing to have more reliable results by learning different representations in a different manner. These results are later merged. The second block is a simpler feed-forward network. Both are followed by normalization.

Similar to the encoder, the decoder is also composed of six identical layers but, each of them has one extra block. The extra block performs masked multi-head attention. As the whole sequence can be given at a single time, if the embedding of the targets was not masked, the model could simply map the inputs to the outputs, harming the learning process. By modifying the block to perform masking, in conjunction with the output embeddings being offset by one position, it ensures that predictions for a certain time only depend on the target outputs from previous times.

There are pre-trained models that facilitate the use of this architecture and that can be fine-tuned for a specific task. BERT [Devlin et al., 2019], T5 [Raffel et al., 2020] and GPT-3 [Brown et al., 2020] are examples of models based on the Transformer architecture.

## 2.6 Ontologies and Knowledge-Based Systems

Ontologies define a set of common terms that describe and represent a domain (area of knowledge) and that can be understood by both humans and machines.

Knowledge-based systems use as source the knowledge of human experts on a certain domain. They resort to Knowledge Bases, whose creation usually has the involvement of experts of a particular domain, that represent a domain based on facts and rules in a way that a machine can use.

WordNet is a lexical database (or a semantically oriented dictionary) for the English language designed for NLP. The database is divided in nouns, adjectives, adverbs and verbs. Lemmata (plural of "lemma") are the base word forms and how these words are indexed in the database. Synsets group synonymous words, with the possibility of each word belonging to one or more synsets. Some of the contents of WordNet are:

- Definition of the word and phrases that use it in context;
- Synonyms and antonyms (same or opposite meaning words);
- Hypernyms and hyponyms (more abstract or more specific words);
- Semantic similarity between words.

In the context of our work, cohyponyms have special importance. We can describe a hypernym as being a broader, or more abstract, way to refer to a narrower, or more specific, concept (hyponym). For example, we can consider "dog"

as a hyponym and "animal" as its hypernym. Using this example, "cat" could be another hyponym for "animal". Cohyponyms are concepts that share the same hypernym. In this example, "cat" and "dog" are cohyponyms.

DBpedia[Lehmann et al., 2015] is a knowledge base that allows to represent information in a structured way. Because of being structured information, the data is machine-readable and allows to make queries. In the case of DBpedia, information is extracted from Wikipedia and converted into semantic triples. Using the SPARQL protocol and RDF documents (semantic triples of resource + property + value), we can query based on relations between facts. For example, given the name of a famous person, we can try to obtain their name or their nationality. Furthermore, with the Simple Knowledge Organization System (SKOS), we are able to organize these facts hierarchically.

Both these examples can be useful to this work, especially to generate distractors. By resorting to WordNet synsets or DBpedia concepts hierarchically organized, we can get concepts that, although different from the correct answer, are related to it.

## 2.7 Word Embeddings

A word embedding can be defined as the representation of a word from text in a numeric vector. Words with similar meanings or that are used in similar contexts tend to have similar word embeddings.

Word2vec [Mikolov et al., 2013b] is a tool commonly used in NLP to obtain word embeddings. It uses simple ANNs to calculate word embeddings based on the target words' context, with the contexts being the words that appear near the target word. In one of the implementations, the CBOW (Continuous Bag Of Words), uses contexts to predict a target word. For that, the contexts are given to the input layer and the target word to the output layer, with a hidden layer in the middle. While training, the values of this hidden layer change to predict the target word, with the final value being the embedding of that word. In the skip-gram implementation, it works the other way around, with the word being used to predict contexts. As similar words tend to have word embeddings with similar values, one of its applications is calculating the similarity between two words. For that, Word2vec uses cosine similarity.

$$\cos(x, y) = \frac{x \cdot y}{\|x\| * \|y\|} \quad (2.1)$$

The main characteristic of GloVe (Global Matrix Factorization and Local Context Window) [Pennington et al., 2014] is the use of a matrix of word-word co-occurrence count, that is, a matrix that shows how many times a word  $j$  appears in a context where word  $i$  also appears. By training with corpora of great dimensions, it is then possible to get the probability, by dividing the number of times  $j$  appeared in a context that also contains with the number of times any word appeared in the context of  $i$ .

With the way Word2vec works with contexts, it only considers what is called local dependencies. GloVe may be described as an extension to Word2vec, as beyond considering local dependencies, the use of the co-occurrence matrix works as a global context that allows the consideration of global statistics.

## 2.8 TF-IDF

The purpose of TF-IDF is to calculate the relevance of terms in a document. By taking into consideration the frequency of terms as well as the number of documents they occur on, it determines a term relevant to a document in case it has a high frequency in that document but is sparse in others. In the context of AQG, it can be used, for example, to identify expressions that, by being more relevant, are more suited to serve as the basis to question generation.

This metric is the combination of Term Frequency (TF) and Inverse Document Frequency (IDF) [Jurafsky and Martin, 2009]. TF is a very simple concept in which, as the name implies, the frequency of a term is calculated taking into account how many times it appears within the document. IDF can be defined by

$$idf_i = \log(N/n_i) \quad (2.2)$$

, where  $N$  is the total number of documents in the collection, and  $n_i$  is the number of documents where the term occurs. Terms that occur much, but only in a small portion of the documents, get a better value, and terms that appear in all documents are disregarded.

So, the final formula for TF-IDF, for a term  $i$  in a document  $j$  is the following:

$$tf-idf_{i,j} = tf_{i,j} * idf_i \quad (2.3)$$

## 2.9 Evaluation Metrics

The metrics described in this subsection are commonly used to automatically evaluate the results of tasks such as Machine Translation, Automatic Summarization and, in what is our case, Question Generation.

BLEU (BiLingual Evaluation Understudy) [Papineni et al., 2002] compares candidate and reference sentences by counting the  $n$ -grams both of them have in common and dividing this number by the total number of  $n$ -grams in the candidate sentence.  $N$ -grams can be understood as sequences of  $n$  words in a text. Unigrams correspond to single words, bigrams to sequences of two words, trigrams to sequences of three, and so on. This can be used to help models capture the meaning of words in context, as the previous and next words can be useful to learn the meaning of a word, or used for word prediction. As an example of the importance of the context, in one context the words "duck" or "book" can be

nouns, and in others, they can be verbs. Examples of n-grams are unigrams (comparison word by word, used in BLEU-1) and bigrams (sequences of two words, used in BLEU-2). The more n-grams in common, the better the candidate. The matches are position-independent. To prevent absurd candidates from being selected (like a sentence such as "The the the the" when the reference sentence contains "... the ..."), the method first calculates how many of each word exists in the reference and uses this number as the maximum number of occurrences that can be considered in the candidate. For example, if "the" occurs three times in the reference sentence, the maximum number of "the"s considered in the candidate sentence will be three. Papineni et al. [2002] mention that BLEU-1 tends to satisfy adequacy, but that longer n-gram matches, like in BLEU-4, account for fluency. The BLEU metric ranges from 0 to 1, with 1 meaning that the candidate is identical to the reference. BLEU also uses a brevity penalty, penalizing strings that are "too short".

ROUGE [Lin, 2004] presents a set of metrics to evaluate NLP tasks, such as summarization. ROUGE-N is similar to BLEU, counting the number of matching n-grams between candidate and reference sentences. While BLEU is considered a precision metric because of having in the denominator the number of n-grams in the candidate, ROUGE-N is considered a recall metric because of having the number of n-grams in the reference.

$$ROUGE-N = \frac{\text{number of matched } n\text{-grams}}{\text{number of } n\text{-grams in reference}} \quad (2.4)$$

ROUGE-L considers the longest common sub-sequence (LCS) between candidate and reference. A longer shared sequence should mean the two sequences are more similar. Here, the author proposes recall, precision and F-score.

$$ROUGE-L_R = \frac{LCS(\text{candidate}, \text{reference})}{\text{length}(\text{reference})} \quad (2.5)$$

$$ROUGE-L_P = \frac{LCS(\text{candidate}, \text{reference})}{\text{length}(\text{candidate})} \quad (2.6)$$

$$ROUGE-L_F = 2 * \frac{ROUGE-L_P * ROUGE-L_R}{ROUGE-L_P + ROUGE-L_R} \quad (2.7)$$

Another metric from the ROUGE set, ROUGE-S (Skip-Bigram Co-Occurrence Statistics) considers the skip-bigrams that occur in both candidate and reference. Skip-bigrams are sets of two words that appear one next to the other or separated by one or more words. For example, in the sentence "this is a simple example", the skip-bigrams are ("this is", "this a", "this simple", "this example", "is a", "is simple", "is example", "simple example"). As in ROUGE-L, the author suggests ROUGE-S recall, precision and F-score.

METEOR [Banerjee and Lavie, 2005] is composed by two phases. In a first phase, all the possible mappings between the unigrams from a generated and

a reference sentence are listed according to: being exactly the same, the same after stemming, or synonymous according to WordNet. This is the default order and each stage only maps unigrams that have not been mapped in any of the preceding stages.

In the second phase, the largest subset of these unigram mappings is selected. If there are multiple subsets with the largest number of mappings, the one with fewer crosses between pairs of mapped unigrams is selected. To understand the crosses, we can observe Figure 2.12, where a candidate and a reference string are one above the other, with lines connecting the matched pairs. When the line that connects a pair intersects the line of another pair, it is considered a cross. Having  $c$  as candidate string and  $r$  as reference string forming pairs  $(c_i, r_j)$  and  $(c_k, r_l)$ , if expression 2.8 has negative value, we can consider there is a cross.

$$(pos(c_i) - pos(c_k)) \times (pos(r_j) - pos(r_l)) \quad (2.8)$$

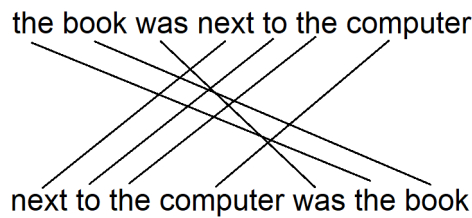


Figure 2.12: Example of crosses between pairs of mapped unigrams (METEOR)

For each pair of strings, the following can be calculated:

$$UnigramPrecision(P) = \frac{\#system\ unigrams\ mapped\ to\ reference}{\#reference\ unigrams} \quad (2.9)$$

$$UnigramRecall(R) = \frac{\#system\ unigrams\ mapped\ to\ reference}{\#system\ unigrams} \quad (2.10)$$

$$F_{mean} = \frac{10PR}{R + 9P} \quad (2.11)$$

A penalty is calculated to benefit longer matches. To do so, it groups the unigrams into the minimum possible number of chunks so that, in each chunk, all unigrams correspond to adjacent unigrams in the reference. For example, if the entire generated string matches the entire reference by order, there is only one chunk, and if there are no matches greater than one, there are as much chunks as unigrams.

$$Penalty = 0.5 \times \left( \frac{\#chunks}{\#matched\ unigrams} \right)^3 \quad (2.12)$$

After this, the final METEOR score is:

$$Score = F_{mean} \times (1 - Penalty) \quad (2.13)$$

## 2.10 Summary

In this chapter, we present concepts, methods and metrics useful to understand the work developed in this thesis and related literature. The majority of the methods presented here were used in the project, while others were presented due to being important to understand the task of Automatic Question Generation (AQG).

The areas of study related to this work, such as Natural Language Processing (NLP), Natural Language Understanding (NLU) and Natural Language Generation (NLG), were defined. The same for even more specific tasks, like Automatic Question Generation (AQG), and the type of questions we focused on, Multiple Choice Questions (MCQs).

We presented a set of Linguistic Analysis methods of great importance for our system. With Coreference Resolution we were able to understand how to process text, by finding and replacing expressions that refer to the same entity. Named Entity Recognition (NER), Part-of-Speech (PoS) Tagging and Shallow Parsing were also explained, with these methods allowing us to identify expressions and label them as belonging to a certain category, something essential in some of the steps of our system. A method to identify sentence and clause structures, by decomposing them into smaller elements, is also described. This was essential to one of our approaches, as we will present in the next chapter.

We dedicated a section for Artificial Neural Network Architectures, describing architectures such as LSTM, GRU, and the one heavily used in this work to generate questions, the Transformer.

The concept of word embedding, as well as implementations to calculate them, namely Word2vec and GloVe, were also covered. Another method covered, non-related to the previous, was TF-IDF. As in this project we make use of NLP techniques to analyze and transform text to achieve the proposed goals, using methods that represent words in numeric vectors and calculate the relevance of terms in a document, respectively, also present high relevancy.

At last, some evaluation metrics were presented, like BLEU and ROUGE, used in this work, but also METEOR, recurrently used in related literature for the task of AQG.

# Chapter 3

## Related Work

In this section we compile some works related to the task of Automatic Question Generation (AQG). Despite our main interest being multiple choice questions (MCQs), works about other types of question are also important as most of the techniques are common to generating questions in general. However, we also need to have into account that some techniques, like distractor selection, are characteristic of MCQs. Distractor selection or generation consists of the selection of wrong alternative answers to be used in MCQ.

[Kurdi et al., 2020] reports the systematic review performed by Alsubait et al. [2016]. This review characterizes AQG in seven dimensions: purpose of generating questions, domain, knowledge sources, generation method, question type, response format and evaluation. They report that there is a higher percentage of studies that deal with domain-specific approaches than with generic ones. However, as detailed by the authors, the domain of the majority of these studies is language learning. Two of the reasons the authors point to research being so much directed to language learning are the existence of many text resources that can be used for reading comprehension (RC) and the good performance of the available NLP tools (e.g., Part-of-Speech (PoS) tagging) in generating language questions. From the results present in [Alsubait et al., 2016], we can have a general idea of the methods commonly used to generate MCQs having text as the knowledge source, as that is our main concern. Usually the format of the questions is fill-in-the-blank or wh-question ("What...?", "Where...?", "Who...?", ...) and are generated via syntax or semantic methods. Despite the existence of text that can be used to generate distractors (e.g., based on word frequency, some type of similarity, PoS or other syntactic properties, ...), it is also possible to select them from external sources such as the lexical knowledge base WordNet [Fellbaum, 1998], thesaurus or textual corpora. These additional inputs are particularly important, given that the authors consider distractor generation the main challenge in MCQ. In these studies, the evaluation is commonly performed by either recurring to students (i.e., predicting test scores) or experts (i.e., evaluation of stems or distractors by experts, comparison of generated questions to questions created by experts). Comparing with other existing methods is also referred.

Based on previous classifications by Yao et al. [2012] and Alsubait [2015], Kurdi et al. [2020] suggest that the level of understanding can be syntactic or se-

mantic and transformation techniques can be based on templates, rules or statistical models. Regarding the methods used in each task, for each of the following AQG steps, the authors identified the following:

- Pre-processing: sentence simplification, sentence classification and content selection;
- Question Construction: stem and correct answer generation, distractor generation, feedback generation and difficulty control;
- Post-processing: verbalisation and question ranking.

The survey also categorizes existing standard datasets based on what they were developed for: machine reading comprehension, Automatic Question Answering (AQA) systems training or AQG.

Ch and Saha [2018] review existing approaches and propose a general workflow to generate MCQ. Their proposed workflow consists of six phases and details the techniques that can be used in each of them (see Figure 3.1). The authors identify some flaws in current techniques that could be tackled in future works:

- In pre-processing, the focus should be on processing hybrid educational text (containing figures, tables, bullet points, mathematical expressions, ...);
- In sentence selection, systems should be able to generate questions from multi-line facts;
- In distractor selection, deep semantic text analysis and neural embedding based methods may be the best options for further research.

They also find that the recent trend for “key selection” are methods based on semantic information and machine learning. Another conclusion from the authors is that there is no standardized evaluation metrics or benchmark data, making it more difficult to compare different systems. To solve that, creating standard evaluation techniques and gold-standard data are needed improvements. Parameters to evaluate stems and distractors are also suggested based on existing works. Kurdi et al. [2020] have a similar conclusion concerning evaluation, referring the need to harmonise evaluation metrics and investigate more feasible evaluation methods. Other findings are the need to automate template construction and the enrichment of question forms and structures. The limited research on the control of questions difficulty and the lack of informative feedback are issues also identified by Kurdi et al. [2020].

After reviewing existent literature about AQG, a workflow to be followed was established. Considering a version of this workflow, consisting of Pre-processing, Question Generation, Distractor Selection and Post-processing, we present some existing works. In the end, the task of Evaluation is also discussed.



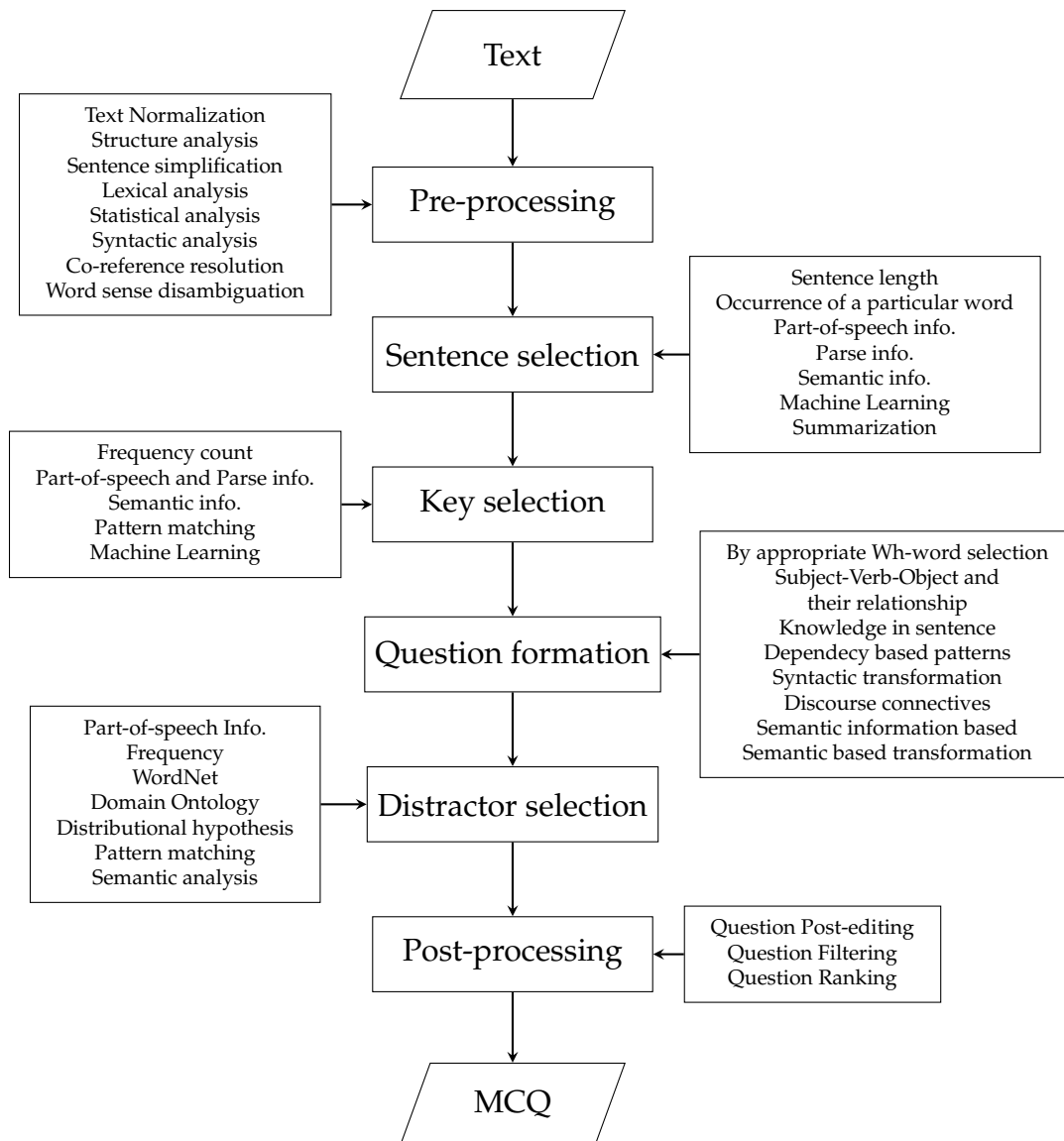


Figure 3.1: Workflow suggested by Ch and Saha [2018]

### 3.1 Pre-processing

Some methods are widely used in pre-processing, being common to most works. That is the case of tokenization, sentence selection and removal of unwanted parts of the texts. For example, in [Hussein et al., 2014], sentence detection is performed based on punctuation marks. Du et al. [2017], while training in SQuAD<sup>1</sup>, use a framework to tokenize and split the sentences. However, they also detect the location of the answer in the sentence and, if an answer spans in more than one sentence, these sentences are concatenated and given as input as they were a single one. SQuAD is a dataset commonly used for the tasks of Automatic Question Answering and Automatic Question Generation composed of article passages from Wikipedia, where for each article it is presented a set of question-answer pairs with the location of the answers in the passage. Resorting to a syntactic parser,

<sup>1</sup><https://rajpurkar.github.io/SQuAD-explorer/>

Ali et al. [2010] divide complex sentences into elementary ones. To do so, a pre-selection of relevant sentences based on locating target answers in the text is first performed, followed by the representation of the complex sentences in a syntactic tree. The newly constructed sentences are generated based on the combination of the sentence's phrases.

The use of tags, like PoS, or the identification of named entity types, using NER, is also common, especially to detect relevant information that can be used to produce questions. In the case of [Ali et al., 2010], NER and PoS tagging are used to select and then classify each sentence based on the possible factoid question types to be asked with the information present in that sentence ("What...?", "Where...?", "When...?", "Who...?", "How many/How much...?"). In this same work, PoS is used to extract information about the verbs and their tenses. Hussein et al. [2014] also determine possible question types based on PoS tags, NER, or even custom remarks and verbs that are identified to later help in tense transformation. [Zhou et al., 2017] also resorts to PoS tagging and NER, and normalizes the text by lower-casing it, something that is also done by Du et al. [2017].

## 3.2 Question Generation

In this subsection, we present question generation approaches, namely based on rules or templates, or based on machine learning techniques, like neural networks (including RNNs and Transformers). Answer selection is also referred when it is an essential part of the question generation process.

### 3.2.1 Generation based on rules and templates

The more traditional approaches to the task of AQG are based on templates or rules. These templates or rules are usually created manually and describe how sentences/paragraphs with a certain structure should be transformed into questions.

Some works analyze syntactic functions and dependencies. The rule-based approach by Mitkov et al. [2006] can be divided in two steps: term extraction and stem generation. In term extraction, nouns and noun phrases that represent domain-specific ideas are identified as possible target answers using shallow parsing. Nouns are considered target answers if their frequency is over a certain threshold that depends on parameters such as length of the text and the number of nouns it contains. Noun phrases need to contain one of these nouns and satisfy a regular expression. TF-IDF is experimented but with not so good results. Sentences with at least one term and SVO (subject-verb-object) or SV (subject-verb) structure are transformed into questions. WordNet [Fellbaum, 1998] can be accessed for replacing the target answers by their hypernym (i.e., a superordinate concept). As an example, in the sentence "*Syntax studies the way words are put together into sentences.*", the word "syntax" could be the domain-specific target answer, "discipline" its hypernym, and the following question could be generated:

*"Which discipline studies the way words are put together into sentences?"*.

A rule based model is also developed by Hussein et al. [2014]. In the training phase, after pre-processing, a sentence is selected. A statistical parser and the remarks obtained in pre-processing (i.e., PoS tagging, NER or custom remarks to help determine type of question, verb identification to help in tense transformation) are used to match the sentence with existing templates. If not found, a new template is created based on the sentence and input given by the user. Optionally, the verb and its different forms can be stored in a database. The result is a large amount of stored template rules after training. To generate questions, the process is similar: select a sentence, extract sections, and apply transformation rules or patterns. Two types of question can be created: wh-questions ("Who...?", "When...?", "Where...?", "What...?", "How much...?"), also referred as factoid questions in other works, or complete questions (based on named entity types and with up to two complete phrases). In the example given in paper, in training phase the sentence *"I found my books on the table"* is not matched with any template and is parsed to obtain the following PoS tags: "PRP VBD PRP\$ NNS IN DT NN". With the input of the user informing what wh-question mark should be used, the verb tenses and the template rule as a series of PoS tags, a new template is created. Specifically for this sentence, the question obtained was "Where did you Not found your books?". It will be possible to generate questions for other sentences that also match this pattern using this same template.

Ali et al. [2010] use syntactic information to select sentences and transform them. One of the system modules is responsible for, from complex sentences, extracting elementary sentences using syntactic information. This is done by constructing syntactic trees with these sentences and grouping some of the sentence's phrases, generating simpler sentences. Then, after classifying the sentences based on the information obtained by PoS tagging and NER (i.e., classes like Human, Entity, Location, Time and Count), factoid questions are generated using a predefined set of interaction rules that consider the relation between class occurrences and verb (e.g. "Human Verb Human Time", "Human Verb Entity"). The example given is a sentence with structure "Human Verb Human" being classified as a sentence from which questions of type "Who" and "Whom" can be generated from and, if followed by a preposition that represents time, also question type "When".

Other works resort to semantic-based methods, ensuring some knowledge of the meaning of the constituents of the sentences. By using Semantic Role Labeling (SRL), as in Flor and Riordan [2018], arguments (e.g., subjects and their type), modifier arguments, and verbal groups are identified, and then used to decide how to modify the sentences.

To some extent, the techniques described above for English can be adapted for Portuguese. Ferreira et al. [2020] show two approaches, one based on syntactic information and another based on SRL. In the first approach, sentences are split in syntactic chunks and checked for named entities. Depending on the information obtained, different transformation rules are applied. In the second, the applied transformations depend on the arguments and verbs are identified by performing SRL.

An advantage of approaches based on rules or templates is that the created models are not “black boxes”, as is the case of neural networks. This means that it is easier to understand and fix the results, which are also more predictable. Moreover, these approaches generally do not require training data. A major disadvantage is the amount of manual labour for producing the rules. It is virtually impossible to cover all possible situations, but, as long as there is data, a supervised learning approach is generally a faster way to adapting to the data, often also achieving a better performance.

### 3.2.2 Generation based on Machine Learning

In this context, recurrent neural networks (RNNs) are usually used in an encoder-decoder architecture. It is also common for them to include an attention mechanism, which allows the decoder to focus more on some part of the data.

The encoder-decoder architecture is a type of neural network widely used in NLP tasks to generate questions without pre-defined rules. Zhou et al. [2017] implement a model whose encoder considers text features (e.g., answer position, word case, PoS and NER tags) to produce a better encoding and generate answer-focused questions. The model is based on bidirectional Gated Recurrent Unit (bi-GRU) cells to read the input in both forward and backward orders. The decoder is based on GRU and has an attention mechanism, for generating answer-specific questions, and a copy mechanism, which calculates the probability of each word being rare or unknown, in order to copy them, as the model tends not to consider such words, thus ending being lost in the process.

Also implementing an encoder-decoder architecture, in this case based on LSTM with attention, Du et al. [2017] developed two variations: one that only considers information from single sentences and other that considers both sentence and paragraph information. Regarding the encoder, both variations use attention-based bidirectional LSTM encoders for the sentences, with only one using the paragraph encoder. In the simpler model, the decoder is initialized with the representation obtained from the sentence encoder, while in the more complex the sentence and paragraph encoders’ output is concatenated so that the decoder can be initialized with it.

Transformers [Vaswani et al., 2017] are a more recent architecture that is also composed by a multilayered encoder-decoder structure and relies more on the attention mechanism. An important advantage of Transformers over RNNs is the simpler architecture and the lower training time. BERT is one of such Transformers, which has been used to automatize question generation in works by Kriangchaivech and Wangperawong [2019] and Chan and Fan [2019]. The later started by a straightforward use of BERT, adapting the model to the task without considering decoded information from previous steps. After that, they try to improve the results by considering sequential information and by trying to resolve ambiguity related to the answer in the input text.

[Brown et al., 2005] explains how the automatic question generation is performed for REAP, a system that provides questions to assess the learning of

new vocabulary based on the student's reading level. Six types of questions are generated (definition, synonym, antonym, hypernym, hyponym and cloze questions), presented as word-bank or multiple choice questions. Word-bank questions present a list of words and a list of sentences or phrases with a blank space that must be completed. Multiple-choice are our typical MCQs, composed by a stem (not transformed, consisting of a sentence/phrase with a blank space) and a set of possible answers in which only one is correct. Resorts to WordNet to obtain the target word sense based on its PoS tag and on the sense more commonly attributed to the word. In order to adapt the REAP system [Brown et al., 2005] to the Portuguese language, Correia et al. [2012] created REAP.PT (READER-specific Practice PorTuguese), a tutoring system aimed at learners of Portuguese as a second language. However, the approach used is very different, being more Machine Learning oriented. Using newspaper articles in Portuguese, REAP.PT highlights the words also present in the Portuguese Academic Word List (P-AWL) and generates multiple fill-in-blank stems for each of them. Texts are divided into indexed sentences. To train the model, a gold standard consisting of sentences manually classified as positive or negative examples is created. At the same time, features (e.g., sentence length, target word position, proper names, co-occurrences) are extracted from the sentences. Both gold standard and features are then used to train the classifier based on a SVM.

[Wang et al., 2020] created a model constituted by a compromise between answer-aware and answer-agnostic question generation. Both answer-aware and answer-agnostic question generation have disadvantages. In answer-aware models, the generated questions can be more trivial, being easier to infer the answer by the subsequent text and not so relevant to be asked. In answer-agnostic models, by removing the bias to the selected answers, there is the possibility of questions not being answerable by the source text, despite being well-formulated. Combining both types of models, the proposed model is built upon the answer-agnostic approach, but with hidden answers being inferred and generated questions based on the induced hidden answer. In this process, the hidden answers are generated given the source text, and then both text and answers are combined to produce the questions so that the probability of the answers being answerable is higher.

According to the reviewed literature, the application of approaches based on neural networks, especially in recent years, resulted in increasing developments. More specifically, RNNs and Transformers have been successfully applied to the task of AQG. The Transformer is a more recent architecture with state-of-the-art performance in NLP. AQG is not an exception to the application of this new architecture.

### **3.3 Distractor Selection**

Distractor generation can be categorized according to the source of the distractors: with origin from the same text or from external sources. WordNet [Fellbaum, 1998] is an external resource commonly used for this purpose. Mitkov

et al. [2006] rely to WordNet to select distractors as semantically close to the correct answer as possible (e.g. "semantics" and "pragmatics" should be preferable to "chemistry" and "mathematics" as distractors to "syntax"). For that, words that share the same hypernym or even the hypernym itself of the target answer are retrieved from WordNet and, in case of being too many, it is given preference to those that also appear in the original text.

In the work of Zhang et al. [2020], the method used to generate distractors was a combination of multiple approaches that result in three types of distractors. Type-1 distractors are generated while dealing with numbers or target words that can be converted to numbers, detected using regular expressions based on PoS. After converted to a number, the value can be increased or decreased by some units, changed within a range around the value or randomly. The example given is of "Friday", that can be convert to the value "5", changed to "4" and converted back to a word as "Thursday". Type-2 distractors are generated when a named entity is detected (e.g., person, location or organization). In those cases, other NEs of the same type can be searched in the text from which the question is being generated, alternatively recurring to external sources like knowledge bases. Type-3 distractors are hypernyms selected from WordNet that have a similarity value to the the target word within a certain interval in order to not be too similar or too different ([0.6,0.85] is the example given). Distractors that contain the target word (e.g., "news" and "breaking news") are removed, as well as in the case of having the same prefix and low edit distance to prevent misspelled versions of the same word. These distractors are then ranked based on Word2vec [Mikolov et al., 2013c] cosine similarity, WordNet WUP score [Wu and Palmer, 1994] and edit distance score. For targets with multiple words, each word is selected according to a fixed preference based on SRL and replaced based on a fixed preference of type of distractor.

Stasaski and Hearst [2017] experimented generating distractors using ontology structure and relationships, as well as using embeddings to calculate the similarity between nodes and answer or question components.

In their work regarding REAP.PT, Correia et al. [2010] also mention multiple ways to generate distractors. They experimented manually selecting distractors for a random small set of stems, choosing as distractors words that were, in relation with the correct answer, quasi-synonymous or quasi-antonymous, false-friends, pseudo-prefix or pseudo-suffix variations or words with similar spelling or sound. They also generate distractors based on similar features (e.g., PoS tag, frequency, according to a distance calculus); by exploring common errors in Portuguese (e.g., by modifying the answer with a table of common mistakes to obtain misspelled word); or by filtering using lexical resources (similar to WordNet, based on synonym sets and the relation between synonyms, hyponyms and hyperonyms). A given example is, to "condução", selecting as distractors "direção" (both nouns derived from synonym verbs, "dirigir" and "conduzir", but with different meanings), "condição" (phonetically and graphemically similar) and "redução" (unrelated but with the same prefix). As the objective is to generate distractors for a language learning context, another method is misspelling letters commonly mistaken, like replacing "ss" with "ç" or "j" with "g".

## 3.4 Post-processing

Mitkov et al. [2006] developed a post-editing environment in which one can edit questions and replace distractors given a list of alternative ones. The post-editor discards questions if they need too much revision or are not about central concepts, or considers them "worthy" if they are ready to be used or only need some post-editing. Needed post-editing is differentiated between "minor" (e.g., insertion of an article, correction of spelling and punctuation), "fair" (e.g., re-ordering, insertion or deletion of several words, replacement of one distractor at most) or "major" (substantial rephrasing of the stem and replacement of at least two distractors).

The system by Hussein et al. [2014] allows to rank, review and store questions in a database to later be used to create exams. After AQG, a list of possible questions is reported. The user can edit or remove the question and classify its difficulty. Then, the question can be stored, being attributed to a specific course and chapter created previously. By doing this, a bank of questions for a specific chapter of a determined course can be created. The saved questions can later be used to automatically generate exams with different levels of difficulty.

## 3.5 Evaluation

There are no standardized methods to evaluate AQG systems. However, we can identify two main categories in the scientific literature: manual evaluation and automatic evaluation.

In manual evaluation, the assessment is usually performed by people who are knowledgeable of linguistics, of the questions topic, or that are representative of the group the questions are destined for. For example, in Flor and Riordan [2018], besides the comparison of two systems, two experts evaluate how well-formed the grammar of the questions is, how semantically adequate the question is in relation to the original sentence, and how relevant the information present in the question is, by comparison with the information present in the original sentence.

Applied to the context of learning, in Stasaski and Hearst [2017]'s work the quality of questions and distractors is evaluated by teachers. To evaluate the quality of the questions, in [Mitkov et al., 2006], students were asked to answer them after being post-edited and approved by a lecturer that certified that they addressed taught materials. Considering two groups, one consisting of the top and the other the bottom scores, three variables were considered: question difficulty (i.e., ratio of students that answered the question correctly); discrimination power (i.e., comparison of the number of students in each group who answered the question correctly); and distractor usefulness (i.e., comparison of the number of students in each group that selected each distractor, testing the premise that a good distractor attracts more students from the lower scoring group than from the upper).

In Zhou et al. [2017], a work that uses the encoder-decoder model, various versions of the model with different implementations are compared, as well as a modified rule based system. The human evaluation considers the following scale: "good", if the question is meaningful and matches the sentence and answer very well; "borderline", if it matches the sentence and answer more or less; or "bad", if it does not match or does not make sense. They also calculate precision and recall to evaluate each question type ("What...?", "How...?", "Who...?", "When...?", "Which...?", "Where...?", "Why...?" or Other). Du et al. [2017] relies on human evaluation for measuring the quality of the questions regarding naturalness (i.e., grammatically and fluency) and difficulty (i.e., sentence-question syntactic divergence and the reasoning needed to answer the question). To perform human evaluation, a random sample of sentence-question pairs was given to four professional English speakers that rate the variables on a 1 to 5 scale.

Human evaluation is performed in [Wang et al., 2020] by resorting to volunteers, where one hundred random sampled context-question pairs are used to evaluate the quality of the model. The volunteers are asked to evaluate the questions considering their Fluency (questions well-posed and natural in terms of grammar and semantics), whether they are Answerable (questions can be answered by the context paragraph) and their Significance (questions focus on the significant parts of the source text). Significance can be evaluated by considering whether the question is simply a syntactical transformation or whether the corresponding answers are trivial.

Automatic evaluation methods simplify and streamline the evaluation process, as we can quite easily calculate a value without great human involvement. This does not only enable to quantify the performance of the tested system, but allows for a quick comparison with other systems for which performance scores were computed with the same metrics. However, compared to human evaluation methods, they also have limitations. Human evaluation, being performed by experts or potential users of the system, can have context specificities in account. Furthermore, automatic evaluation is based on the comparison between candidates and references. If a word from a candidate has the same meaning as the correspondent from the reference but is a different word, it will simply be considered different, affecting the score. Moreover, an adequate reference dataset is necessary to perform the evaluation, something that is not always readily available.

BLEU [Papineni et al., 2002] is a very common metric with some variations and is used in many of the reviewed works. For example, in automatic evaluation, Zhou et al. [2017] use BLEU-4 variation and Du et al. [2017] use BLEU-1, BLEU-2, BLEU-3 and BLEU-4 variations. In addition to BLEU, Du et al. [2017] also use METEOR [Banerjee and Lavie, 2005] and one of the metrics from ROUGE [Lin, 2004], ROUGE-L. [Wang et al., 2020] compare their answer-aware and answer-agnostic joint model approach with multiple answer-aware and answer-agnostic models, according to BLEU (1,2,3 and 4), Meteor and Rouge-L. These metrics are used to compare the generated questions with reference questions created by humans.



### 3.6 Summary

In this section we present a summary of the main works compiled in this chapter. Each of them has described its reference, some brief highlights, dataset and evaluation methods used and what language did they focus on. These works compose a representative list of the variety of approaches available to the task of AQG with text as the knowledge source, from rule or template based to machine learning. Some are about AQG in general while others are more focused on the generation of MCQs, some of those specifically on distractor selection methods. Regarding the dataset, SQuAD is effectively the most commonly used. There is good variety of human and automatic evaluation methods. English and Portuguese are the languages that appear in this summary as they were the languages on which this research focused on.

Table 3.1: Related Research

Reference	Highlights	Dataset	Evaluation	Language
Mitkov et al. [2006]	Rule-based. Shallow parsing. SVO or SV sentences. WordNet to select distractors.	-	Human	English
Zhou et al. [2017]	Feature-rich encoder and attention-based decoder. Copy Mechanism to deal with rare/unknown words	SQuAD	BLEU, Human	English
Hussein et al. [2014]	Wh-questions or about known entities. Sentences matched with templates based on syntactic information. While training, a new template can be created by the user if not matched. Automatic set of questions the user can edit.	-	-	English
Flor and Rior-dan [2018]	Generation of yes/no and wh-questions using SRL.	SQuAD	Comparison with NN system. Human	English
Stasaski and Hearst [2017]	Distractor generation using ontology relationships and structure or by using embeddings to calculate the similarity between nodes and answer or question components.	-	Human	English

Continued on next page

Table 3.1 – continued from previous page

Reference	Highlights	Dataset	Evaluation	Language
Ali et al. [2010]	Wh-questions. Rules. Syntactic tree to extract elementary sentences. After PoS and NER tagging, classifiers determine the question type	TREC-2007	-	English
Correia et al. [2012]	Portuguese version of REAP. Fill-in-the-blank (cloze) questions. SVM classifier, trained with features extracted from the sentences and a Gold Standard.	News corpus, P-AWL	-	Portuguese
Correia et al. [2010]	Multiple ways to generate distractors	-	-	Portuguese
Du et al. [2017]	RNN encoder-decoder architecture with attention mechanism. Compares sentence-level and paragraph-level encoding.	SQuAD	Human, BLEU, METEOR, ROUGE	English
Ferreira et al. [2020]	One approach based on rules that use NER and syntactic information. Other based on SRL.	Multieight-04	BLEU, ROUGE-L	Portuguese
Chan and Fan [2019]	Three different NN based on BERT: a straightforward, considering sequential information and addressing ambiguity	SQuAD	BLEU, ROUGE-L, METEOR	English
Kriangchaivech and Wangperawong [2019]	Generation of questions using Transformers, with no hadcrafted templates	SQuAD	WER	English
Zhang et al. [2020]	Combines multiple methods (PoS tagging, NER, SRL, regular expressions, domain knowledge bases, word embedding, word edit distance, WordNet, ...) to generate distractors	US SAT (Scholastic Assessment Test)	Human	English
Ch and Saha [2018]	Survey about Automatic MCQ Generation from text	-	-	English
Kurdi et al. [2020]	Survey about AQG for educational purposes	-	-	English
Wang et al. [2020]	Combination of answer-aware and answer-agnostic approaches in a joint model	SQuAD	Human, BLEU, METEOR, ROUGE	English

# Chapter 4

## A Pipeline for Question Generation

Recalling the main objective established in Chapter 1, we set out to develop a system comprised of the integration of computational tools for automatically generating MCQs from contents written in English. Generating questions with this system, even if minor adjustments are needed, would require less effort and time compared to generating the questions entirely from scratch.

From the reviewed literature, we identified different approaches and techniques for the task of Question Generation in the English language. In a early phase of the work, with the objective of experimenting some of the resources identified as useful for the development of the project and determine if they were feasible for future work, a preliminary experimentation was performed. Based on a workflow inspired by what was reviewed in some of the related works and by the preliminary experimentation, we outlined a pipeline to achieve our goals.

The proposed system pipeline, from plain text to generated MCQs, can be seen in Figure 4.1. It is divided into five steps: Pre-processing, Answer Selection, Question Generation, and Distractor Selection. Pre-processing is responsible to prepare the text for the next steps. In Answer Selection, we select answer candidates from the text and that will serve as the basis to generate the questions. Question Generation, as the name suggests, is composed of the methods responsible to create the text of the questions, that is, the stems. We consider answer selection and question generation to be an iterative "block". That means that the generation of a question occurs immediately after the selection of an answer. For each answer, the system generates the possible questions, and only then proceeds to the next answer. It is also in this step that we differentiate between the two approaches. At last, in Distractor Selection we resort to methods that select candidate distractors from the text or from external sources. In this section, we describe each of the steps of the workflow.

Despite the proposal of two approaches to generate questions (excluding the various options available to select distractors that are available to both), the majority of the steps are common to both, with the difference being in Question Generation. While both approaches are based on existing works, the decision to include both was not only to compare them but also because of the control we have over their performance. In the rule-based approach, we have full control of

how it works, as the rules can be changed. The other approach, based on existing works that address the use of ANNs and already available Transformer models fine-tuned for our task, is more of a "black-box" option which we cannot change much about how they work. To change something, we would need to train the models from scratch. Examples of these fine-tuned models are a T5 model<sup>1</sup> or a solution built using multiple pre-trained models<sup>2</sup>, both available in HuggingFace.

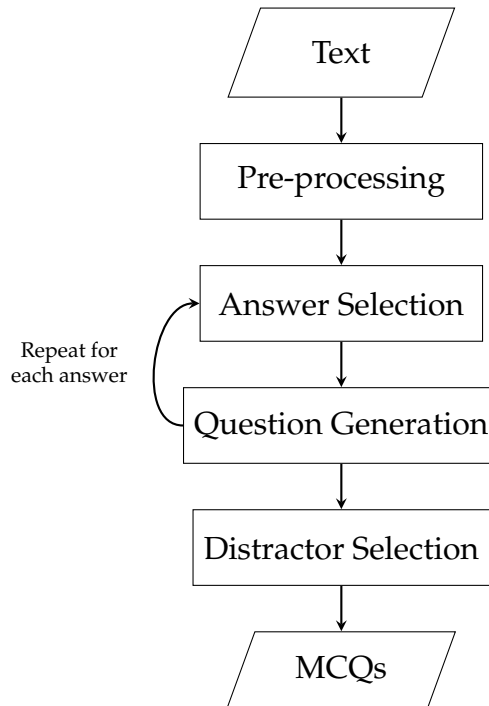


Figure 4.1: System workflow

In this chapter, we describe the system steps present in Figure 4.1 and how they were implemented with some examples. We also establish the types of evaluation used and the chosen metrics.

## 4.1 Pre-processing

In Pre-processing, the input written contents are prepared for the following steps. Let's consider text length. In case input text is considered to be too long (e.g., limitations of the length of the contexts to be used in Question Generation), it may be beneficial to divide it into smaller pieces. If it is already structured into subdivisions, like paragraphs and chapters, we can start by using those pre-existing divisions. In much of the literature, the text is broken apart so that each document is a single sentence. Here, we have our first choice: having a single sentence or multiple sentence documents (with "document" being each "block of text" given as input). Smaller documents can make the generation of questions advantageous,

<sup>1</sup><https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap>

<sup>2</sup>[https://github.com/AMontgomerie/question\\_generator](https://github.com/AMontgomerie/question_generator)

with less processing required. We verified that in reviewed literature a great number of the implemented methods were designed to only consider each sentence individually when generating questions. However, documents resulting from not dividing the text so much also have their perks. Having multiple sentences per document requires more complex co-reference resolution. By resolving pronouns found in sentences different from the ones the noun/noun phrases they refer to are, we can detect if a certain entity is present in more than one sentence. Doing this also helps diminish ambiguity. Considering this, some experiments will be performed to decide which will be our preferred choice.

Sorting the documents by the relevancy of their sentences, or selecting only a predefined number of the most important documents, can be a way to distinguish them and deal with time or processing constraints. Another related method that may help with the relevancy problem is summarization. When summarizing, the main ideas remain present in the text at the same time as they become less complex, which can be beneficial in later steps.

There are also simpler techniques that can be used in pre-processing, such as lemmatization or removal of stop words and punctuation. For example, after using lemmatization, some words that previously would not be recognized as answer candidates may be recognized afterward, causing the document to have a higher percentage of words from which answers can be derived.

In our implementation, we centered on co-reference resolution. Written contents commonly have pronouns or expressions that make more sense while in context, as they refer to other expressions present in the text. When generating questions automatically, such questions can include these pronouns, possibly making it difficult to fully understand them. To prepare the contents for the following phases of the pipelines, it is beneficial to substitute these expressions with the ones they refer to and that is more comprehensible while isolated, compared to the originals.

To perform co-reference resolution we used the "neuralcoref"<sup>3</sup> library. With this library, we are able not only to group text segments that refer to the same thing but also to retrieve the complete text with these expressions already replaced. In Figure 4.2 we have an example of how co-reference resolution changed the text of the first section of the article "Coimbra" from Wikipedia. Due to incompatibilities with another library used for question generation – each requiring different versions of the NLP library SpaCy – we could not incorporate the use of co-reference resolution in the same Python environment. So, as a second plan, a second environment with adequate libraries was created.

The written contents used in experimentation and evaluation were Wikipedia articles as the source text and the SQuAD dataset as reference. Specific to the Wikipedia articles, we pre-saved the articles so that we would not need to search them every time we would use them, as well as to circumvent the problem of using a different Python environment for co-reference resolution. To do so, we used the "wikipedia"<sup>4</sup> library. Besides each Wikipedia article being saved locally

---

<sup>3</sup><https://spacy.io/universe/project/neuralcoref>

<sup>4</sup><https://pypi.org/project/wikipedia/>

Coimbra (, also US: , UK: , Portuguese: [kuˈiβɾe] (listen) or [ˈkwĩβɾe]) is a city and a municipality in Portugal. The population of the municipality at the 2011 census was 143,397, in an area of 319.40 square kilometres (123.3 sq mi). The fourth-largest urban centre in Portugal (after Lisbon, Porto and Braga), it is the largest city of the district of Coimbra and the Centro Region. About 460,000 people live in the Região de Coimbra, comprising 19 municipalities and extending into an area of 4,336 square kilometres (1,674 sq mi). Among the many archaeological structures dating back to the Roman era, when **the Região de Coimbra** was the settlement of Aeminium, are **the Região de Coimbra** well-preserved aqueduct and cryptoporticus. Similarly, buildings from the period when **the Região de Coimbra** was the capital of Portugal (from 1131 to 1255) still remain. During the late Middle Ages, with **the late Middle Ages** decline as the political centre of the Kingdom of Portugal, **the Região de Coimbra** began to evolve into a major cultural centre. This was in large part helped by the establishment of the University of **the Região de Coimbra** in 1290, the oldest academic institution in the Portuguese-speaking world. Apart from attracting many European and international students, the University of Coimbra is visited by many tourists for **the University of Coimbra** monuments and history. **the University of Coimbra** historical buildings were classified as a World Heritage site by UNESCO in 2013: "**the Região de Coimbra** offers an outstanding example of an integrated university city with a specific urban typology as well as **the Região de Coimbra** own ceremonial and cultural traditions that have been kept alive through the late Middle Ages."

Coimbra -> the Região de Coimbra  
 its -> the Região de Coimbra  
 its -> the late Middle Ages  
 its -> the University of Coimbra

Figure 4.2: Co-reference resolution example in the Coimbra article from Wikipedia

in their original version, we also saved them with co-reference resolution (done in a different environment due to libraries' version incompatibility).

When using one of the articles, we loaded them using a function that splits the document based on the section titles. We decided to take advantage of the already existing text sections to divide the text. For each section, newlines were replaced by blank spaces.

## 4.2 Answer Selection

Once text is divided into smaller documents and their content is pre-processed, the next step is to identify terms or expressions that could be used as answers. As established in the pipeline (Figure 4.1), the system generates questions considering answers as a starting point. This is due to part of the methods used in the Question Generation approaches needing the answer to be obtained first, namely answer-aware transformers. To do this automatically, we can rely on statistic or linguistic information, such as TF-IDF, named entities, part-of-speech, and syntactic chunks. For that, we can use available tools for statistic and linguistic processing, such as SpaCy<sup>5</sup> or NLTK<sup>6</sup>.

Main concepts or ideas usually appear more times in texts about certain topics, especially if they are domain specific. Knowing this, word frequency can be a way to identify relevant terms. It is important to verify the existence of words that can have high frequency but are irrelevant as possible target answers, like stop words, and remove them. An evolution of that idea is TF-IDF. This metric does not only take into consideration the frequency of the word, but also the number of documents where it occurs. In that way, words with high frequency in a certain document but sparse in others can be understood as representative of concepts specific to the document and, therefore, be relevant.

<sup>5</sup><https://spacy.io/>

<sup>6</sup><https://www.nltk.org/>

Using PoS tagging or shallow parsing allows the identification of the syntactic function of the words. Then we can choose to use words from a certain class (e.g. nouns or noun chunks) as answer candidates. By using NER we can identify mentions to entities of a specific type (e.g. person, location, ...). The named entity type can be then used to determine how the sentence will be transformed into a question.

During the development, we tested multiple methods: terms (single words), clauses, bigrams, trigrams, named entities and noun chunks, as well as a combination of terms, named entities and noun chunks. We also experimented with transformers for this task. To do that, we isolated the answer selection part from pre-existing question generation pipelines<sup>7</sup>.

All of these methods were evaluated. Based on the results (see Chapter 5), the final version includes named entities, noun chunks and candidates selected by a transformer.

In our implementation we consider each sentence individually while looking for answer candidates. In the case of named entities, we use SpaCy to detect these expressions, returning a list of the candidates found as well as their named entity labels.

The process is similar to noun chunks, where the same library is used. However, as in subsequent steps it matters if an answer (or part of it) corresponds to a certain label, we try to associate each noun chunk with a named entity label. For that, we start by looking for named entities identical to the noun chunk. If one is found, we attribute to the noun chunk the label of the named entity. If not, we still try to find a label by repeating the process, but this time for the tokens of the noun chunk. To obtain the tokens, we process the text, removing punctuation, tokenizing and excluding stopwords (using NLTK). If there are tokens identified as a certain label, we attribute to the noun chunk the first found.

Our transformer method is based on an already existing prepend question generation pipeline<sup>8</sup>. We isolated the part of the pipeline that selects answers based on a given context. Then we try to attribute a named entity label to the answers in the same way we did it to noun chunks.

## 4.3 Question Generation

After obtaining a candidate answer and the corresponding context (e.g., sentence), the objective is to generate the text of the question, that is, the stem. As referred to before in this chapter, the pipeline (Figure 4.1) was initially projected so that, for each selected answer, the system would generate the possible questions and then proceed to do the same for the next candidate.

Some systems replace the term in the sentence that will be the answer with a blank space. In that case, the sentence remains declarative. However, we estab-

---

<sup>7</sup>[https://github.com/patil-suraj/question\\_generation](https://github.com/patil-suraj/question_generation)

<sup>8</sup>[https://github.com/patil-suraj/question\\_generation](https://github.com/patil-suraj/question_generation)

lished in a earlier phase of the work that we wanted our system to transform the sentence into a question, as it seemed an interesting task that would add value to the work. Rule-based approaches transform sentences by modifying and rearranging their words. To do so, the structure of the sentence must match pre-existing rules, usually handcrafted. This includes transformations based on subject-verb (SV) and subject-verb-object (SVO) structures, as well as Semantic Role Labelling (SRL). To apply the rules, the identification of PoS tags, chunks and named entity types can be helpful.

An alternative could be the use of templates. These are usually simpler models, with generated questions resulting from the addition of a certain component from the original document to an almost ready-to-use question. However, this type of approach would need the production of a high quantity of handcrafted templates, making the system less autonomous than using rules, that are more adaptable to different contexts.

As mentioned before, the same workflow is considered when resorting to Transformers. Transformers are considered state-of-the-art models. There are several models fine-tuned for question generation in English, available out-of-the-box and easy to use. As mentioned before, the HuggingFace website is where several models are available and which can be used in Python with the "transformers" library. The main con of Transformers is the less control we have. Unlike rules, Transformers already trained and fine-tuned cannot be changed or adapted, being kind of a "black box", unless we train them with other hyperparameters or new data.

In theory, we consider generating the questions for each answer, then proceed to generate the questions for the next, and so on. Despite that, during development, we found it to be more intuitive to first find all the candidate answers and then proceed to generate the questions for all these answers. However, as in pre-processing we split the source texts according to their sections, in practice we first find the answers for a section, then generate the questions for these answers, and proceed to perform this cycle for the remaining sections.

The implemented methods were based on rules and Transformers. While we resorted to transformers already fine-tuned for this task, the rule-based method was completely implemented by us, with the exception of the use of libraries to do linguistic analysis, such as finding clauses, named entities and so on. The rules are listed in Appendix A.

### 4.3.1 Rules

In our rule-based approach, questions are produced by giving as input the answer and correspondent label obtained in the answer selection phase, as well as the sentence that contains the answer.

The first step is detecting the clauses that compose the sentence. This can be achieved using the Claucy library. It is also relevant to detect the verb tense from the sentence, as well as if the clauses contain the verb, because we can only



generate questions from clauses with a verb. Beyond clause detection, the library also returns its type (SV, SVA, SVC, SVCA, SVO, SVOA, SVOC, SVOCA, SVOO or SVOOA), based on its components (subject, verb, direct object, indirect object, complement or adverbial). To notice that a clause may have an adverbial without the library reflecting it in the type. While implementing, we decided to join the rules between types whose difference is having or not an "A", as the same rules can be applied to both. In practice, it is as if we only consider five distinct types – SV, SVC, SVO, SVOC and SVOO – all of them possibly containing adverbials or not.

After determining the clause type, it is detected in which component the answer is present. For example, if the clause is of type SV or SVA, we check if the answer is present in the subject or in an adverbial. Or, if it is of type SVO or SVOA, we check for the answer in the subject, direct object or an adverbial. According to the type, where the answer is located and the verb characteristics (existence of helping verb, form of "to be" or "to have", ...), the respective rules are applied.

Much of what differentiates the rules is how verbs are treated. When the question is about the subject, that is, the answer is contained in the subject, we simply include the verb in the question as it is in the original sentence. In the other cases, there are multiple alternatives. For example, from the sentence "*Luís de Camões wrote a considerable amount of lyrical poetry and drama.*", with type SVO (Clacy output being <SVO, Luís de Camões, wrote, None, a considerable amount of lyrical poetry and drama, None, []>) and the answer being "*Luís de Camões*", by applying the rule "pronoun + verb + direct object + adverbials + ?" we can get the question "*Who wrote a considerable amount of lyrical poetry and drama?*".

If the verb is composed of only one word and is a conjugation of "to be", the verb is not altered. In this case, from the sentence "*Luís de Camões epic work was Os Lusíadas.*" of type SVC (Clacy output being <SVC, Luís de Camões epic work, was, None, None, Os Lusíadas, []>) and answer "*Os Lusíadas*", by applying the rule "pronoun + verb + subject + adverbials + ?", we can get the question "*What was Luís de Camões epic work?*".

When the verb is composed of at least two words and the helping verb is a conjugation of "to be" or "to have", the rule determines that they are split, with the helping verb put before the subject and the main verb after. However, for these cases we had some problems. For example, in the case of the verb being "*had been writing*", Clacy only identifies "*been writting*" as the verb. Similarly, for the verb "*had been*", the only part identified as a verb was "*been*". In these cases, we could not generate the questions as intended.

In the other cases, an auxiliary verb (according to the original tense), is put before the subject and the lemma of the original verb is put after. Returning to the example "*Luís de Camões wrote a considerable amount of lyrical poetry and drama.*", if we intend "*a considerable amount of lyrical poetry and drama*" to be the answer, we get "*What did Luís de Camões write?*". The auxiliary verb is determined according to the verb tense or the label of the answer. If the tense is "VDB", that is, past, the auxiliary is "*did*". This is the case of our example. If the tense is "VBZ", that is,

third person singular present, or if the label is "PERSON", the auxiliary is "does" (as we consider that it is referring to "He", "She" or "It"). The default is "do", as we consider that it is referring to "I", "You", "We" or "They".

The question pronoun used in a certain question is determined by the label of the answer. If the label is:

- "PERSON" or "ORG", the pronoun is "Who";
- "TIME", "DATE" or "ORDINAL", the pronoun is "When";
- "GPE", "FAC" or "LOC", the pronoun is "Where";
- "QUANTITY", "MONEY", "CARDINAL" or "PERCENT", "How much/many" is used.

If the answer label does not match any of the above (being for example "EVENT", "LANGUAGE", "LAW", "NORP", "PRODUCT" or "WORK\_OF\_ART"), or the answer does not have a label, the default "What" is used.

Figures 4.3, 4.4, 4.5 and 4.6 contain more examples of the process of generating questions with rules. From the initial sentence and the answer (in bold), the structure type and its components are determined. From there, according to the conditions referred above, the respective rule is applied.

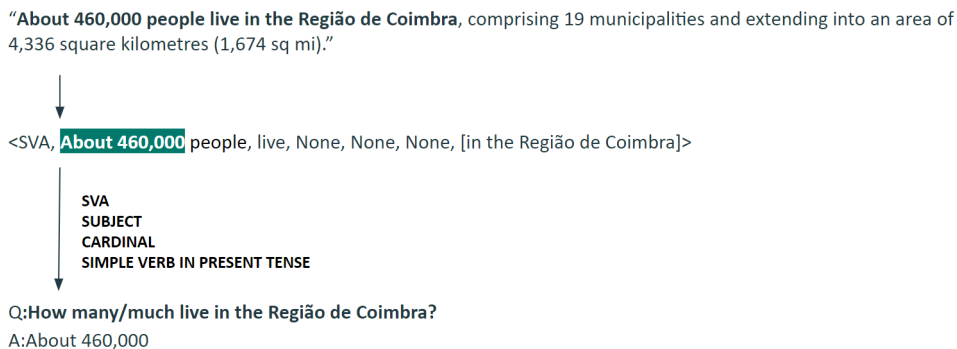


Figure 4.3: Example of question generated from a clause of type SVA, with the answer being a named entity of label CARDINAL present in the subject, and a simple verb in present tense

Later, with the question already generated, we check the pronouns with a masked language, also based on a transformer, BERT<sup>9</sup>. If a question contains "How many/much", it is needed to determine the most probable option. To do that, we replaced "many/much" by a "[MASK]" token. Then we give the altered string and the answer to determine what word should be there. In the case we are trying to resolve if the text should contain "much" or "many", we simply replace "much/many" by [MASK] and give the altered question plus the answer to the unmasker. It returns various alternative words, from which we obtain the one with the best score. Using the question "How much/many live in Coimbra?" as

<sup>9</sup><https://huggingface.co/bert-base-uncased>

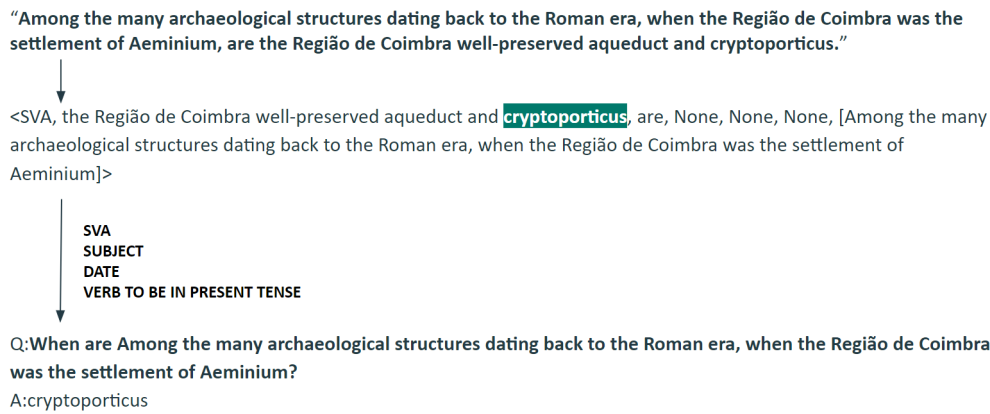


Figure 4.4: Example of question generated from a sentence of type SVA, with the answer being a named entity of label DATE present in the subject, and a verb "TO BE" in present tense

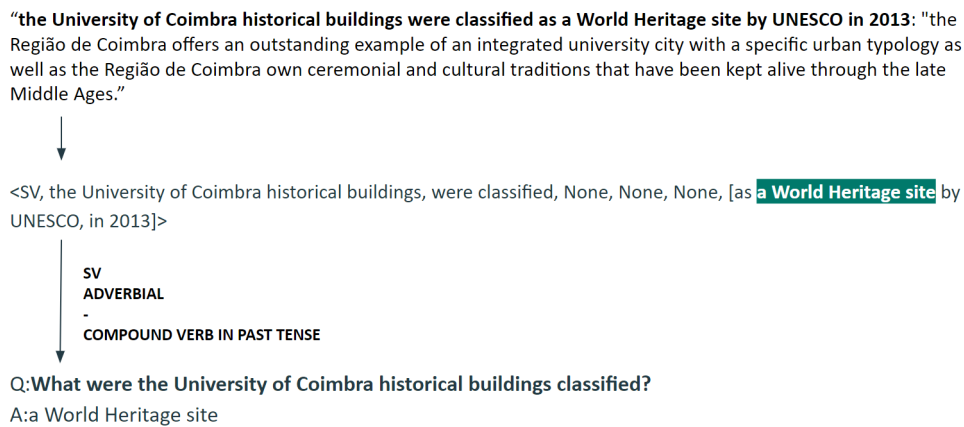


Figure 4.5: Example of question generated from a clause of type SV, with the answer being a noun chunk present in an adverbial without NE label, and a compound verb in past tense

an example, we transform it into "How [MASK] live in Coimbra?". The unmasker returns "live", with the final version of the question being "How many live in Coimbra?".

In the other cases, we repeat the same process twice. First only with the question, then with the question and the answer together. With the tokens returned by each of these cases, plus the original token that we are trying to determine if it is the more proper to the question, we use a voting system. In this voting system, we check which of them occurs the most, being the one that stays in the sentence. Using the question "What were the University of Coimbra historical buildings classified?", we check if the pronoun "What" is the correct. For this, we give the unmasker the string "[MASK] were the University of Coimbra historical buildings classified?", that is the original question with "What" replaced by "[MASK]", and "[MASK] were the University of Coimbra historical buildings classified? a World Heritage", composed by the same treatment to the question plus the answer. From these cases, we get two different outputs. For the first, we get "What", with the question being equal to the original. For the second, we get "Why", and the ques-

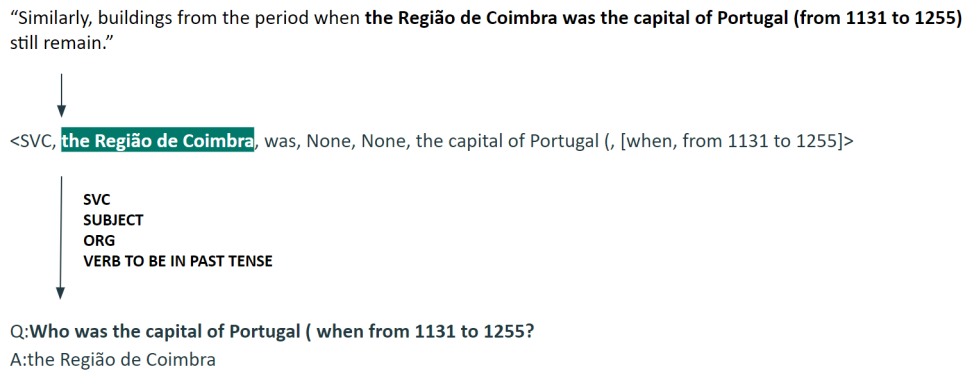


Figure 4.6: Example of question generated from a clause of type SVC, with the answer being a noun chunk present in the subject with NE label ORG, and a verb "TO BE" in past tense

tion would be *"Why were the University of Coimbra historical buildings classified?"*. As considering the totality of the options we got two "What" and one "Why", we remain with the "What".

### 4.3.2 Transformer

In an early phase of the work, with the objective of experimenting some of the resources identified as useful for the development of the project and determine if they were feasible for future work, a preliminary experimentation was performed. It was in this preliminary experimentation that we first used the transformer [Romero, 2021]<sup>10</sup>. This is a model based on T5 [Raffel et al., 2020] and fine-tuned on SQuAD v1.1 for AQG.

Later we experimented with some more from a Github repository<sup>11</sup>. This repository includes some answer-aware transformers (i.e., which require the answers for generation) – QG, QA-QG and QG Prepend – and one answer-agnostic (E2E). They are also based on the T5 model and trained on SQuAD v1. In this case, the answer-aware transformers also perform answer selection. QG was implemented for the single task of Question Generation. On the other hand, QG-QA can be used for both for Question Generation and Question Answering. While QG and QG-QA use highlights to identify the answer in the context (e.g., "*<hl> 42 </hl> is the answer to life, the universe and everything.*"), QG Prepend, as the name denotes, prepends the answer to the context (e.g., "*answer: 42 context: 42 is the answer to life, the universe and everything.*"). The highlights implementation is based on [Chan and Fan, 2019] and E2E is based on the ideas from [Lopez et al., 2020]. [Romero, 2021] also uses the prepend format.

In the final version of the system, we resort to [Romero, 2021]<sup>12</sup>, as it was the one with better results in Chapter 5. With the other answer-aware we had some problems in isolating the Question Generation part without deteriorated results,

<sup>10</sup><https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap>

<sup>11</sup>[https://github.com/patil-suraj/question\\_generation](https://github.com/patil-suraj/question_generation)

<sup>12</sup><https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap>

something that we needed to be able to use them with other Answer Selection Methods. We can also opt for the answer-agnostic transformer (E2E).

For example, by using a answer-agnostic transformer (E2E) to generate a question from the sentence *"Coimbra is a city and a municipality in Portugal."*, we get *"Coimbra is a municipality in what country?"*. Using a answer-aware transformer ([Romero, 2021]), we are able to choose a answer, like *"Coimbra"*, getting the question *"What is the name of the city in Portugal?"*. If we instead use *"Portugal"* as a answer, we get *"In what country is Coimbra located?"*.

## 4.4 Distractor Selection

Distractors are an essential part of a MCQ. We identified two ways to select distractors: extracting words or phrases from the same document the questions were generated from, or resorting to external sources, like knowledge bases and ontologies. To select from the same document, we identified name entities of the same label as the correct answer. To obtain distractors from external sources, we resorted to GloVe (Pennington et al. [2014]), WordNet (Fellbaum [1998]), DBpedia (Auer et al. [2007]) and a Transformer (voidful/bart-distractor-generation-both<sup>13</sup>).

### 4.4.1 Named Entities

This method receives as input the correct answer and its named entity label, the text from where it was retrieved (in the case of our tests, not only the section but the entire Wikipedia article), a penalty value (to be applied when the distractor or a distractor token is synonym with the answer or answer token) and the number of distractors we want to select.

We start by verifying if the answer has a named entity label. If not, we proceed to determine if the answer or its tokens have one (in this case tokens not necessarily being only single words but also substrings that correspond to an entity with a respective label). Having now one or more named entities from the correct answer, we proceed to verify the existence of named entities in the text. For each pair <correct answer, candidate distractor>, we check if they are different strings (after processing), if they have the same label and if the candidate distractor is not already in the distractors list. The processing applies lowercasing, punctuation removal and stop words removal (the last using the NLTK list). If the conditions verify, their similarity is checked via SpaCy, that uses a method based on Word2vec. If they have common or synonym tokens, a penalty is applied to the value of the similarity. To identify synonymous, we resort to WordNet.

If the number of possible distractors is less than the number of desired distractors, we proceed to apply a similar process, this time for the tokens of the answers.

<sup>13</sup><https://huggingface.co/voidful/bart-distractor-generation-both>

After this, the list of possible distractors is sorted in descending order, with the first ones returned. In our tests, we limited the number of returned distractors to five.

For example, by selecting named entities as distractors for the answer "the Arctic Ocean", a named entity with label LOC (location), we got:

- "the Gulf Stream"
- "North Atlantic Drift"
- "the Northern Hemisphere"
- "the Emba River"
- "the Southern Hemisphere"

#### 4.4.2 GloVe

Using the Gensim library, we were able to obtain a word embedding model pre-trained with Wikipedia data ('glove-wiki-gigaword-100'). We chose this model mainly because we chose to test the system with Wikipedia articles, and so a model trained with Wikipedia data seemed suitable. There were other models trained with more or less quantity of data, with we choosing this model as the compromise between not being the model trained with less data but also not being the one with the bigger size, what would result in taking more time to perform the same task.

We start by preprocessing the correct answer as in the previous approach, obtaining a version of it in lower-case, without punctuation and stopwords, as well as obtaining its tokens.

First, we get the  $n$  most similar words to the answer according to word embeddings. In case the number of words selected as distractors is not sufficient, we try to get more according to the processed version of the correct answer. If it still is not sufficient, we repeat the process, this time for the tokens of the answer. However, in this case, the distractors will not be the most similar words, but versions of the correct answer with the token replaced with the correspondent alternative word.

For each distractor, we then verify if it has synonyms common to the correct answer or if they share tokens (resorting to WordNet to obtain the synonyms). In these cases, the similarity score takes a penalty.

In the end, the distractors are sorted in descending order according to the similarity score, with the first  $n$  being returned.

For example, by generating distractors with GloVe for the correct answer "the Kingdom of Portugal", we got:

- "the kingdom of spain";

- "the kingdom of brazil";
- "the kingdom of argentina";
- "the kingdom of italy";
- "the kingdom of greece".

In this example, the method only returned "spain", "brazil", "argentina", "italy" and "greece", with the distractors being the replacement of "Portugal" with these tokens in the answer.

#### 4.4.3 WordNet

In the case of distractors selected via WordNet, we only give as input the correct answer, the penalty and the number of distractors we want to select.

The correct answer is pre-processed, getting the answer lower-cased and without stopwords and their tokens (also not considering stopwords).

Then we proceed to get the cohyponyms of the answer. In this process, the blank spaces are replaced by underscores, as in WordNet synsets whose names have multiple words use underscores instead of blank spaces. First, we get the synsets the expression belongs to and, for each of them, we get the hypernyms. Then, we get the hyponyms of each hypernym. To prevent having distractors with the same meaning of the correct answer, we verify if they are not synonyms. After getting the names of the lemmas of each cohyponym synset, and replacing the underscores with blank spaces, the cohyponyms are returned.

If the number of retrieved cohyponyms is less than the necessary number, we proceed to repeat the process for the pre-processed version of the answer. If after that the number of distractors is not still sufficient, we do that again for each token.

After getting the distractors, we filter them. For each, the similarity with the correct answer is calculated and, if they have tokens in common, the penalty is applied. The list of distractors is then sorted by descending order of similarity and the first  $n$  are returned.

For example, using this method to generate distractors for the answer "Physics and Infrared Astronomy", we got:

- "earth science";
- "chemical science";
- "life science";
- "optics";
- "chemistry".

#### 4.4.4 DBpedia

We also applied the concept of cohyponyms in the DBpedia method. Initially, we tried with multiple proprieties, searching for broader concepts (hypernyms), concepts of the same type or related to the same class. After some experimentation, only the distractors retrieved with "skos:broader" seemed good enough to be distractors. As a consequence of this, we opted to only obtain them via this propriety.

As in the WordNet approach, the function receives the answer, the number of distractors wanted and the penalty. The answer is preprocessed in a similar way to that described before, obtaining a processed version of the answer and its tokens.

To better cover the possible results retrieved from DBpedia queries, we preprocess the answer so that we can use a version with the first character upper-cased and the other version lower-cased, as DBpedia labels revealed to be case-sensitive.

Firstly, we query for URIs that have as a label our possible answer, retrieving DBpedia labels. Then, for each of those labels, we query for broader concepts of the URIs obtained using "skos:broader". Inversely, we proceed to, for each of the broader concepts, obtain the narrower concepts. If the narrower concept is not already in the distractors list and is different from the answer, we proceed to add it to the distractors list.

Similarly to the previous methods, if the number of distractors obtained is not at least the same as the desired, we first repeat the process with the processed correct answer and, if still not sufficient, with the tokens.

With all the distractor candidates, we filter them in the same way as already described. For each <correct answer, distractor candidate> pair we calculate their similarity or if they have synonyms or tokens in common. If positive, a penalty is applied to the similarity score. In the end, the best-scored candidate distractors are returned.

Using as example the answer "holographic recording", resorting to DBpedia gave us the following distractors:

- "Video storage";
- "Computer data storage";
- "Data synchronization";
- "Data modeling";
- "Data quality".



### 4.4.5 Transformer

We resorted to an already fine-tuned BART transformer<sup>14</sup>, based on the ideas from [Chung et al., 2020]. It was trained with the dataset RACE ([Lai et al., 2017]).

To obtain distractors, it is necessary to include in the input both question and answer ("*question answer*"). One of the disadvantages of this method is that this input has a maximum length limit of 1024 characters. The other is, when generating multiple distractors, the possibility of some of them being repeated.

While in the paper this transformer was based on (Chung et al. [2020]), the examples include interrogative sentences, the dataset used to train it includes declarative sentences. Our questions not being in the same style as the sentences found in RACE may be the explanation to some less good results, like the repeated distractors. However, we did not find many models trained for this task easily available, and so decided to use it anyway as resorting to a transformer to perform this task remained interesting.

For example, by selecting distractors for the answer "Roger Taylor", we got the following distractors:

- "John Deacon";
- "Queen";
- "Freddie Mercury";
- "Jack Deacon";
- "The British band".

## 4.5 Summary

In this section, we presented the pipeline established to develop a system capable of generating MCQs automatically for the English language. Each of the steps that compose the pipeline – Pre-processing, Answer Selection, Question Generation and Distractor Selection – are described, referring to how we could approach the development of each of them, its methods and what was effectively implemented.

In Chapter 5 we present in detail the evaluation performed to compare the different methods and their results. This evaluation was comprised of both automatic evaluation and evaluation based on human opinion.

---

<sup>14</sup><https://huggingface.co/voidful/bart-distractor-generation-both>



# Chapter 5

## Evaluation

While reviewing related works, we identified two types of evaluation metrics that we could use to access our system: evaluation performed automatically and evaluation based on human opinion. In this section, we describe how we performed both types of evaluation, the metrics used, the results obtained, and what we concluded from them.

Throughout both the development of the project as well as in its conclusion, it was necessary to evaluate the performance of the approaches that were considered during the development or effectively implemented. Evaluation during the development served mainly for, within the multiple approaches considered, deciding those we should focus on. In the final phase of the work, the evaluation was performed so that we could draw conclusions about the performance of the developed system.

During the development of the project, we analyzed various answer selection methods, as well as multiple models of question generation, using automatic evaluation. In automatic evaluation, we applied metrics identified in the reviewed scientific literature (BLEU and ROUGE), presented in Chapter 3. This was possible because of the existence of reference data, like SQuAD<sup>1</sup> (Stanford Question Answering Dataset). This dataset is composed of Wikipedia article passages (we also refer to them as paragraphs), where for each passage there is a list of answers, their location on the paragraph and the corresponding question created by humans. We can consider them as question-answer pairs, in which the answer is stated once or more in the paragraph. In Answer Selection, we resorted to additional metrics so we could draw more conclusions, as detailed in Section 5.1.

Regarding the final evaluation, we resorted to human opinions. In addition to this evaluation allowing us to draw conclusions about the performance of the system, it enables us to do it considering its end users. As humans will be the end users and biggest beneficiaries of the system, it makes sense for the system to be evaluated by them. This was performed in two ways. Using forms mainly distributed to people related to the project (IPN and Mindflow), we tried

---

<sup>1</sup><https://rajpurkar.github.io/SQuAD-explorer/>

to obtain data relative to the quality of the generated questions and which one of the distractor selection methods gave more relevant results. To do so, we used Wikipedia articles about generally well-known thematic. We were also asked by Mindflow to perform a more profound analysis based on the same articles, but considering a greater amount of text and, consequently, more generated questions (only the stems, not considering the distractors). This more detailed evaluation ended up being performed by the author of this thesis.

## 5.1 Automatic Evaluation

Automatic evaluation is the fastest way of getting some conclusions on the performance of the methods, which may help in selecting the best methods. Moreover, automatic evaluation is replicable and subjects all methods to the same task and evaluation metrics. To know what would be the best-suited answer selection methods to be implemented, we tested the following:

- Terms: individual words (unigrams);
- Bigrams: groups of two words;
- Trigrams: groups of three words;
- Named Entities: phrases that are classified into a certain group (e.g., persons, geographic sites, dates, ...);
- Noun Chunks: nouns and the words describing the noun;
- Clauses: part of a sentence, or a sentence itself;
- T+NEs+NCs: Terms, named entities and noun chunks.

We used as a reference the dev set of SQuAD v1.1. The dataset is constituted by a set of passages and, for each passage, a series of answerable questions and their answers are listed. In Figure 5.1 we have a portion of the dataset. The example contains a passage (context) from a Wikipedia article, questions about the passage created by humans and answers identified in the text that can be used as reference. For example, for the answers *"Mediterranean"* or *"a Mediterranean climate"*, *"What kind of climate does southern California maintain?"* is an example of a question that can be generated.

### 5.1.1 Automatic Evaluation of Answer Selection Methods

For the task of Answer Selection, all of the methods referred above were tested with and without stop words (Table 5.1 and Table 5.2). That is, we tried to see if the presence of stop words would influence the scores. The answers were also sorted accordingly with TF-IDF. The metrics were the average of the values for each paragraph of the following:

```

{"context":"Southern California contains a Mediterranean climate, with infrequent rain
and many sunny days. Summers are hot and dry, while winters are a bit warm or mild and
wet. Serious rain can occur unusually. In the summers, temperature ranges are 90-60's
while as winters are 70-50's, usually all of Southern California have Mediterranean cli-
mate. But snow is very rare in the Southwest of the state, it occurs on the Southeast
of the state.",

"qas":[

{"answers":[{"answer_start":31, "text":"Mediterranean"}, {"answer_start":29, "text":"a
Mediterranean climate"}, {"answer_start":31, "text":"Mediterranean"}], "question":"What
kind of climate does southern California maintain?", "id":"5705fc3a52bb89140068976a"},

{"answers":[{"answer_start":59, "text":"infrequent rain"}, {"answer_start":59,
"text":"infrequent rain"}, {"answer_start":59, "text":"infrequent rain"}], "ques-
tion":"Other than many sunny days, what characteristic is typical for the climate in
souther California?", "id":"5705fc3a52bb89140068976b"},

{"answers":[{"answer_start":243, "text":"60's"}, {"answer_start":243, "text":"60's"},
{"answer_start":243, "text":"60's"}], "question":"What is the low end of the temperature
range in summer?", "id":"5705fc3a52bb89140068976c"},

{"answers":[{"answer_start":353, "text":"very rare"}, {"answer_start":353, "text":"very
rare"}, {"answer_start":353, "text":"very rare"}], "question":"How frequent is snow in
the Southwest of the state?", "id":"5705fc3a52bb89140068976d"},

{"answers":[{"answer_start":269, "text":"70"}, {"answer_start":269, "text":"70"}, {"an-
swer_start":269, "text":"70"}], "question":"What is the high end of the temperature
range in winter?", "id":"5705fc3a52bb89140068976e"}}

]

```

Figure 5.1: Example from SQuAD: Passage (context) from a Wikipedia article, examples of questions about the passage, and potential answers identified in the text (and their location in the text)

- All: proportion of the candidate answers present in at least one of the answers for the correspondent passage;
- Top 10: proportion of the ten best-scored candidate answers (accordingly to TF-IDF) present in at least one of the answers for the correspondent passage;
- Last Position (LP): position of the last candidate answer that appears in at least one of the answers for the correspondent passage;
- BLEU-1 (B-1), BLEU-2 (B-2), BLEU-3 (B-3) and BLEU-4 (B-4);
- Rouge-L (R-L).

BLEU (1, 2, 3, and 4) and ROUGE-L are metrics commonly used in AQG that compare the similarity between two segments of text using concepts such as n-grams and longest common sub-sequences, allowing us to compare candidates and references. They were used in the evaluation of both Answer Selection and Question Generation steps. To perform the evaluation with each of these metrics, all candidates (selected answer or generated question) are compared with all references (answers identified or questions suggested, respectively) that belong to the same passage. Then, the evaluation score is the result of the average of the values of each comparison. As these are some of the most commonly used metrics for this type of task, they have high relevance.

Table 5.1: Comparison of Answer Selection Methods

	All	Top 10	LP	B-1	B-2	B-3	B-4	R-L
<b>Terms</b>	0.2595	<b>0.4179</b>	73.2636	0.1901	0.0013	0.0013	0.0013	0.0178
<b>Named Entities</b>	<b>0.3509</b>	0.3635	9.8737	<b>0.3829</b>	<b>0.1598</b>	0.0755	0.0379	<b>0.0615</b>
<b>Noun Chunks</b>	0.2255	0.2620	26.9229	0.3212	0.1538	0.0643	0.0233	0.0485
<b>Clauses</b>	0.0186	0.0196	<b>5.9473</b>	0.2383	0.1023	0.0597	<b>0.0402</b>	0.0607
<b>Bigrams</b>	0.1200	0.1722	96.3537	0.2765	0.1235	0.0010	0.0010	0.0387
<b>Trigrams</b>	0.0795	0.1039	102.3848	0.2850	0.1316	<b>0.0859</b>	0.0009	0.0498
<b>T+NEs+NCs</b>	0.2471	0.3969	97.3652	0.2527	0.0615	0.0224	0.0075	0.0276

In addition to these metrics, for Answer Selection and after sorting the candidate answers based on their TF-IDF values, we also determined the proportion of the candidate answers present in at least one of the reference answers for the correspondent passage (All), the same only for the ten best-scored candidate answers according to TF-IDF (Top 10) and the position of the last common answer to both candidates and references sets (LP). We used these additional metrics so that we could draw conclusions not only about how similar the selected answers were to the ones present in the dataset but also if they were effectively present (as being equal to or contained by a reference answer). By restricting to the ten best-scored according to TF-IDF, we tried to analyze if the proportion of candidates supposedly more specific to their passage was bigger than when considering all of them. By registering the average of the last positions, we tried to have an idea of how distributed (more "concentrated" or more "dispersed") were answers common to both sets according to their importance.

In Table 5.1 we have the scores obtained from all the mentioned metrics to all the answer selection methods. The best value, for each metric, is in bold. Observing the table, we can verify that for the metric "All" (proportion of the candidate answers present in at least one of the answers for the correspondent passage), the "Named Entities" method has the best result. Considering only the ten best-scored candidates, "Terms" appears in first, the method that joins terms, named entities and noun chunks in second, and the method that only considers named entities in third. Considering the position of the last candidate answer that appears in at least one of the answers for the correspondent passage, the first is "Clauses", followed by "Named Entities". Note that, the lower this score is, the better. In BLEU-1, BLEU-2 and ROUGE-L, the "Named Entities" method appears as the best classified, while in BLEU-3 and BLEU-4 the same method appears as the second best (with "Trigrams" as first in BLEU-3 and "Clauses" as first in BLEU-4). Generally, we can consider that the Named Entities method stood out, considering that at a great number of metrics it was the best scored and, even in the cases when it was not, it was classified as the second or third best method.

We repeated the same evaluation process, this time excluding stop words from both selected and reference answers, to verify if their presence influenced the scores. In general, we can say that the results are quite similar. The differences reside in BLEU-3 and BLEU-4. In BLEU-3 the best-scored method was "Noun Chunks", overcoming the results of "Trigrams", despite both being quite close. Considering BLEU-4, the new best-scored method is also "Noun Chunks".

Table 5.2: Comparison of Answer Selection Methods (without stop words)

	All	Top 10	LP	B-1	B-2	B-3	B-4	R-L
<b>Terms</b>	0.2569	<b>0.3861</b>	51.7271	0.2083	0.0018	0.0018	0.0018	0.0242
<b>Named Entities</b>	<b>0.3493</b>	0.3549	8.6710	<b>0.3694</b>	<b>0.1655</b>	0.0572	0.0259	<b>0.0662</b>
<b>Noun Chunks</b>	0.2047	0.1998	17.1316	0.2899	0.1618	<b>0.0731</b>	<b>0.0315</b>	0.0556
<b>Clauses</b>	0.0128	0.0128	<b>3.1532</b>	0.2190	0.0983	0.0443	0.0255	0.0631
<b>Bigrams</b>	0.1207	0.1407	61.2015	0.2664	0.1282	0.0016	0.0016	0.0474
<b>Trigrams</b>	0.0636	0.0692	58.3816	0.2716	0.1367	0.0725	0.0016	0.0588
<b>T+NEs+NCs</b>	0.2436	0.3522	70.0875	0.2549	0.0716	0.0263	0.0098	0.0345

It makes sense that for both groups of tests, the methods that selected single words ("Terms" and "T+NEs+NCs") had the second and third best scores in "All" and were the better evaluated in "Top 10" (excluding "NEs" that was the second best when not considering stopwords). In these metrics, we consider that a candidate answer is present in the references not only if it is equal to a reference answer, but also if it is contained in one. Because of this, single words have an advantage.

Regarding "LP", the fact that "Clauses" was the method better evaluated in both may be because of how unique each generated candidate is. As this method selects entire clauses as candidates, the chance of a candidate being equal or contained by a reference is low. Especially in regard to being contained, since the longest a candidate is and the more words it contains, the worse is the probability of that same sequence of words being present in a reference answer in the same order. It is likely that we have not penalized enough when candidates do not appear in the dataset, valuing methods that, despite generating fewer candidates that appear in the references, the few that appear have good TF-IDF scores.

It also makes sense that the method that selects answer candidates with exactly three words ("Trigrams") has the better score for BLEU-3, the metric that considers the number of common words between reference and candidate. The same applies to "Clauses", the method with the potential to select candidates of big dimension, and BLEU-4. However, as referred before, by removing stop words, the scores change a little. While trigrams and clauses have a high probability of containing stop words, noun chunks, which are primarily composed of nouns and adjacent words that describe the noun, are less likely to have a stop word. This might explain why, for metrics BLEU-2, BLEU-3, BLEU-4 and ROUGE-L, its scores increase.

Considering the values obtained in these two groups of tests, we can consider that, overall, the method more consistent in being the best-scored was "Named Entities". Because of this conclusion, from this group of methods, the selection of answers based on named entities was the one used in subsequent tests.

Still related to answer selection, as referred to in Chapter 4, we identified a GitHub repository with transformers capable of both answer-aware and answer-agnostic approaches. In the answer-aware approaches, we were able to isolate the part that performs answer selection. This allowed, for example, to mix answer selection performed by a transformer with question generation based on rules.

Table 5.3: Comparison of Answer Selection Methods (transformers)

	All	Top 10	LP	B-1	B-2	B-3	B-4	R-L
<b>QG</b>	<b>0.4365</b>	<b>0.4355</b>	<b>4.2199</b>	<b>0.4467</b>	<b>0.2705</b>	<b>0.1474</b>	<b>0.0884</b>	<b>0.0886</b>
<b>QG Prepend</b>	0.1200	0.1185	4.5787	0.4378	0.2617	0.1427	0.0864	0.0862
<b>QG-QA</b>	<b>0.4365</b>	<b>0.4355</b>	<b>4.2199</b>	<b>0.4467</b>	<b>0.2705</b>	<b>0.1474</b>	<b>0.0884</b>	<b>0.0886</b>

The answer-aware transformers are QG, implemented for the single task of Question Generation, and QG-QA, capable of both Question Generation and Question Answering. As described in Chapter 4, while QG and QG-QA use highlights to identify the answer in the context (e.g., "*<hl> 42 </hl> is the answer to life, the universe and everything.*"), QG Prepend prepends the answer to the context (e.g., "*answer: 42 context: 42 is the answer to life, the universe and everything.*"). By only considering the results of the answer selection component and evaluating it based on the same metrics as the previous methods, we obtained the scores present in Table 5.3.

In this table we can see that QG and QG-QA have the same results, meaning that for this task it does not matter which of them we choose. Both score much higher in comparison with QG Prepend when considering "All" and "Top 10". In the other metrics, they are also better, but with a lower difference. Comparing these results with the ones obtained with previous methods, QG and QG-QA present a better performance for all the metrics (using the results that do not exclude stop words). In the case of QG Prepend, it also did not score particularly well in "All" and "Top 10" in comparison with the methods from Table 5.1. However, it outscored them in all of the other metrics.

The use of highlights granted better results, but we found it more difficult to isolate the answer selection part of the pipeline than in the prepend transformer. Despite not having the best performance when compared to the other approach, as it still got better results in the majority of metrics compared to the earlier tested methods, we opted to resort to this transformer in subsequent methods when performing answer selection.

### 5.1.2 Automatic Evaluation of Question Generation Methods

We proceeded to compare question generation methods. For that, in addition to the answer-aware methods already evaluated regarding answer selection (QG, QA-QG and QG Prepend), we also tested an answer-agnostic transformer (E2E). The tests were performed without modifying any of them.

We also resorted to [Romero, 2021]. This answer-aware transformer uses the prepend format and is specific to question generation, meaning that we had to use a method to select answers. We chose QG Prepend to select candidate answers, as this was the transformer we better managed to isolate the answer selection parts as described earlier. The rule-based approach was also evaluated, using named entities and noun chunks for the task of answer selection.

The results, obtained via BLEU and ROUGE, can be seen in Table 5.4. We were



Table 5.4: Comparison of Question Generation Methods

	<b>B-1</b>	<b>B-2</b>	<b>B-3</b>	<b>B-4</b>	<b>R-L</b>
<b>QG</b>	0.5327	0.2409	0.1301	0.0774	0.2260
<b>QA-QG</b>	0.5403	0.2464	0.1347	0.0808	0.2291
<b>QG Prepend</b>	0.5459	0.2499	0.1348	0.0792	0.2336
<b>E2E</b>	0.5347	0.2434	0.1313	0.0804	0.2268
<b>[Romero, 2021] (answers from QG Prepend)</b>	<b>0.5576</b>	<b>0.2566</b>	<b>0.1389</b>	<b>0.0816</b>	<b>0.2377</b>
<b>Rules (answers from NEs)</b>	0.3853	0.1324	0.0638	0.0357	0.1549
<b>Rules (answers from NCs)</b>	0.3774	0.1223	0.0574	0.0311	0.1470

surprised by verifying that, for all metrics except B-4, QG Prepend scored better than the other two answer-aware transformers from the same repository (QG and QA-QG). E2E also got good results and, despite scoring a little lower in almost all the metrics compared to QG Prepend, got a better score for B-4. At every metric, [Romero, 2021] (with answers selected with QG Prepend) was the best-scored method.

The rule-based approach, for both named entities and noun chunks as answer selection methods, was the approach with the worst values. That was expected, given that transformers are considered state-of-the-art. However, the dataset used is also relevant. SQuAD is not exhaustive, that is, the questions present in SQuAD are only examples of what can be formulated for each passage. Because of that, good questions can get scores that do not represent them well as there may be no similar questions present in the dataset. Transformers also have the advantage of having been trained in a portion of this dataset, and so it is also expected that they can be better prepared to generate questions in the same style.

Once more, we confirmed the relevancy of selecting named entities instead of noun chunks, as that was the version that got the best scores between the two rule-based approaches.

## 5.2 Human Evaluation

We also performed evaluation based on human opinion. We resorted to forms answered mostly by people related to IPN and Mindlow, but also to a more extensive analysis done by the author of the thesis. This analysis considered a higher number of questions. Appendix B includes prints of the forms distributed to collect data for the evaluation.

To evaluate the questions we had to restrict the number of options considered in the system so that the number of forms (and the number of questions presented on the form) would not overwhelm the people answering them. Knowing this, we restricted the approaches according to the observations taken during implementation and the analysis of the results in automatic evaluation to the following options:

- Answer selection: named entities or transformer;

- Question generation: rules or transformer.

As the texts that serve as the source of the questions, we opted for Wikipedia articles in English. We chose articles that covered content relatively known by the general population so that the content of the questions would not be a barrier influencing the data collection and results. These were the articles used (versions from 14/07/2022):

- "Coimbra"<sup>2</sup>;
- "Europe"<sup>3</sup>;
- "Queen (band)"<sup>4</sup>;
- "Cristiano Ronaldo"<sup>5</sup>;
- "Star Wars (film)"<sup>6</sup>.

The human evaluation performed can be divided into two types: evaluation of question generation (and on a lighter note answer selection, as the questions generated depend on the correspondent answers), and evaluation of distractor selection.

### 5.2.1 Human Evaluation of Question Generation Methods

In the case of question generation, we started by collecting the data through forms distributed to a more general group of people, most of which related to IPN and Mindflow. But we were also asked to do a more extensive evaluation, with more questions generated from a larger quantity of text. In this case, the evaluation ended up being performed by the author of the thesis himself.

So, we had four combinations of approaches (two options for answer selection and also two options for question generation) for five different articles. Again, for the size of the information to be evaluated not to scale too much, we restricted the size of the information to a smaller context. In the more general evaluation, we restricted it to the first three sentences of each article. In the case of the more extensive evaluation, it was the first two sections of the article. In all of them, co-reference resolution was applied.

In the more general evaluation, we evaluated whether:

- Questions are of sufficient quality to be included in a questionnaire without major editing required;

---

<sup>2</sup><https://en.wikipedia.org/wiki/Coimbra>

<sup>3</sup><https://en.wikipedia.org/wiki/Europe>

<sup>4</sup>[https://en.wikipedia.org/wiki/Queen\\_\(band\)](https://en.wikipedia.org/wiki/Queen_(band))

<sup>5</sup>[https://en.wikipedia.org/wiki/Cristiano\\_Ronaldo](https://en.wikipedia.org/wiki/Cristiano_Ronaldo)

<sup>6</sup>[https://en.wikipedia.org/wiki/Star\\_Wars\\_\(film\)](https://en.wikipedia.org/wiki/Star_Wars_(film))

Table 5.5: Number of responses to forms

	Number of responses
<b>Coimbra</b>	21
<b>Europe</b>	12
<b>Queen (band)</b>	12
<b>Cristiano Ronaldo</b>	11
<b>Star Wars (film)</b>	10
<b>Total</b>	<b>66</b>

Table 5.6: Answer distribution for the statement: "Questions are of sufficient quality to be included in a questionnaire without major editing required."

	Strongly Disagree	Disagree	Agree	Strongly Agree
<b>ne_transformer</b>	7.6%	39.4%	33.3%	19.7%
<b>tr_transformer</b>	<b>6.1%</b>	<b>28.8%</b>	<b>40.9%</b>	<b>24.2%</b>
<b>ne_rules</b>	59.1%	<b>28.8%</b>	10.6%	1.5%
<b>tr_rules</b>	42.4%	39.4%	13.6%	4.5%

- The question set has good coverage in terms of possible questions to ask about the text;
- The questions presented are a good starting point for creating a questionnaire within this theme.

In Table 5.5 we present the number of responses to forms we got. In total, we got 66 responses (21 for the article "Coimbra", 12 for the article "Europe", 12 for the article "Queen (band)", 11 for "Cristiano Ronaldo" and 10 for "Star Wars (film)"). The tables and graphs presented gather the data collected from all five articles. This data makes no distinction between articles so the values are representative of the performance of each approach in general and not specific to a certain source text. Each approach is identified in the following way:

- **ne\_transformer**: named entities for answer selection, transformer for question generation;
- **tr\_transformer**: transformers for both answer selection and question generation;
- **ne\_rules**: named entities for answer selection, rules for question generation;
- **tr\_rules**: transformer for answer selection, rules for question generation.

Table 5.6 and Figure 5.2 show the the distribution of answers to the statement "Questions are of sufficient quality to be included in a questionnaire without major editing required". Probably what stands out the most is the great discrepancy of the "Strongly Disagree" bar in the approaches based on rules, in comparison to the approaches based on transformers for question generation. If we compare approaches regarding answer selection, we can see that "tr\_transformer" has better results than "ne\_transformer", and the same happens in the other pair, with "tr\_rules" having better results than "ne\_rules" (especially evidenced by the proportion of the "Strongly Disagree" bar). Sorting the approaches from the best to

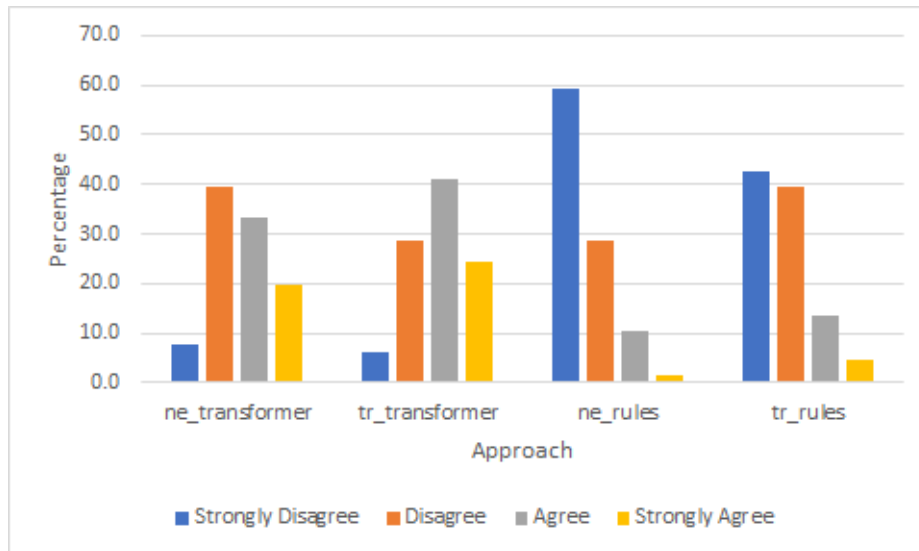


Figure 5.2: Answer distribution for the statement: "Questions are of sufficient quality to be included in a questionnaire without major editing required."

Table 5.7: Answer distribution for the statement: "The question set has good coverage in terms of possible questions to ask about the text."

	Strongly Disagree	Disagree	Agree	Strongly Agree
ne_transformer	0.0%	15.2%	51.5%	33.3%
tr_transformer	18.2%	24.2%	48.5%	9.1%
ne_rules	22.7%	31.8%	39.4%	6.1%
tr_rules	31.8%	42.4%	22.7%	3.0%

the worst evaluated, we can consider "tr\_transformer" was the better evaluated, followed by "ne\_transformer", then "tr\_rules" and at last "ne\_rules". This means that methods that used transformers to both results were better evaluated.

Observing Table 5.7 and Figure 5.3, we can see that with the exception of "tr\_transformer", all other approaches were better classified than in the previous statement. This is a good indicator of the coverage of the questions. The approaches that use the transformer for question generation obtained, again, better results than the rule-based. However, considering the answer selection methods, approaches that use named entities were better evaluated than the ones that resort to a transformer. We must highlight the fact that nobody chose "Strongly Disagree" for "ne\_transformer". Sorting according to the results, the approach better evaluated was "ne\_transformer", followed by "tr\_transformer", then "ne\_rules" and finally "tr\_rules".

Table 5.8: Answer distribution for the statement: "The questions presented are a good starting point for creating a questionnaire within this theme".

	Strongly Disagree	Disagree	Agree	Strongly Agree
ne_transformer	0.0%	9.1%	43.9%	47.0%
tr_transformer	7.6%	19.7%	48.5%	24.2%
ne_rules	18.2%	37.9%	36.4%	7.6%
tr_rules	30.3%	31.8%	31.8%	6.1%

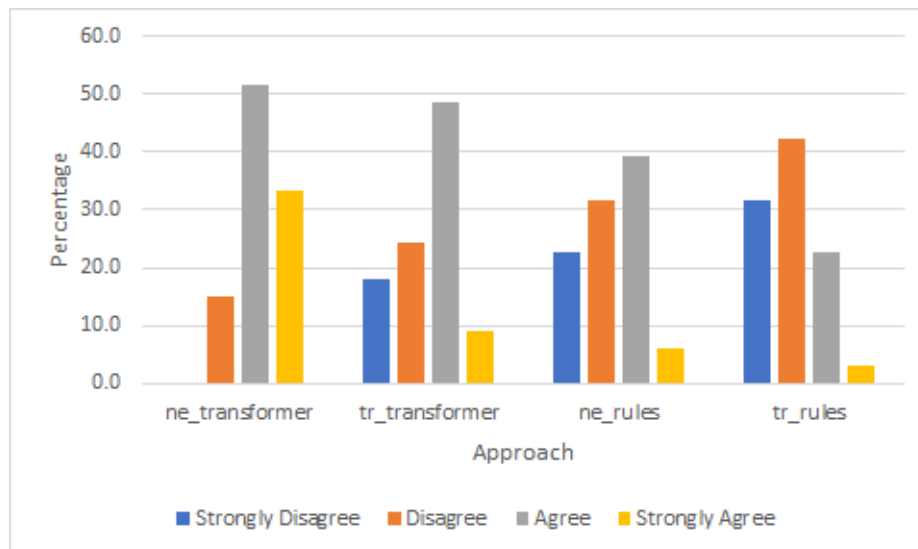


Figure 5.3: Answer distribution for the statement: "The question set has good coverage in terms of possible questions to ask about the text."

The objective of the results presented in Table 5.8 and Figure 5.4 was to figure out if the developed approaches were a good starting point to create questionnaires. Similar to the opinion of those who responded to the previous statement, we can see that approaches that implement a transformer instead of rules to generate questions were better evaluated and named entities were considered better than transformers as a method to select answers. Again, the percentage of people that chose "Strongly Disagree" for "ne\_transformer" was zero. So, in a descending order from best to worst evaluated, we have "ne\_transformer", "tr\_transformer", "ne\_rules" and "tr\_rules".

If we consider the negative opinions ("Strongly Disagree" plus "Disagree") versus the positive opinions ("Agree" plus "Strongly Agree") in all statements, for the approaches that use the transformer to generate questions, the positives always outweigh the negatives. For the rule-based approach, we verify the inverse. We can conclude that the approaches that resort to the transformer were always better evaluated than the rule-based approaches. However, opting for named entities or expressions selected by a transformer is not so straightforward. While the transformer for answer selection presented better results for "Questions are of sufficient quality to be included in a questionnaire without major editing required", named entities were considered better to create a set of questions with good coverage of the text and as a better starting point to create a questionnaire.

The main goal of this work is to develop a system capable of automatically generating MCQs, with the possibility of these questions needing minor adjustments. We can consider that at least "ne\_transformer" and "tr\_transformer" are in a good position. These approaches have a generally positive opinion regarding the quality and the coverage of the questions generated, as well as being a good starting point to create questionnaires. The positive responses on quality are especially important, as the statement included "without major editing required", which aligns with our goals.

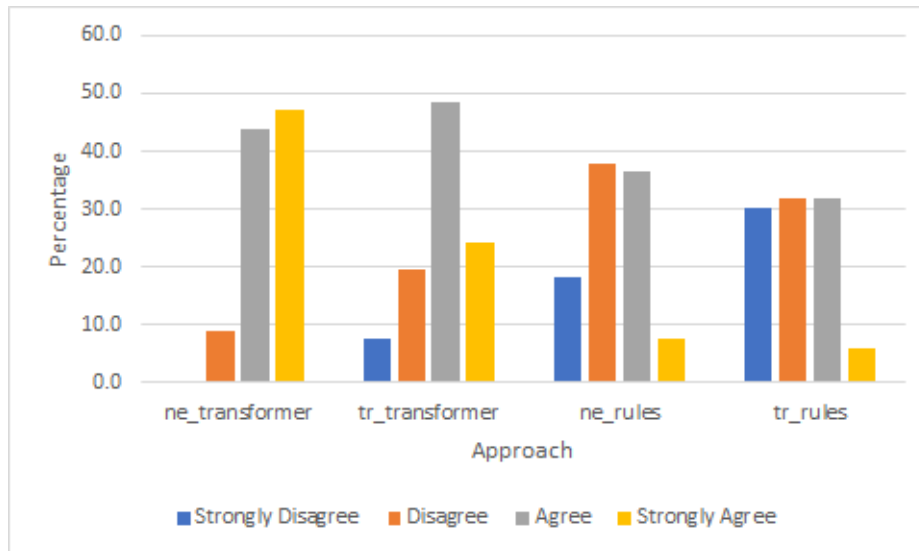


Figure 5.4: Answer distribution for the statement: "The questions presented are a good starting point for creating a questionnaire within this theme".

In the case of the more extensive question generation evaluation, with questions generated based on the two first sections of each article, we tried to get insights on the following:

- Question is well formulated;
- Question is pertinent;
- Answer is correct.

While analyzing the questions, we ended up creating and loosely following a guideline to help in the attribution of a positive label:

- Question well formulated:
  - Grammatically correct, without major errors, and allowing the presence of extra punctuation or simple terms that do not mislead the meaning of the sentence (e.g., "(", "that");
  - The question has the structure of the intended type, starting with a question pronoun and not requiring "Yes" or "No" as answers;
  - Without missing information about what is being asked to (e.g., pronouns in which we do not understand well what they refer to).
- Question is pertinent:
  - Important for the article and in accordance with the meaning of the original sentence;
  - It is possible to answer according to the sentence (or in case the question is not well formulated, understand what the meaning and answer of the sentence would be);

Table 5.9: Questions generated from the sentence "In 1949, the Council of Europe was founded with the idea of unifying Europe to achieve common goals and prevent future wars" using the approach "ne\_transformer"

Question	Answer	Well formulated	Pertinent	Correct answer
When was the Council of Europe founded?	1949	1	1	1
It was founded In 1949 with the idea of unifying Europe to achieve common goals and prevent future wars?	the Council of Europe	0	1	1
It was founded In 1949 with the idea of unifying Europe to achieve common goals and prevent future wars?	Europe	0	1	0

Table 5.10: Answer distribution for the statement: "Question well formulated."

	Disagree	Agree
<b>ne_transformer</b>	8.0%	92.0%
<b>tr_transformer</b>	7.6%	92.4%
<b>ne_rules</b>	61.5%	38.5%
<b>tr_rules</b>	55.7%	44.3%

- When necessary, the meaning of the original article was verified (without resolution of co-references).
- Answer is correct:
  - Of the expected "type" (e.g. if the question is "When...?" the answer must be a date);
  - There is sufficient information to know that it is correct;
  - Due to the way the answers are selected, it is accepted if incomplete (e.g., "Freddy Mercury" as answer to "Who did Queen comprise?").

Let's consider the sentence *"In 1949, the Council of Europe was founded with the idea of unifying Europe to achieve common goals and prevent future wars"*. From this sentence and resorting to the approach "ne\_rules", the questions present in Table 5.9 were generated. We considered the first question well formulated as it has no grammatical errors, is a "Wh-question" (in this case, "When...?") and has no missing information. We also considered it pertinent, as it is in accordance with the original sentence and can be answered. It also has a correct answer. The next two questions are the same but were generated from different answers. We cannot consider them well formulated, as they do not respect the type of question wanted. Despite this, they can be considered pertinent, as reading the original sentence we can understand how this question could be better formulated and what would be the correct answer. Regarding the answer, by reading the original sentence we can easily understand that *"the Council of Europe"* is a correct answer and *"Europe"* is not.

Considering the results to the statement "Question well formulated", present in Table 5.10, we can refer that the approach that achieved better-formulated questions was "tr\_transformer". Furthermore, the second best was also an ap-

Table 5.11: Answer distribution for the statement: "Question is pertinent."

	<b>Disagree</b>	<b>Agree</b>
<b>ne_transformer</b>	21.7%	78.3%
<b>tr_transformer</b>	<b>16.4%</b>	<b>83.6%</b>
<b>ne_rules</b>	44.1%	55.9%
<b>tr_rules</b>	34.9%	65.1%

Table 5.12: Answer distribution for the statement: "Answer is correct."

	<b>Disagree</b>	<b>Agree</b>
<b>ne_transformer</b>	26.4%	73.6%
<b>tr_transformer</b>	<b>19.6%</b>	<b>80.4%</b>
<b>ne_rules</b>	62.0%	38.0%
<b>tr_rules</b>	49.2%	50.8%

proach that uses a transformer for question generation. Discriminating the approaches based on the answer selection method, the ones that use a transformer performed better than the ones that select named entities. In order, from the best to the worst, we have "tr\_transformer", "ne\_transformer", "tr\_rules" and "ne\_rules".

In Table 5.11 we are presented with the results of which approach produces more pertinent questions, while in Table 5.12 the results are on whether the answer is correct for the question. Sorting the approaches based on the results, from the best to worst, we obtain the same order as in the previous results. For all of the statements, the descending order of evaluation was the following: "tr\_transformer", "ne\_transformer", "tr\_rules", "ne\_rules".

Combining the results of both question generation evaluations performed, we can conclude that generating questions resorting to a transformer always produced better results. However, considering answer selection, the results are more dividing, meaning that both methods (based on named entities or the transformer) are alternatives to be considered.

## 5.2.2 Human Evaluation of Distractor Selection Methods

Distractors are an important component of MCQs, and so their evaluation also has a high value. To evaluate the distractors generated by the implemented methods, we also resorted to forms.

In the creation of the form about distractor selection, we decided to restrict the number of questions presented. To have a good representation of the possible cases of answers we chose ten, each one with a different named entity label. Given that we chose five articles, the form contains two questions for each article. In total, twelve people answered the form. The content of the form can be seen in Appendix B.

For every question, distractors were selected based on named entities, WordNet, DBpedia, GloVe and a transformer. For each method, we obtained the following metrics:



Table 5.13: Distractors' form results

	Choice frequency	Production average	Proportion
<b>NER</b>	329	5	0.548
<b>WordNet</b>	226	3.5	0.538
<b>DBpedia</b>	121	2	0.504
<b>GloVe</b>	<b>339</b>	4.8	<b>0.589</b>
<b>Transformer</b>	232	4.2	0.460

- Choice frequency: number of times a distractor from the method was selected;
- Production average: Average number of distractors produced by the method. We established five as the limit of distractors generated by each method for each question as it seemed a number sufficiently high to generate a sample from which good distractors could be chosen, but low enough to not overwhelm those who were responding to the forms;
- Proportion (of good distractors): Division of "Choice frequency" by the result of the total number of distractors generated multiplied by the number of responses.

In cases the same distractor was selected by more than one method, if selected as a good distractor by the people that answer the form, its frequency increased in every method that generated it. Distractors that only differ in case, punctuation, spaces and little differences like "meters" and "metres" or "kilometers", "km", "kms" and "kilometre", were considered as a single distractor. In cases the same distractor appeared more than once in the same method (especially in the transformer), we also only counted it once.

The results can be seen in Table 5.13. GloVe is the method with more selected distractors, closely followed by NER. WordNet and Transformer have similar values, with Transformer being a little better. DBpedia was the method with the lowest number of distractors selected by the people who answered the form.

However, we can also justify these numbers based on the average number of produced distractors by a method for each question. DBpedia is clearly the worst, averaging in only two distractors produced. This is due to, in many cases, our DBpedia implementation not being able to generate any distractors, despite that, when it does, it generates all of the five. GloVe, despite the production average being lower, has a higher choice frequency when compared with NER. All of the others follow the logic of choice frequency and production average being higher or lower simultaneously.

To better analyze this we also calculated the proportion of good distractors of these two metrics. GloVe is indeed the method with better proportion. NER and WordNet, despite having a big difference in values for "Choice Frequency", have similar values for "Proportion". DBpedia, which had worse results, gets close to the former two. Transformer, which seemed to have reasonable "Choice frequency", is now the one which have the worst result. As we referred before, we only counted generated distractors that were different, being the repetition

the main problem of this method. To notice that almost all the methods achieved a proportion higher than 0.5, meaning that more than half of the distractors generated by these methods were considered good.

Specific to GloVe, one of its particularities is that, when the correct answer is composed of multiple words, instead of replacing the whole correct answer, it only does it for one of them. For example, having "*football player*" as the correct answer and searching for similar words to "*football*", one of them can be "*rugby*", with the final distractor being "*rugby player*". We think this might have positively influenced the results in some of the cases.

Based on these results, we can assume that NER and GloVe were the most relevant methods for selecting distractors, followed by WordNet. While NER and GloVe had good results for all "Choice frequency", "Production average" and "Proportion", WordNet had a good balance between distractors considered good and distractors selected. This means that despite generating fewer distractors, they were relevant. DBpedia is in a similar position of a good balance between the first two metrics, but their absolute value cannot be considered good enough. Transformer did not correspond to the expectations, which might be the result of the difficulties explained in Chapter 4.

### 5.3 Main Conclusions

In this chapter, we resorted to automatic evaluation to analyze the methods considered to be used during the development of this work, as well as an evaluation based on human opinion to draw conclusions of its final version.

Regarding the automatic evaluation of non-transformer Answer Selection methods, selecting named entities as candidate answers was revealed to be the most consistent method with the majority of the metrics. That was further confirmed when, for question generation using rule-based approaches, the method that used named entities scored better than the one that selected noun chunks.

As referred to at the end of both automatic and human Question Generation evaluations, when comparing the results of transformer approaches with rule-based approaches, the approaches that generated questions based on transformers always performed better. Answer selection, however, was more divisive. By comparing the method that generated questions based on named entities with the one that resorts to expressions selected by a transformer, we observed that the results were quite balanced. This means that both are alternatives to be considered for this task.

Regarding distractors, selecting named entities and expressions from GloVe or Wordnet revealed good results. In particular, the GloVe method stood out, especially when considering the proportion of good distractors. We can say that we achieved good results by selecting distractors from both the source text (NER) and resorting to external sources (WordNet and GloVe).

# Chapter 6

## Conclusion

In this work, we explored approaches to the task of Automatic Generation of Multiple Choice Questions. The main goal was to be able to develop a system that integrated various methods to automatically generate MCQs from given written content. The pipeline adopted to implement the system revealed to be a good option to integrate multiple types of approach to select answers, generate questions and select distractors.

Some of the approaches did not achieve the best results, namely the rule-based. Such results were expected, at least in comparison with Transformers, with the implementation of rules being more due to the possibility of having more control over the process of generating questions and serving as a baseline to the Transformer. However, being an approach made from scratch (using only some libraries that helped in Linguistic Analysis), it was interesting to implement. According to human evaluation, the approaches that used a Transformer for Question Generation were able to generate questions with sufficient quality to be included in a questionnaire without major editing required, with good coverage of the source texts used and that can serve as a starting point in the creation of questionnaires. Named entities or expressions selected by a Transformer in the task of Answer Selection were not evaluated so differently, both being good options. However, in the more extensive evaluation, approaches with answers selected by a Transformer performed a little better.

Regarding distractors, we were able to generate distractors from both the source text as well as from external sources. NER and GloVe were the methods with better performance, having the higher number of distractors chosen as well as having the highest average of produced distractors per question. Their results were followed by WordNet. The other methods, especially the one that resorts to DBpedia, need further development.

There are other aspects that can still be improved in future work. Related to non-machine-learning approaches to question generation, we can improve the rules, or even explore methods not included in our experimentation, like SRL. About distractors, we still need to be able to generate distractors that vary in "levels of incorrectness", with some more incorrect than others. We also did not implement methods to deal with the validation and ranking of the questions gen-

erated, a step of the pipeline that could improve the quality of the generated questions suggested to the user.

Overall, we were able to conduct a study that compared and mixed multiple methods of NLP applied to the generation of MCQs and created a system that, despite still needing some human intervention, stands as a good starting point to further developments. Having into account our main goals, we can conclude that the integration of various methods resulted in approaches with positive results for the task of AQG, composed by a pipeline capable of performing each of the sub-steps defined in the goals: Answer Selection, Question Generation and Distractor Selection.

The further development of systems like this might present many benefits in the future. Improving already existing methods and considering more types of questions and their difficulty, a system like this seems to have the potential to improve the creation of test and questionnaires, making this task take less time and be a complementary tool to aid in the context of education and training.

# References

- Husam Ali, Yllias Chali, and Sadid A Hasan. Automatic question generation from sentences. In *Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts*, pages 213–218, 2010.
- T Alsubait. *Ontology-based multiple-choice question generation*. PhD thesis, School of Computer Science, The University of Manchester, 2015.
- Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. Ontology-based multiple choice question generation. *KI-Künstliche Intelligenz*, 30(2):183–188, 2016.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.
- Satanjeev Banerjee and Alon Lavie. 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, June 2005. Association for Computational Linguistics. URL <https://aclanthology.org/W05-0909>.
- Jonathan Brown, Gwen Frishkoff, and Maxine Eskenazi. Automatic question generation for vocabulary assessment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 819–826, 2005.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Dhawaleswar Rao Ch and Sujana Kumar Saha. Automatic multiple choice question generation from text: A survey. *IEEE Transactions on Learning Technologies*, 13(1):14–25, 2018.
- Ying-Hong Chan and Yao-Chung Fan. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, 2019.
- Ho-Lam Chung, Ying-Hong Chan, and Yao-Chung Fan. A bert-based distractor generation scheme with multi-tasking and negative answer training strategies. *arXiv preprint arXiv:2010.05384*, 2020.

- Rui Correia, Jorge Baptista, Maxine Eskenazi, and Nuno Mamede. Automatic generation of cloze question stems. In *International Conference on Computational Processing of the Portuguese Language*, pages 168–178. Springer, 2012.
- Rui Pedro dos Santos Correia, Jorge Baptista, Nuno Mamede, Isabel Trancoso, and Maxine Eskenazi. Automatic generation of cloze question distractors. In *Second language studies: acquisition, learning, education and technology*, 2010.
- Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.
- Xinya Du, Junru Shao, and Claire Cardie. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*, 2017.
- Jacob Eisenstein. Natural language processing. 2018.
- Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.
- João Ferreira, Ricardo Rodrigues, and Hugo Gonçalo Oliveira. Assessing factoid question-answer generation for portuguese (short paper). In *9th Symposium on Languages, Applications and Technologies (SLATE 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.
- Michael Flor and Brian Riordan. A semantic role-based approach to open-domain automatic question generation. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 254–263, 2018.
- Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.
- Hafedh Hussein, Mohammed Elmogy, and Shawkat Guirguis. Automatic english question generation system based on template driven scheme. *International Journal of Computer Science Issues (IJCSI)*, 11(6):45, 2014.
- Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall series in Artificial Intelligence. Prentice Hall, Pearson Education International, Englewood Cliffs, NJ, 2nd edition, 2009.
- Kettip Kriangchaivech and Artit Wangperawong. Question generation by transformers. *arXiv preprint arXiv:1909.05017*, 2019.

- Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204, 2020.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.
- Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195, 2015.
- Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL <https://aclanthology.org/W04-1013>.
- Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and Charibeth Cheng. Transformer-based end-to-end question generation. *arXiv preprint arXiv:2005.01107*, 4, 2020.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proc of Workshop track of ICLR*, 2013b.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013c.
- Ruslan Mitkov, Ha Le An, and Nikiforos Karamanis. A computer-aided environment for generating multiple-choice test items. *Natural language engineering*, 12(2):177–194, 2006.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Procs of 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*, pages 1532–1543. ACL, 2014.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

- Manuel Romero. T5 (base) fine-tuned on squad for qg via ap. <https://huggingface.co/mrm8488/t5-base-finetuned-question-generation-ap>, 2021.
- Katherine Stasaski and Marti A Hearst. Multiple choice question generation utilizing an ontology. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 303–312, 2017.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- Bingning Wang, Xiaochuan Wang, Ting Tao, Qi Zhang, and Jingfang Xu. Neural question generation with answer pivot. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9138–9145, 2020.
- Zhibiao Wu and Martha Palmer. Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*, 1994.
- Xuchen Yao, Gosse Bouma, and Yi Zhang. Semantics-based question generation and implementation. *Dialogue & Discourse*, 3(2):11–42, 2012.
- Cheng Zhang, Yicheng Sun, Hejia Chen, and Jie Wang. Generating adequate distractors for multiple-choice questions. *arXiv preprint arXiv:2010.12658*, 2020.
- Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer, 2017.



# Appendices



# Appendix A

## Question Generation Rules

- SV or SVA:
  - Answer in subject:
    - \* pronoun + verb + adverbials + ?
  - Answer in adverbial:
    - \* pronoun + verb + subject + ?
    - \* pronoun + verb0 + subject + verb1 + ?
    - \* pronoun + aux + subject + verb + ?
- SVO or SVOA:
  - Answer in subject:
    - \* pronoun + verb + direct object + adverbials + ?
  - Answer in direct object:
    - \* pronoun + verb + subject + adverbials + ?
    - \* pronoun + verb0 + subject + verb1 + adverbials + ?
    - \* pronoun + aux + subject + verb + adverbials + ?
  - Answer in adverbial:
    - \* pronoun + verb + subject + direct object + ?
    - \* pronoun + verb0 + subject + verb1 + direct object + ?
    - \* pronoun + aux + subject + verb + direct object + ?
- SVC or SVCA:
  - Answer in subject:
    - \* pronoun + verb + complement + adverbials + ?
  - Answer in complement:
    - \* pronoun + verb + subject + adverbials + ?
    - \* pronoun + verb0 + subject + verb1 + adverbials + ?
    - \* pronoun + aux + subject + verb + adverbials + ?

- Answer in adverbial:
  - \* pronoun + verb + subject + complement + ?
  - \* pronoun + verb0 + subject + verb1 + complement + ?
  - \* pronoun + aux + subject + verb + complement + ?
- SVOC or SVOCA:
  - Answer in subject:
    - \* pronoun + verb + direct object + complement + adverbials + ?
  - Answer in direct object:
    - \* pronoun + verb + subject + complement + adverbials + ?
    - \* pronoun + verb0 + subject + verb1 + complement + adverbials + ?
    - \* pronoun + aux + subject + verb + complement + adverbials + ?
  - Answer in complement:
    - \* pronoun + verb + subject + direct object + adverbials + ?
    - \* pronoun + verb0 + subject + verb1 + direct object + adverbials + ?
    - \* pronoun + aux + subject + verb + direct object + adverbials + ?
  - Answer in adverbial:
    - \* pronoun + verb + subject + direct object + complement + ?
    - \* pronoun + verb0 + subject + verb1 + direct object + complement + ?
    - \* pronoun + aux + subject + verb + direct object + complement + ?
- SVOO or SVOOA:
  - Answer in subject:
    - \* pronoun + verb + direct object + indirect object + adverbials + ?
  - Answer in direct object:
    - \* pronoun + verb + subject + indirect object + adverbials + ?
    - \* pronoun + verb0 + subject + verb1 + indirect object + adverbials + ?
    - \* pronoun + aux + subject + verb + indirect object + adverbials + ?
  - Answer in indirect object:
    - \* pronoun + verb + subject + direct object + adverbials + ?
    - \* pronoun + verb0 + subject + verb1 + direct object + adverbials + ?
    - \* pronoun + aux + subject + verb + direct object + adverbials + ?
  - Answer in adverbial:
    - \* pronoun + verb + subject + direct object + indirect object + ?
    - \* pronoun + verb0 + subject + verb1 + direct object + indirect object + ?
    - \* pronoun + aux + subject + verb + direct object + indirect object + ?

# Appendix B

## Evaluation Forms

These forms had the objective of evaluating the quality of the questions for the English language automatically generated from various methods.

The sources were articles from the English version of Wikipedia, more concretely the first three sentences of each article. To these sentences was applied coreference resolution. Based on the text presented for each article, people had to answer questions related to each of the groups of generated questions presented.

Here we show prints of the forms used to collect data. The questions from B.1 were used to evaluate each of the groups of generated questions represented below. In total, there were five forms for Question Generation, each for a different article (Figures B.2 to B.26).

We also conducted an evaluation for Distractor Selection Methods. In a single form (Figures B.27 to B.37), we presented the distractors generated by all the methods for each question (without repetition of options).

Sobre as Perguntas Geradas \*

	Discordo totalmente	Discordo	Concordo	Concordo totalmente
As perguntas apresentam qualidade suficiente para serem incluídas num questionário sem ser necessária grande edição	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O conjunto de perguntas tem uma boa cobertura a nível das perguntas possíveis de fazer relativamente ao texto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
As perguntas apresentadas são um bom ponto de partida para a criação de um questionário dentro deste tema	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure B.1: Questions to evaluate the performance of Question Generation approaches

## B.1 Coimbra

### Avaliação Geração Automática de Perguntas (Coimbra)

Este formulário tem como objetivo avaliar a qualidade de perguntas para a língua inglesa geradas automaticamente a partir de vários métodos.

A fonte foi o artigo "Coimbra" da Wikipédia em língua inglesa (<https://en.wikipedia.org/wiki/Coimbra>), mais concretamente as três primeiras frases. As essas frases foi aplicada resolução de correferências, sendo este o texto a partir do qual foram geradas as perguntas:

"Coimbra is a city and a municipality in Portugal. The population of the municipality at the 2011 census was 143,397, in an area of 319.40 square kilometres (123.3 sq mi). The second-largest urban area in Portugal outside Lisbon and Porto Metropolitan Areas after Braga, it is the largest city of the district of Coimbra and the Centro Region."

Com base no texto responda às questões apresentadas sobre cada um dos grupos de perguntas geradas.

\*Obrigatório

Figure B.2: Information about the form for the article "Coimbra", including text from which questions were generated

Grupo de Perguntas Geradas

Pergunta: Where is a city and a municipality in Portugal ?  
Resposta: Coimbra

Pergunta: Where is Coimbra ?  
Resposta: Portugal

Pergunta: Population was 143,397, in an area of 319.40 square kilometres (123.3 sq mi)?  
Resposta: 2011

Pergunta: How ##th was The population of the municipality at the 2011 census in an area of 319.40 square kilometres (123.3 sq mi)?  
Resposta: 143,397

Pergunta: How much was The population of the municipality at the 2011 census 143,397,?  
Resposta: 319.40 square kilometres

Pergunta: How much was The population of the municipality at the 2011 census 143,397,?  
Resposta: 123.3 sq mi

Pergunta: What is it ?  
Resposta: the Centro Region

Figure B.3: Questions generated using NER for Answer Selection and rules for Question Generation for the article "Coimbra"

Grupo de Perguntas Geradas

Pergunta: What is the name of the city in Portugal?  
Resposta: Coimbra

Pergunta: In what country is Coimbra located?  
Resposta: Portugal

Pergunta: What year was the population of the municipality at its peak?  
Resposta: 2011

Pergunta: What was the population of the municipality in 2011?  
Resposta: 143,397

Pergunta: What was the population of the municipality in 2011?  
Resposta: 319.40 square kilometres

Pergunta: What is the area of the municipality?  
Resposta:123.3 sq mi

Pergunta: Where does Lisbon rank in terms of the size of the urban area?  
Resposta: second

Pergunta: What country is Lisbon in?  
Resposta: Portugal

Pergunta: What city is the second largest in Portugal?  
Resposta: Lisbon

Pergunta: What is the largest urban area in Portugal?  
Resposta: Porto Metropolitan Areas

Pergunta: What is the second largest urban area in Portugal?  
Resposta: Braga

Pergunta: What is the largest city in the district?  
Resposta: Coimbra

Pergunta: Along with Coimbra, what region is Porto the largest city in?  
Resposta: the Centro Region

Figure B.4: Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Coimbra"

Grupo de Perguntas Geradas

Pergunta: Where is a city and a municipality in Portugal ?  
Resposta: Coimbra

Pergunta: Where is Coimbra ?  
Resposta: Portugal

Pergunta: How ##th was The population of the municipality at the 2011 census in an area of 319.40 square kilometres (123.3 sq mi)?  
Resposta: 143,397

Pergunta: What is it ?  
Resposta: the largest city of the district of Coimbra and the Centro Region

Figure B.5: Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Coimbra"

Grupo de Perguntas Geradas

Pergunta: What is the name of the city in Portugal?  
Resposta: Coimbra

Pergunta: In what country is Coimbra located?  
Resposta: Portugal

Pergunta: What was the population of the municipality in 2011?  
Resposta: 143,397

Pergunta: What is Lisbon's largest city?  
Resposta: the largest city of the district of Coimbra and the Centro Region

Figure B.6: Questions generated using Transformers for both Answer Selection and Question Generation for the article "Coimbra"

## B.2 Cristiano Ronaldo

**Avaliação Geração Automática de Perguntas (Cristiano Ronaldo)**

Este formulário tem como objetivo avaliar a qualidade de perguntas para a língua inglesa geradas automaticamente a partir de vários métodos.

A fonte foi o artigo "Cristiano Ronaldo" da Wikipédia em língua inglesa ([https://en.wikipedia.org/wiki/Cristiano\\_Ronaldo](https://en.wikipedia.org/wiki/Cristiano_Ronaldo)), mais concretamente as três primeiras frases. As essas frases foi aplicada resolução de correferências, sendo este o texto a partir do qual foram geradas as perguntas:

"Cristiano Ronaldo dos Santos Aveiro (born 5 February 1985) is a Portuguese professional footballer who plays as a forward for Premier League club Manchester United and captains the Portugal national team. Often considered the best player in the world and widely regarded as one of the greatest players of all time, Cristiano Ronaldo has won five Ballon d'Or awards and four European Golden Shoes, the most by a European player. Cristiano Ronaldo has won 32 trophies in Cristiano Ronaldo career, including seven league titles, five UEFA Champions Leagues, and the UEFA European Championship."

Com base no texto responda às questões apresentadas sobre cada um dos grupos de perguntas geradas.

\*Obrigatório

Figure B.7: Information about the form for the article "Cristiano Ronaldo", including text from which questions were generated



Grupo de Perguntas Geradas	
Pergunta: He is a Portuguese professional footballer who plays as a forward for Premier League club Manchester United and captains the Portugal national team ?	Resposta: Cristiano Ronaldo dos Santos Aveiro
Pergunta: He is a Portuguese professional footballer who plays as a forward for Premier League club Manchester United and captains the Portugal national team ?	Resposta: Santos Aveiro
Pergunta: He is a Portuguese professional footballer who plays as a forward for Premier League club Manchester United and captains the Portugal national team ?	Resposta: 5 February 1985
Pergunta: What is Cristiano Ronaldo dos Santos Aveiro (born 5 February 1985) ?	Resposta: Portuguese
Pergunta: Who is Cristiano Ronaldo dos Santos Aveiro (born 5 February 1985) ?	Resposta: Premier League
Pergunta: Who does who play?	Resposta: Premier League
Pergunta: Who is Cristiano Ronaldo dos Santos Aveiro (born 5 February 1985) ?	Resposta: Manchester United
Pergunta: Who does who play?	Resposta: Manchester United
Pergunta: Where is Cristiano Ronaldo dos Santos Aveiro (born 5 February 1985) ?	Resposta: Portugal
Pergunta: Where does who play?	Resposta: Portugal
Pergunta: Who considered the best player in the world and Often?	Resposta: Cristiano Ronaldo
Pergunta: Who regarded widely as one of the greatest players of all time?	Resposta: Cristiano Ronaldo
Pergunta: He has won five Ballon d'Or awards and four European Golden Shoes, the most by a European player ?	Resposta: Cristiano Ronaldo
Pergunta: How much has Cristiano Ronaldo won ?	Resposta: five
Pergunta: Who has Cristiano Ronaldo won ?	Resposta: Ballon d'Or
Pergunta: How much has Cristiano Ronaldo won ?	Resposta: four
Pergunta: What has Cristiano Ronaldo won ?	Resposta: European
Pergunta: What has Cristiano Ronaldo won ?	Resposta: European
Pergunta: He has won 32 trophies in Cristiano Ronaldo career, including seven league titles, five UEFA Champions Leagues, and the UEFA European Championship?	Resposta: Cristiano Ronaldo
Pergunta: How much has Cristiano Ronaldo won in Cristiano Ronaldo career, including seven league titles, five UEFA Champions Leagues, and the UEFA European Championship?	Resposta: 32
Pergunta: He has won 32 trophies in Cristiano Ronaldo career, including seven league titles, five UEFA Champions Leagues, and the UEFA European Championship?	Resposta: Cristiano Ronaldo
Pergunta: How much has Cristiano Ronaldo won 32 trophies?	Resposta: seven
Pergunta: How much has Cristiano Ronaldo won 32 trophies?	Resposta: five
Pergunta: Who has Cristiano Ronaldo won 32 trophies?	Resposta: UEFA Champions Leagues
Pergunta: Who has Cristiano Ronaldo won 32 trophies?	Resposta: the UEFA European Championship

Figure B.8: Questions generated using NER for Answer Selection and rules for Question Generation for the article "Cristiano Ronaldo"

Grupo de Perguntas Geradas	
Pergunta: Who is the Portuguese professional footballer?	Resposta: Cristiano Ronaldo dos
Pergunta: Who is Cristiano Ronaldo dos Santos?	Resposta: Santos Aveiro
Pergunta: When was Cristiano Ronaldo born?	Resposta: 5 February 1985
Pergunta: What nationality is Cristiano Ronaldo?	Resposta: Portuguese
Pergunta: What league is Manchester United in?	Resposta: Premier League
Pergunta: What Premier League club does Cristiano Ronaldo play for?	Resposta: Manchester United
Pergunta: What country does Cristiano Ronaldo captain?	Resposta: Portugal
Pergunta: Who has won the most Golden Shoes by a European player?	Resposta: Cristiano Ronaldo
Pergunta: How many Ballon d'Or awards has Cristiano Ronaldo won?	Resposta: five
Pergunta: What award has Cristiano Ronaldo won?	Resposta: Ballon d'Or
Pergunta: How many European Golden Shoes has Cristiano Ronaldo won?	Resposta: four
Pergunta: What is the most Golden Shoes won by a player?	Resposta: European
Pergunta: What is the most Golden Shoes won by a player?	Resposta: European
Pergunta: Who has won 32 trophies in his career?	Resposta: Cristiano Ronaldo
Pergunta: How many trophies has Cristiano Ronaldo won?	Resposta: 32
Pergunta: Who has won 32 trophies in his career?	Resposta: Cristiano Ronaldo
Pergunta: How many league titles has Ronaldo won?	Resposta: seven
Pergunta: How many UEFA Champions League titles has Ronaldo won?	Resposta: five
Pergunta: What is the name of the league Ronaldo has won five times?	Resposta: UEFA Champions Leagues
Pergunta: What is the name of the competition Ronaldo has won?	Resposta: the UEFA European Championship

Figure B.9: Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Cristiano Ronaldo"

Grupo de Perguntas Geradas

Pergunta: He is a Portuguese professional footballer who plays as a forward for Premier League club Manchester United and captains the Portugal national team ?  
Resposta: Cristiano Ronaldo dos Santos Aveiro

Pergunta: How much has Cristiano Ronaldo won ?  
Resposta: five

Pergunta: How much has Cristiano Ronaldo won ?  
Resposta: four

Pergunta: How much has Cristiano Ronaldo won in Cristiano Ronaldo career, including seven league titles, five UEFA Champions Leagues, and the UEFA European Championship?  
Resposta: 32

Pergunta: seven  
Resposta: How much has Cristiano Ronaldo won 32 trophies?

Figure B.10: Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Cristiano Ronaldo"

Grupo de Perguntas Geradas

Pergunta: Who is the Portuguese professional footballer?  
Resposta: Cristiano Ronaldo dos Santos Aveiro

Pergunta: Who is the Portuguese professional footballer?  
Resposta: Cristiano Ronaldo dos Santos Aveiro

Pergunta: How many Ballon d'Or awards has Cristiano Ronaldo won?  
Resposta: five

Pergunta: How many European Golden Shoes has Cristiano Ronaldo won?  
Resposta: four

Pergunta: How many trophies has Cristiano Ronaldo won?  
Resposta: 32

Pergunta: How many league titles has Ronaldo won?  
Resposta: seven

Figure B.11: Questions generated using Transformers for both Answer Selection and Question Generation for the article "Cristiano Ronaldo"

## B.3 Europe

### Avaliação Geração Automática de Perguntas (Europe)

Este formulário tem como objetivo avaliar a qualidade de perguntas para a língua inglesa geradas automaticamente a partir de vários métodos.

A fonte foi o artigo "Europe" da Wikipédia em língua inglesa (<https://en.wikipedia.org/wiki/Europe>), mais concretamente as três primeiras frases. As essas frases foi aplicada resolução de correferências, sendo este o texto a partir do qual foram geradas as perguntas:

"Europe is a continent, also recognised as a part of Eurasia, located entirely in the Northern Hemisphere and mostly in the Eastern Hemisphere. Comprising the westernmost peninsulas of Eurasia, Europe shares the continental landmass of Afro-Eurasia with both Asia and Africa. Europe is bordered by the Arctic Ocean to the north, the Atlantic Ocean to the west, the Mediterranean Sea to the south and Asia to the east."

Com base no texto responda às questões apresentadas sobre cada um dos grupos de perguntas geradas.

\*Obrigatório

Figure B.12: Information about the form for the article "Europe", including text from which questions were generated

Grupo de Perguntas Geradas

Pergunta: What is a continent ?  
Resposta: Europe

Pergunta: What does Europe comprise ?  
Resposta: Eurasia

Pergunta: What does Europe share with both Asia and Africa?  
Resposta: Eurasia

Pergunta: Where Comprising the westernmost peninsulas of Eurasia ?  
Resposta: Europe

Pergunta: Where shares the continental landmass of Afro-Eurasia with both Asia and Africa?  
Resposta: Europe

Pergunta: What does Europe share with both Asia and Africa?  
Resposta: Afro-Eurasia

Pergunta: Why does Europe share the continental landmass of Afro-Eurasia?  
Resposta: Asia

Pergunta: Why does Europe share the continental landmass of Afro-Eurasia?  
Resposta: Africa

Pergunta: It is bordered by the Arctic Ocean to the north to the east?  
Resposta: Europe

Pergunta: It is bordered by the Arctic Ocean to the north to the east?  
Resposta: the Arctic Ocean

Pergunta: It is bordered by the Arctic Ocean to the north to the east?  
Resposta: the Atlantic Ocean

Pergunta: It is bordered by the Arctic Ocean to the north to the east?  
Resposta: the Mediterranean Sea

Pergunta: It is bordered by the Arctic Ocean to the north to the east?  
Resposta: Asia

Figure B.13: Questions generated using NER for Answer Selection and rules for Question Generation for the article "Europe"

Grupo de Perguntas Geradas

Pergunta: What continent is located in the Northern Hemisphere?  
Resposta: Europe

Pergunta: Europe is a part of what continent?  
Resposta: Eurasia

Pergunta: Where is Europe located?  
Resposta: the Northern Hemisphere

Pergunta: Where is Europe located?  
Resposta: the Eastern Hemisphere

Pergunta: What continent is Europe located in?  
Resposta: Eurasia

Pergunta: What continent shares landmass with Asia and Africa?  
Resposta: Europe

Pergunta: What continent does Europe share with Asia and Africa?  
Resposta: Afro-Eurasia

Pergunta: Along with Africa, what continent does Europe share landmass with?  
Resposta: Asia

Pergunta: Along with Asia, what continent does Europe share landmass with?  
Resposta: Africa

Pergunta: What continent borders the Arctic Ocean to the north?  
Resposta: Europe

Pergunta: What ocean borders Europe to the north?  
Resposta: the Arctic Ocean

Pergunta: What ocean borders Europe to the west?  
Resposta: the Atlantic Ocean

Pergunta: What is the southern border of Europe?  
Resposta: the Mediterranean Sea

Pergunta: What continent borders Europe to the east?  
Resposta: Asia

Figure B.14: Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Europe"

Grupo de Perguntas Geradas

Pergunta: What does Europe share with both Asia and Africa?  
Resposta: Afro-Eurasia

Pergunta: Why does Europe share the continental landmass of Afro-Eurasia?  
Resposta: Asia and Africa

Pergunta: It is bordered by the Arctic Ocean to the north to the east?  
Resposta: the Arctic Ocean

Figure B.15: Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Europe"

Grupo de Perguntas Geradas

Pergunta: Europe is a part of what continent?  
Resposta: Eurasia

Pergunta: What continent does Europe share with Asia and Africa?  
Resposta: Afro-Eurasia

Pergunta: What two continents does Europe share landmass with?  
Resposta: Asia and Africa

Pergunta: What ocean borders Europe to the north?  
Resposta: the Arctic Ocean

Figure B.16: Questions generated using Transformers for both Answer Selection and Question Generation for the article "Europe"

## B.4 Queen

**Avaliação Geração Automática de Perguntas (Queen (band))**

Este formulário tem como objetivo avaliar a qualidade de perguntas para a língua inglesa geradas automaticamente a partir de vários métodos.

A fonte foi o artigo "Queen (band)" da Wikipédia em língua inglesa ([https://en.wikipedia.org/wiki/Queen\\_\(band\)](https://en.wikipedia.org/wiki/Queen_(band))), mais concretamente as três primeiras frases. As essas frases foi aplicada resolução de correferências, sendo este o texto a partir do qual foram geradas as perguntas:

"Queen are a British rock band formed in London in 1970. a British rock band formed in London in 1970 comprised Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass). a British rock band formed in London in 1970 earliest works were influenced by progressive rock, hard rock and heavy metal, but a British rock band formed in London in 1970 gradually ventured into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock."

Com base no texto responda às questões apresentadas sobre cada um dos grupos de perguntas geradas.

\*Obrigatório

Figure B.17: Information about the form for the article "Queen (band)", including text from which questions were generated

Grupo de Perguntas Geradas	
Pergunta: You are Queen ?	Resposta: British
Pergunta: You are Queen ?	Resposta: London
Pergunta: You are Queen ?	Resposta: 1970
Pergunta: They comprised Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass) ?	Resposta: British
Pergunta: They comprised Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass) ?	Resposta: London
Pergunta: They comprised Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass) ?	Resposta: 1970
Pergunta: Who did a British rock band formed in London in 1970 comprise ?	Resposta: Freddie Mercury
Pergunta: Who did a British rock band formed in London in 1970 comprise ?	Resposta: Brian May
Pergunta: Who did a British rock band formed in London in 1970 comprise ?	Resposta: Roger Taylor
Pergunta: Who did a British rock band formed in London in 1970 comprise ?	Resposta: John Deacon
Pergunta: They were influenced by progressive rock, hard rock and heavy metal?	Resposta: British
Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?	Resposta: British
Pergunta: They were influenced by progressive rock, hard rock and heavy metal?	Resposta: London
Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?	Resposta: London
Pergunta: They were influenced by progressive rock, hard rock and heavy metal?	Resposta: 1970
Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?	Resposta: 1970
Pergunta: They were influenced by progressive rock, hard rock and heavy metal?	Resposta: British
Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?	Resposta: British
Pergunta: They were influenced by progressive rock, hard rock and heavy metal?	Resposta: London
Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?	Resposta: London
Pergunta: They were influenced by progressive rock, hard rock and heavy metal?	Resposta: 1970
Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?	Resposta: 1970

Figure B.18: Questions generated using NER for Answer Selection and rules for Question Generation for the article "Queen (band)"

Grupo de Perguntas Geradas

Pergunta: Queen are a rock band from what country?  
Resposta: British

Pergunta: Where was Queen formed?  
Resposta: London

Pergunta: When was Queen formed?  
Resposta: 1970

Pergunta: What nationality was Freddie Mercury?  
Resposta: British

Pergunta: Where was the band formed?  
Resposta: London

Pergunta: When was the band formed?  
Resposta: 1970

Pergunta: Who was the lead singer of the band?  
Resposta: Freddie Mercury

Pergunta: Who was the guitarist of Freddie Mercury?  
Resposta: Brian May

Pergunta: Who played the drums for the band?  
Resposta: Roger Taylor

Pergunta: Who played the bass for the band?  
Resposta: John Deacon

Pergunta: What nationality was the band formed in London in 1970?  
Resposta: British

Pergunta: Where did the British rock band form in 1970?  
Resposta: London

Pergunta: When did a British rock band form in London?  
Resposta: 1970

Pergunta: What nationality was the band formed in London in 1970?  
Resposta: British

Pergunta: Where did the British rock band form in 1970?  
Resposta: London

Pergunta: When did a British rock band form in London?  
Resposta: 1970

Figure B.19: Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Queen (band)"

Grupo de Perguntas Geradas

Pergunta: You are Queen ?  
Resposta: 1970

Pergunta: They comprised Freddie Mercury (lead vocals, piano), Brian May (guitar, vocals), Roger Taylor (drums, vocals) and John Deacon (bass) ?  
Resposta: 1970

Pergunta: They were influenced by progressive rock, hard rock and heavy metal?  
Resposta: 1970

Pergunta: He ventured gradually into more conventional and radio-friendly works by incorporating further styles, such as arena rock and pop rock?  
Resposta: 1970

Figure B.20: Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Queen (band)"



Grupo de Perguntas Geradas

Pergunta: When was Queen formed?  
Resposta: 1970

Pergunta: When was the band formed?  
Resposta: 1970

Pergunta: When did a British rock band form in London?  
Resposta: 1970

Figure B.21: Questions generated using Transformers for both Answer Selection and Question Generation for the article "Queen (band)"

## B.5 Star Wars

### Avaliação Geração Automática de Perguntas (Star Wars (film))

Este formulário tem como objetivo avaliar a qualidade de perguntas para a língua inglesa geradas automaticamente a partir de vários métodos.

A fonte foi o artigo "Star Wars (film)" da Wikipédia em língua inglesa ([https://en.wikipedia.org/wiki/Star\\_Wars\\_\(film\)](https://en.wikipedia.org/wiki/Star_Wars_(film))), mais concretamente as três primeiras frases. As essas frases foi aplicada resolução de correferências, sendo este o texto a partir do qual foram geradas as perguntas:

"Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) is a 1977 American epic space opera film written and directed by George Lucas, produced by Lucasfilm and distributed by 20th Century Fox. A New Hope) is the first film in the Star Wars film series and fourth chronological chapter of the "Skywalker Saga". Set "a long time ago" in a fictional universe where the galaxy is ruled by the tyrannical Galactic Empire, the story focuses on a group of freedom fighters known as the Rebel Alliance, who aim to destroy the Empire's newest weapon, the Death Star."

Com base no texto responda às questões apresentadas sobre cada um dos grupos de perguntas geradas.

**\*Obrigatório**

Figure B.22: Information about the form for the article "Star Wars (film)", including text from which questions were generated

Grupo de Perguntas Geradas	
Pergunta: What is a 1977 American epic space opera film written and directed by George Lucas, produced by Lucasfilm and distributed by 20th Century Fox ?	Resposta: Star Wars: Episode IV – A New Hope
Pergunta: What is Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) ?	Resposta: 1977
Pergunta: What is Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) ?	Resposta: American
Pergunta: Who is Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) ?	Resposta: George Lucas
Pergunta: Who is Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) ?	Resposta: Lucasfilm
Pergunta: When is Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) ?	Resposta: 20th Century
Pergunta: ( is A New Hope) ?	Resposta: first
Pergunta: ( is A New Hope) ?	Resposta: fourth
Pergunta: ( is A New Hope) ?	Resposta: the "Skywalker Saga"
Pergunta: Where does the story set?	Resposta: Galactic Empire
Pergunta: How is the galaxy ruled?	Resposta: Galactic Empire
Pergunta: Where does the story focus?	Resposta: the Rebel Alliance
Pergunta: Who does the story set?	Resposta: Empire
Pergunta: Who is the galaxy ruled?	Resposta: Empire
Pergunta: Who does the story focus?	Resposta: Empire
Pergunta: Who does who aim ?	Resposta: Empire
Pergunta: Where does the story focus?	Resposta: the Death Star
Pergunta: Where does who aim ?	Resposta: the Death Star

Figure B.23: Questions generated using NER for Answer Selection and rules for Question Generation for the article "Star Wars (film)"

Grupo de Perguntas Geradas

Pergunta: What was the retroactive name of the 1977 Star Wars film?  
Resposta: Star Wars: Episode IV – A New Hope

Pergunta: When was Star Wars first released?  
Resposta: 1977

Pergunta: What nationality is Star Wars?  
Resposta: American

Pergunta: Who directed the 1977 Star Wars film?  
Resposta: George Lucas

Pergunta: Who produced the film Star Wars?  
Resposta: Lucasfilm

Pergunta: What company distributed Star Wars?  
Resposta: 20th Century

Pergunta: What is the first film in the Star Wars series?  
Resposta: first

Pergunta: What is the rank of the fourth chapter of the Skywalker Saga?  
Resposta: fourth

Pergunta: What is the fourth chapter of the Star Wars series?  
Resposta: the "Skywalker Saga"

Pergunta: Who is the galaxy ruled by?  
Resposta: Galactic Empire

Pergunta: What group of freedom fighters is the story about?  
Resposta: the Rebel Alliance

Pergunta: Who is the Galactic Empire ruled by?  
Resposta: Empire

Pergunta: What is the name of the weapon the Rebel Alliance is trying to destroy?  
Resposta: the Death Star

Figure B.24: Questions generated using NER for Answer Selection and a Transformer for Question Generation for the article "Star Wars (film)"

Grupo de Perguntas Geradas

Pergunta: What is a 1977 American epic space opera film written and directed by George Lucas, produced by Lucasfilm and distributed by 20th Century Fox ?  
Resposta: Star Wars: Episode IV – A New Hope

Pergunta: Who is Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) ?  
Resposta: George Lucas

Pergunta: It is the first film in the Star Wars film series and fourth chronological chapter of the "Skywalker Saga" ?  
Resposta: A New Hope

Pergunta: Where does the story focus?  
Resposta: the Rebel Alliance

Pergunta: Where does the story set?  
Resposta: the tyrannical Galactic Empire

Pergunta: How is the galaxy ruled?  
Resposta: the tyrannical Galactic Empire

Figure B.25: Questions generated using a Transformer for Answer Selection and rules for Question Generation for the article "Star Wars (film)"

Grupo de Perguntas Geradas

Pergunta: What was the retroactive name of the 1977 Star Wars film?  
Resposta: Star Wars: Episode IV – A New Hope

Pergunta: Who directed the 1977 Star Wars film?  
Resposta: George Lucas

Pergunta: What is the name of the fourth chapter of the Skywalker Saga?  
Resposta: A New Hope

Pergunta: What group of freedom fighters is the story about?  
Resposta: the Rebel Alliance

Pergunta: Who controls the galaxy?  
Resposta: the tyrannical Galactic Empire

Figure B.26: Questions generated using Transformers for both Answer Selection and Question Generation for the article "Star Wars (film)"

## B.6 Distractors

**Geração Respostas Incorretas para Perguntas de Escolha Múltipla**

Este formulário tem como objetivo avaliar a qualidade de opções incorretas para perguntas de escolha múltipla em inglês geradas automaticamente a partir de vários métodos.

Com base em vários artigos da Wikipedia, após aplicar resolução de correferências, obtiveram-se vários pares de perguntas e respostas. Para cada um desses pares, foram geradas potenciais respostas incorretas.

Pretende-se que, com base na frase a partir da qual a pergunta foi gerada, a pergunta em si e a resposta correta, indique quais os conjuntos de respostas incorretas que considera adequados.

**\*Obrigatório**

Figure B.27: Information about the form to evaluate distractors

1/5 Artigo: Coimbra (<https://en.wikipedia.org/wiki/Coimbra>)

Frase: Coimbra is a city and a municipality in Portugal.  
Pergunta: In what country is Coimbra located?  
Resposta: Portugal

Potenciais respostas incorretas: \*

- Lisbon
- Algarve
- Brazil
- France
- Porto
- Spain
- Greece
- Portuguese
- Italy
- Argentina
- Poland
- Coimbra
- Portuguese Province
- Portuguese City

Figure B.28: Distractors selected for a question generated from the article "Coimbra". Answer NE label: GPE

Frase: About 460,000 people live in Coimbra, comprising 19 municipalities and extending into an area of 4,336 square kilometres (1,674 sq mi).  
Pergunta: What is the area of Coimbra?  
Resposta: 4,336 square kilometres

Potenciais respostas incorretas: \*

- 73 meters
- 920 metres
- 1,674 sq mi
- 10 hectares
- 3,018 feet
- metre
- railway yard
- plot of land
- plot of ground
- picnic area
- 4,336 square kilometers
- 4,336 square miles
- 4,336 square 160
- 3,000 square kilometers
- 3,400 square kilometers
- 3,732 square kilometers
- 3,732 kilometers

Figure B.29: Distractors selected for a question generated from the article "Coimbra". Answer NE label: QUANTITY

2/5 Artigo: Star Wars (film) ([https://en.wikipedia.org/wiki/Star\\_Wars\\_\(film\)](https://en.wikipedia.org/wiki/Star_Wars_(film)))

Frase: Star Wars (retroactively titled Star Wars: Episode IV – A New Hope) is a 1977 American epic space opera film written and directed by George Lucas, produced by Lucasfilm and distributed by 20th Century Fox.  
Pergunta: Who directed the 1977 Star Wars film?  
Resposta: George Lucas

Potenciais respostas incorretas: \*

- Peter Jackson
- Michael Moore
- w. lucas
- The Star Wars
- The Star Wars Star
- Star Wars
- Mark Hamill
- william lucas
- charles lucas
- howard lucas
- john lucas
- Peter Cooper
- Robert E. Howard
- Jason Scott
- James Taylor
- John Williams
- Martin Smith
- Charles
- John Simon
- John Carter

Figure B.30: Distractors selected for a question generated from the article "Star Wars (film)". Answer NE label: PERSON

Frase: When adjusted for inflation, Star Wars is the second-highest-grossing film in North America (behind Gone with the Wind) and the fourth-highest-grossing film of all time.  
Pergunta: Where does Star Wars rank in grossing movies in North America?  
Resposta: second

---

Potenciais respostas incorretas: \*

- 10th
- third
- first
- Star Wars
- Star Wars first
- sixth
- fifth
- fourth
- eighth

Figure B.31: Distractors selected for a question generated from the article "Star Wars (film)". Answer NE label: ORDINAL



3/5 Artigo: Queen (band) ([https://en.wikipedia.org/wiki/Queen\\_\(band\)](https://en.wikipedia.org/wiki/Queen_(band)))

Frase: The latter featured "Bohemian Rhapsody", which stayed at number one in the UK for nine weeks and helped popularise the music video format.  
Pergunta: What song was featured in the video?  
Resposta: Bohemian Rhapsody

Potenciais respostas incorretas: \*

- Rhapsody
- beatnik
- Czech
- moravian rhapsody
- bohemia rhapsody
- Queen
- Queen in London
- British rock
- rock band
- British
- bohemian serenade
- transylvanian rhapsody
- aristocratic rhapsody
- Austrian
- Grecian
- Hungarian
- Queen + Wyclef Jean
- Queen released Jazz
- Seaside Rendezvous
- John Lennon Ealing College

Figure B.32: Distractors selected for a question generated from the article "Queen (band)". Answer NE label: WORK\_OF\_ART

Frase: Queen are a British rock band formed in London in 1970.  
Pergunta: When was Queen formed?  
Resposta: 1970

---

Potenciais respostas incorretas: \*

- 1974
- British
- Before London
- 1971
- Before 1970
- By London
- After London
- 1977
- 1969
- 1965
- 1966
- 1968
- 1972
- 1973

Figure B.33: Distractors selected for a question generated from the article "Queen (band)". Answer NE label: DATE

4/5 Artigo: Europe (<https://en.wikipedia.org/wiki/Europe>)

Frase: Europe is bordered by the Arctic Ocean to the north, the Atlantic Ocean to the west, the Mediterranean Sea to the south and Asia to the east.  
Pergunta: What ocean borders Europe to the north?  
Resposta: the Arctic Ocean

Potenciais respostas incorretas: \*

- the Gulf Stream
- sea
- Geography of North America
- the arctic sea
- China and Europe
- China
- the Middle Pacific
- the Atlantic Ocean
- the Black Sea
- the arctic atlantic
- the arctic coast
- the arctic seas
- the arctic waters
- Environment of North America
- Flora of Northern Europe
- Shipwrecks of North Asia
- Flora of Eastern Europe
- territorial waters
- seven seas
- international waters
- high sea
- North Atlantic Drift
- the Northern Hemisphere
- the Emba River
- the Southern Hemisphere

Figure B.34: Distractors selected for a question generated from the article "Europe". Answer NE label: LOC

Frase: In 1949, the Council of Europe was founded with the idea of unifying Europe to achieve common goals and prevent future wars.  
Pergunta: What organization was founded in 1949?  
Resposta: the Council of Europe

Potenciais respostas incorretas: \*

- the Tsardom of Russia
- Organization of American States
- Government by continent
- the commission of europe
- the council of european
- Europe's umbrella of Europe
- Europe's continent
- Europe's branch of Europe
- Europeans
- the council of america
- the council of asia
- the council of countries
- Lists of continents
- Personifications of continents
- Buildings and structures by continent
- History by continent
- Commonwealth of Independent States
- League of Nations
- Organization for the Prohibition of Chemical Weapons
- United Nations agency
- the European Capital of Sport
- the European Capital of Culture
- the Union of Krewo
- the European Theatre of World War II

Figure B.35: Distractors selected for a question generated from the article "Europe". Answer NE label: ORG

4/5 Artigo: Cristiano Ronaldo ([https://en.wikipedia.org/wiki/Cristiano\\_Ronaldo](https://en.wikipedia.org/wiki/Cristiano_Ronaldo))

Frase: Cristiano Ronaldo dos Santos Aveiro (born 5 February 1985) is a Portuguese professional footballer who plays as a forward for Premier League club Manchester United and captains the Portugal national team.  
 Pergunta: What nationality is Cristiano Ronaldo?  
 Resposta: Portuguese

Potenciais respostas incorretas: \*

- Spanish
- Footballers
- English
- English and Brazilian
- English player
- Portuguese
- argentine
- italian
- portugal
- brazilian
- Romanian
- Catalan
- Norwegian
- Italian
- Dutch
- Argentine-Spanish

Figure B.36: Distractors selected for a question generated from the article "Cristiano Ronaldo". Answer NE label: NORP

Frase: Often considered the best player in the world and widely regarded as one of the greatest players of all time, Cristiano Ronaldo has won five Ballon d'Or awards and four European Golden Shoes, the most by a European player.  
 Pergunta: How many Ballon d'Or awards has Cristiano Ronaldo won?  
 Resposta: five

Potenciais respostas incorretas: \*

- six
- seven
- seven awards
- seven games
- seven cups
- three
- four
- eight

Figure B.37: Distractors selected for a question generated from the article "Cristiano Ronaldo". Answer NE label: CARDINAL