Generating and Ranking Distractors for Multiple-Choice Questions in Portuguese

Hugo Gonçalo Oliveira
Center of Informatics and Systems, University of Coimbra, Portugal
Department of Informatics Engineering, University of Coimbra, Portugal

Igor Caetano
Instituto Pedro Nunes, Coimbra, Portugal
Department of Informatics Engineering, University of Coimbra, Portugal

Renato Matos
Center of Informatics and Systems, University of Coimbra, Portugal
Department of Informatics Engineering, University of Coimbra, Portugal

Hugo Amaro
Instituto Pedro Nunes, LIS, Coimbra, Portugal

Abstract
In the process of multiple-choice question generation, different methods are often considered for distractor acquisition, as an attempt to cover as many questions as possible. Some, however, result in many candidate distractors of variable quality, while only three or four are necessary. We implement some distractor generation methods for Portuguese and propose their combination and ranking with language models. Experimentation results confirm that this increases both coverage and suitability of the selected distractors.

1 Introduction
Recent breakthroughs in Natural Language Processing (NLP) made knowledge even more accessible with tasks like Question Answering. In most cases, however, this does not mean that training and assessing humans is no longer necessary. Here, another task that benefits from NLP is Question Generation (QG) [12]. As the name suggests, QG aims at creating questions automatically (e.g., from learning materials), thus reducing the time that educators spend in the production of tests and leaving more time for activities like class preparation or interaction with students.

Due to straightforward grading, multiple-choice questions (MCQs) are a popular kind of questions. In addition to the question stem, MCQs have a list of alternative answers, out of which one is correct and the others are distractors. The creation of MCQs has also been automated [1] in a process that considers the generation of the distractors. Many distractor generation methods have been proposed, but they are rarely suitable to every type
of question, making it necessary to combine different methods. At the same time, some of the methods, or their combination, may produce a large set of candidates, while, in most cases, only three or four distractors are necessary. A selection has to be done, but the produced distractors are often of variable quality, so a random selection is rarely the best option.

We compile a set of distractor generation methods common in the literature, describe their adaptation to Portuguese, and apply them to a set of machine reading comprehension questions. To minimise the impact of random selection, we further propose a straightforward method for ranking distractors. It is based on pretrained language models, namely BERT [8] or GPT2 [23], and their application to computing the likelihood of textual sequences. A manual evaluation of results for a set of questions confirms that such models are a good option for ranking the distractors. They can be applied to distractors by different methods, thus increasing the number of covered questions, as well as the proportion of good distractors.

In the remainder of the paper, after reviewing some related work, we describe the implemented methods for generation and ranking; we report on a performed experiment and its evaluation; we conclude with final remarks and possible future directions.

2 Related Work

When generating MCQs [1], distractors have to be generated in addition to the stem of the questions. The quality of distractors has been estimated with several automatic methods, including named entities (NEs) of the same category, relatedness in WordNet, semantic types in DBPedia, or distributional semantics [21]. Not surprisingly, most of the previous methods were also applied to distractor generation.

When the answer is a word of a specific part-of-speech (PoS) or a named entity (NE) of a specific category [29], context words of the same type can be used as distractors. When the answer is a number, distractors can be obtained by increasing or decreasing it [29].

Alternatively, distractors can be retrieved from external resources, such as WordNet [9] or DBPedia [14]. From the latter, words that share a hypernym with the answer (co-hyponym) [18, 29] or that are similar enough [29] can be used. If too many distractors are obtained this way, preference can be given to those that appear in the context [18]. From DBPedia, distractors can be obtained by removing restrictions in the SPARQL query that answers the question [26]. Concepts that share properties or are related with the answers have also been obtained from other ontologies [28]. The external resource can also be a model of distributional semantics, where words similar to the answer can be obtained from [28, 11]. Other methods include using words with similar spelling [11] or masked language modelling [2]. Transformers like T5 may also be fine-tuned for generating MCQs, including the distractors [16].

In some of the previous, distractors can be ranked, according to one or more of the following features: PoS similarity [25, 2], semantic similarity with the answer [11, 2, 25], proximity of frequency [11, 25], or confidence score of a language model [2]. In any case, distractors cannot be synonyms of the answer.

Specifically for Portuguese, there is some work on QG. The majority relies on linguistic knowledge, such as syntactic dependencies [6, 22] or semantic roles [10], sometimes focusing exclusively on named entities [22]. But there is recent work with neural [15] approaches.

For distractor generation in Portuguese, words that shared traces with the answer have been used [6]. Specifically for cloze-style questions, multiple approaches were applied for distractor generation [3], which could be words with similar features (e.g., PoS, frequency), words obtained by exploring common errors in Portuguese, or related words (e.g., hyponyms and hypernyms in lexical resources). In the scope of listening comprehension, distractors were obtained from phonetically-similar words [20].
3 Approach

Several distractor generation methods were compiled from the literature and adapted to Portuguese. In order to select a subset of distractors by the previous, we rely on language models for computing their likelihood as answers to the question. This section describes the distractor generation and ranking methods.

3.1 Distractor Generation

Five distractor generation methods were implemented in this work. Due to their specificities, they do not produce distractors for every single question-answer pair. Yet, the number of covered questions can be maximised by a combination of methods.

The first method, hereafter Ctx, is the only that selects distractors from a given context and it only applies if such a context is available. If the answer is a NE, other entities of the same category that appear in the context are selected and used directly as distractors. Otherwise, context words of the same PoS of words in the answer are selected to replace the latter and result in new distractors.

Since many answers are or include numbers (e.g., ages, years, quantities), a method (Nb) was implemented for generating distractors specifically for them. They are obtained by replacing each numeric token of the answer by a range of numbers resulting from the addition or subtraction of units.

Having in mind that distractors should be semantically-similar to the correct answer, the remaining generation methods resort to three different resources for getting words of the same category. One (WN) gets co-hyponyms, i.e., words that share a hypernym, from a WordNet-like [9] lexical database.

Since wordnets cover mostly lexicographic knowledge, for world concepts, we get distractors from DBPedia [14] (DBP), an open multilingual knowledge base extracted from Wikipedia. Words that share one or more properties are good distractor candidates. In a parallelism with WN, we focus on words of the same category.

Distractors are also obtained from the most similar words (Sim), according to a word2vec-like [17] model. To avoid the inclusion of alternative correct answers, synonyms and hypernyms of the answer are removed with the help of WordNet.

If no distractors are obtained for the full answer with the previous three methods, they are applied to each open token in the answer, which is then replaced by the retrieved words and used as distractors. Possible outputs of the described methods, when implemented according to section 4, are illustrated in Tables 1 and 2.

| Table 1 Examples of context, question, answer, and distractors extracted from context. |

<table>
<thead>
<tr>
<th>Context</th>
<th>Question</th>
<th>Answer</th>
<th>Distractors</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>O caixão de Bell foi</td>
<td>Qual cantor se apres...</td>
<td>Jean MacDonald</td>
<td>Beinn Bhreagh, Robert Louis Stevenson</td>
<td>Ctx</td>
</tr>
<tr>
<td>construído com pinho</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Beinn Bhreagh... pediu aos convidados para não usarem preto (a cor</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tradicional do funeral)......</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>durante o qual o solista</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jean MacDonald cantou um verso de “Requiem” de</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Robert Louis Stevenson:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2  Examples of questions and answers in the dataset, followed by distractors generated by different methods.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
<th>Distractors</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Que peça do uniforme foi substituída pelo boné de patrilha?</td>
<td>boina preta</td>
<td>balackava preta, gorro preta, fez preta, quepe preta, ...</td>
<td>WN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>jaqueta preta, gorro preta, camisola preta, boina branca, boina vermelha, boina prateada, ...</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sim</td>
<td></td>
</tr>
<tr>
<td>Onde está localizado o templo de Walhalla?</td>
<td>Baviera</td>
<td>Berlim Leste, Renânia do Norte-Vestfália, Baden, Hamburgo, Saxônia, ...</td>
<td>DBP</td>
</tr>
<tr>
<td>Quando Victoria pediu a Palmerston que retomasse seu escritório?</td>
<td>Junho de 1859</td>
<td>Junho de 1849, Junho de 1850, ... , Junho de 1868, Junho de 1869</td>
<td>Nb</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Janeiro de 1859, Agosto de 1859, Novembro de 1859</td>
<td></td>
</tr>
<tr>
<td>Qual é a substância mais comumente abusada durante a adolescência nos EUA?</td>
<td>álcool</td>
<td>água potável, anfetamina, café, leite, tabaco, ...</td>
<td>WN</td>
</tr>
</tbody>
</table>

3.2 Distractor Ranking

Given their specificities, the five distractor generation methods will not generate distractors for all types of questions. The problem is that, in many cases, there will still be many distractors, even if only three or four are necessary. At the same time, their quality will be variable. For instance, without further polishing, distractors might include typos (e.g., titulos soberanos, agencia de jornais) or, after replacement: result in inconsistent gender / number (e.g., boina adequado); result from very generic connections (e.g., Portas Citosina or Portas Teriflunomida for Portas USB, because USB was an American Invention); be related to a different sense of the answer (e.g., Anatomia de Yongying for Corpo de Yongying); or simply result in odd mixes (e.g., Oskar New York Times). This is why, instead of just using a random sample of all the produced distractors, a method for either selecting the most promising, or for discarding problematic ones, can be useful. Here, we could opt for classifying distractors as good or bad. However, this discrimination is often subjective (see Section 4) and, even when distractors are good, they might have different “levels” of suitability. Therefore, we opt for ranking distractors and propose to use language models (LMs) in what they were originally developed for: computing the likelihood of text sequences. A sequence will consist of the question immediately followed by the answer, e.g., the first distractor in table 2 results in the following sequence: Que peça do uniforme foi substituída pelo boné de patrilha? balackava preta. For each question, a sequence like the previous is produced for each distractor, and distractors are ranked in descending order of the likelihood of their sequence. Considering that, in any case, selected distractors should be reviewed by a human, it should be easier to manually select distractors from a ranking than from a set, possibly containing dozens of options.
4 Experimentation

To test the distractor generation and ranking methods, they were applied to a selection of Portuguese questions and answers. Obtained distractors were then manually evaluated and some conclusions were taken. This section describes the data used, the implementation of the methods, and finally presents the results and their discussion.

4.1 Evaluation Data

Distractors were generated for a random selection of 124 context-question-answer tuples in the validation portion of a Portuguese translation of the SQuAD [24] dataset, produced by the Deep Learning Brasil group\(^1\). Since MCQs typically have short answers, the sample was restricted to questions of three-token answers or less. The first three columns in Table 1, context, question, answer, illustrate the entries of the dataset. The original version of SQuAD has been extensively used for training question answering and generation models and it seemed appropriate to our experimentation. In opposition to another popular dataset, RACE [13], it does not contain distractors, but, as far as we know, RACE is not available for Portuguese. In any case, it would be difficult to automatise the evaluation of generated distractors, because there are often many suitable options.

4.2 Implementation

To implement the distractor generation for Portuguese, several tools and resources were used. In the Ctx method, the context is first tagged with the spaCy\(^2\) toolkit, using the largest available model for Portuguese, `pt_core_news_lg`. This enables the identification of NEs and of the words’ PoS. Only words of open PoS were considered for replacement. The same model was used for obtaining the most similar words in the Sim method. In the Nb method, numeric tokens `nt` are identified with Python’s `isnumeric()` function. Then, all the numbers in the `[nt - 10, nt]` and `[nt, nt + 10]` intervals are generated to be used as replacements. The WN method relied on the NLTK interface to wordnet\(^3\). For Portuguese, it resorts to OpenWordNet-PT \([7]\). For DBP, DBPedia was accessed through its SPARQL endpoint\(^4\). For Portuguese, it first uses the `skos:broader` property, which links concepts with their broader categories, i.e., we get the labels of concepts that share a broader category with the answer. If no distractors are obtained, we do the same for the `dct:subject` property, which links concepts with related subjects, i.e., we retrieve the labels of concepts related to the same subjects as the answer.

For ranking distractors, three LMs were tested, all available from the HuggingFace transformers library\(^5\): BERTimbau \([27]\), both base and large, a BERT model pretrained for Portuguese; and GPORtuguese-2\(^6\), GPT2-small fine-tuned with 1GB of Portuguese text. For the BERT models, we relied on the FitBERT\(^7\) tool, also based on the transformers library. This tool relies on pre-softmax logit scores for ranking a list of options according to their suitability to replace a mask in a given masked sentence. In this case, the input

---

1 https://drive.google.com/file/d/1Q0laIlv2h2BC468MwUFxUST0Eyn7gUkn
2 https://spacy.io/
3 https://www.nltk.org/howto/wordnet.html
4 https://dbpedia.org/sparql
5 https://huggingface.co/transformers/
6 https://huggingface.co/pierreguillou/gpt2-small-portuguese
7 https://github.com/Qordobacode/fitbert
sentence was the question followed by a mask, and the options were the generated distractors. With GPT2, the likelihood of each sequence of tokens was approximated by the exponential of the loss of the model for this sequence.

4.3 Evaluation

Distractor generation methods were applied to each question of the evaluation data and their results were ranked by each language model. For evaluation purposes, at most three distractors were selected from each generation and each ranking method. When a generation method resulted in more than three distractors, their selection was random. As for ranking methods, they were applied to the set of all distractors by all the methods, before the previous selection, out of which the top-3 were selected.

Distractors resulting from the previous process were then shuffled for manual evaluation, which was done by two judges, one expert in Natural Language Processing and a Data Science student. Given the context, the question, the correct answer, and list of distractors, judges were asked to classify each distractor as: (0) unsuitable, i.e., nonsense or a synonym of the answer; (1) close, but a minor edition is needed, e.g., changing the gender, number or tense of a word; (2) suitable. Both judges were aware of the distractor generation methods but, during the evaluation process, did not have access to the source of each distractor. In order to compute agreement, distractors for the first 25 questions (230) were evaluated by both judges. Considering the three classes, Cohen’s \( \kappa \) was 0.61 (substantial agreement), which increased to 0.77 when the unsuitable (0) and close (1) classes were merged.

With the distractors classified, we observed the coverage of each method, as well as on the proportion of suitable distractors generated. The coverage of each method approximates the proportion of distractors of the target type generated for each question, considering a maximum of three per question, and is given by the total number distractors of the type divided by the times the number of questions. Table 3 summarizes these results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Coverage</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctx</td>
<td>42.2%</td>
<td>33.8%</td>
<td>16.7%</td>
<td>49.7%</td>
</tr>
<tr>
<td>Nb</td>
<td>21.8%</td>
<td>3.7%</td>
<td>3.7%</td>
<td>93.0%</td>
</tr>
<tr>
<td>WN</td>
<td>39.2%</td>
<td>24.7%</td>
<td>11.0%</td>
<td>64.4%</td>
</tr>
<tr>
<td>DBP</td>
<td>25.0%</td>
<td>19.4%</td>
<td>7.5%</td>
<td>73.1%</td>
</tr>
<tr>
<td>Sim</td>
<td>54.0%</td>
<td>43.4%</td>
<td>6.8%</td>
<td>49.7%</td>
</tr>
<tr>
<td>GPT2</td>
<td>96.0%</td>
<td>29.7%</td>
<td>13.3%</td>
<td>56.9%</td>
</tr>
<tr>
<td>BERT-base</td>
<td>96.0%</td>
<td>19.4%</td>
<td>8.4%</td>
<td>72.2%</td>
</tr>
<tr>
<td>BERT-large</td>
<td>96.0%</td>
<td>18.9%</td>
<td>6.7%</td>
<td>74.4%</td>
</tr>
</tbody>
</table>

Despite varying across methods, there is a significant proportion of unsuitable distractors with all methods but Nb. This is also the method with the greatest proportion of suitable distractors, followed by DBP, but, even if sometimes by a low margin, all provide at least around 50% suitable distractors. It is easy to generate distractors for numbers. With the current simplistic method, some situations could go wrong (e.g., negative quantities), but they were a minority in the evaluation sample. However, such questions account for only one fifth of the sample, and other methods must be used for the remaining questions.

---

8 For the shared 25 questions, only the classifications of the first judge were considered.
Looking at the coverage, we confirm that no method applies to a large proportion of questions. Greatest coverages are by Sim and Ctx, but these are also the least accurate methods. With Sim, there is not much control on the obtained words, which are sometimes plurals of the answer, or words of the same family. As for Ctx, we checked that the majority of issues did not result from complete distractors obtained from context, but from replacements of words with the same PoS.

The ranking methods consider the generations of each method, thus significantly increasing the coverage and still having a better proportion of suitable distractors (except for Nb). The 4% of distractors missing with these methods occur in a minority of situations where no generation method could generate a distractor. Among the models used, BERTimbau is preferable to GPorTuguese-2. This is not necessarily due to the model architecture, but may be caused by the data they were pretrained on. BERTimbau was pretrained for Portuguese from scratch, whereas GPorTuguese-2 is GPT2, pretrained for English, then fine-tuned for Portuguese. Performance of the two versions of BERTimbau, base and large, are very similar.

5 Conclusion

We have described the implementation of several methods for generating distractors, to be used in the creation of MCQs in Portuguese. They are complementary but their combination and ranking by a language model provides both the best coverage and accuracy. The utility of such a straightforward method was confirmed by an experimentation where distractors were generated for a selection of questions and then manually classified.

This research contributes to the development of SmartEDU, a platform that aims at accelerating the process of producing education materials [5], with a focus on MCQs and slide deck generation [4]. In the future, we will work on improving the current methods and how some deal with incorrect spellings, such as missing accents, missing characters, or unexpected characters (e.g., seculo 19, assistência de financiamento, -Assistência de financiamento). Due to the low quality of some translations in the version of SQuAD used, we will consider experimentation in other datasets (e.g., factoid sentences and questions [10], or questions manually produced for SmartEDU). Moreover, we will devise the inclusion of additional methods and explore other language models, not only for ranking, but also for generating distractors. For English, several options are available, such as a T5 transformer fine-tuned for distractor generation [16], given a context, a question and an answer. A similar model could be trained for Portuguese, possibly taking advantage of SQuAD. Generating everything with a language model is indeed more flexible, requires less programming and access to less third-party tools and resources. With some recent models, it can be done with a simple instruction prompt [19], which may additionally include a few complete examples for guiding generation (e.g., few-shot learning). On the other hand, the proposed approach has the main advantage of being transparent. For instance, we can easily track the origin of the distractors and discriminate them by type.

We make the implementation of the generation and ranking methods available from the following notebook:
https://github.com/NLP-CISUC/smartedu-aqg/blob/main/Generating_Ranking_Distractors_PT.ipynb
References


