Towards a multi-agent dialogue system with contextual agents and tailored distractors

Leonor Llansol
leonor.llansol@tecnico.ulisboa.pt
Instituto Superior Técnico
Lisbon, Portugal

Abstract

When training an end-to-end model to perform response selection, most systems take advantage of a possible answer (gold reply) and one or more not possible answers (the distractors). The latter are randomly selected from the corpus, despite the fact that, in a real scenario, possible response candidates are usually similar to the gold reply. Therefore, in this work, we introduce the concept of tailored distractors, corresponding to different methods of selecting distractors that are closer to the gold reply. We show that these distractors have a positive impact in the response selection task, but also if we consider a generative dialogue system.

Keywords: Response selection, distractor selection

1 Introduction

Chatbots have been getting a great deal of attention lately, in a time when NLP is developing faster than ever. They are a type of dialogue system, or conversational agent, designed to have extended conversations with the user, having a similar behavior to human interaction (Jurafsky and H. Martin, 2019). They can be generative based, where their responses are generated, or retrieval based, where their main focus is the task of response selection. This consists on, among a set of possible responses, select the correct one, considering the context of the conversation.

These systems' training consists on, for each context, feeding it a positive example – the correct response – and a negative example – a distractor. The negative examples are usually randomly sampled utterances (Lowe et al., 2016).

However, in retrieval-based systems, a search engine is used to retrieve a number of candidates, from which the model selects a response. Thus, the candidates already have some degree of similarity between them, as proven by an experiment where we used a corpus of 360 chitchat questions for the Portuguese language, and a corpus with movie subtitles (Ameixa et al., 2013) to retrieve the candidate responses. First, we computed the similarity between all the candidate responses retrieved by Whoosh[1]; then, we did the same but with responses randomly chosen from that corpus. In both settings, for each question in the chitchat corpus, n candidates were retrieved, where n is 2, 5, 10 or 20. The results are shown in Table 1, where each value is averaged over five runs of that setting.

# Retrieved candidates	Whoosh	Random
2	0.3499 ±0.2445	0.1703 ±0.20736
5	0.3427 ±0.2410	0.1693 ±0.21034
10	0.3295 ±0.2250	0.16676 ±0.20818
20	0.3252 ±0.2196	0.16936 ±0.20844

Table 1. Spacy similarity of responses retrieved by Whoosh and random responses

We conclude that, on average, when using Whoosh, the similarity amongst candidates decreases as the number of candidates increases; no correlation is found when using random candidates. Furthermore, we see that candidates retrieved by the search engine are, on average, two times more similar than the ones randomly retrieved.

Therefore, training a model with random distractors may not be the best choice, when, in a real-world scenario, the model will have to distinguish a correct answer among a set of strong contenders. Here, the question that we intend to answer in this paper arose: will it impact our model's performance to select tailored distractors, rather than choosing them randomly?

Through this study, we aim to answer the following questions:

1. Does selecting tailored distractors impact the performance metrics?
2. Is a current neural dialogue system sensitive to changes in the context of the conversation?

To answer Question 1, we propose four techniques: noisy, Whoosh, semantic similarity and top ranking distractor selection. We evaluate these on an adaptation of the TransferTransfo (Wolf et al., 2019) model that does both retrieval and generation, using a GPT-2 based model, particularly, DialoGPT (Zhang et al., 2020b). Finally, we choose the setting with best results and test it on a customer support corpus, and make experiments with different perturbations introduced in the context to see if the model is affected by them, answering Question 2.

[1] https://whoosh.readthedocs.io/en/latest/intro.html (Last accessed on: 06/12/2020)

2 Related Work

In the previous section, it was seen that the task of NSP includes fetching a random sentence from a corpus, as a negative example. In this section, we study the importance of selecting negative examples using some heuristic, instead of selecting them randomly, as one of the objectives of this thesis is to study the hypothesis that selecting tailored distractors improves the performance of a retrieval and generative model.

The task of **distractor selection** was created to aid in the creation of multiple choice questions (MCQ) from long texts. Mitkov and Ha (2003) introduced this task that uses NLP methods, such as term extraction, word sense disambiguation and WordNet (University, 2010), to generate questions and corresponding items. While one of the items is the correct answer, the others are **distractors**, which must be *semantically close* to the correct answer, so that finding the correct answer is less obvious for students. In this novel approach, the selection of distractors is done using WordNet. Through user evaluation, it was realized that, from all the tasks involved in MCQ generation, the task of distractor selection was the one that needed more improvement.

Mitkov and Ha (2003) created MCQs from long texts, but using only one sentence of the text for each question. Araki et al. (2016) was the novel system to create MCQs from multiple sentences, in a way that requires the student to take inference steps, such as coreference resolution, to find the correct answer.

Mitkov et al. (2009) studied how to improve the quality of the selected distractors by testing different ways of calculating semantic similarity, but no method was found to outperform the others.

Another traditional task that has been automatized and uses distractors is Cloze (Taylor, 1953) (Jiang and Lee, 2017) (Gao et al., 2020), which is a test where parts of a text have been removed and the student must fill the gaps, choosing from a set of candidates that include the correct missing span and distractors.

The mentioned systems train their models using English exams designed by teachers.

The task of distractor selection for multiple choice questions usually consists on computing a metric that compares each distractor (d) to the correct answer (c). Namely, as mentioned, for the task of multiple choice questions, Mitkov and Ha (2003) computes the semantic similarity using the Wordnet, which retrieves hypernyms and hyponyms, to have d semantically close to c. Gao et al. (2020), to select distractors for the Cloze task, use the **length difference** between c and d, the **cosine similarity** between c and d, the **distractor frequency**, where d has highest score if it appears less, and the **frequency difference** between c and d. Jiang and Lee (2017), also for the Cloze task, compute a **semantic similarity** using word2vec (Mikolov et al., 2013), a **spelling similarity** and a **word co-ocurrence similarity**, assuming that sentences with common words or spelling are harder to distinguish by students.

On this work, we select distractors computing a semantic similarity with the correct response, among other methods, as we will see further.

Current dialogue systems, namely response selection systems, use a dialogue corpus, some of them the Ubuntu Dialogue Corpus (Lowe et al., 2016). To train their models, for each training example, they need the context of a conversation, and one positive and one (or more) negative examples. This negative example is, in most response selection systems, *randomly* sampled from the corpus (Lowe et al., 2016) (Gunasekara et al., 2019) (Wu et al., 2017) (Zhou et al., 2016) (Zhou et al., 2018) (Henderson et al., 2019) (Gu et al., 2019) (Ma et al., 2019) (Yuan et al., 2019). (Zhang et al., 2017) propose a more sophisticated approach, where negative examples are randomly chosen from all other utterances *within the same document*, instead of randomly chosen from the whole corpus, so "distractors are likely from the same subconversation or even from the same sender but at different time steps". Devlin et al. (2019) also use random distractors in their NSP pre-training task.

Recent works have motivated the importance of selecting distractors instead of using random ones. Based on the assumption that, in real-world scenarios, models have to select a correct response from a set of strong distractors instead of random ones, that is, distractors that are harder to distinguish from the correct response than random ones, Lin et al. (2020) propose the creation of a grayscale dataset to train response selection systems: instead of considering the ground-truth response the `correct` response and all the distractors as `incorrect`, they use a multi-level ranking, where the ground-truth response is *white*, randomly sampled utterances are *black*, and utterances obtained using retrieval or generative systems are *gray*.

In order to evaluate how a response selection system performs with strong distractors, Sato et al. (2020) propose a method to build test sets with *well-chosen false candidates*. The choice of these candidates consists on retrieving candidates related to the ground-truth response, based on the similarity between their content words, and, from these, remove utterances that are acceptable as a response through human evaluation. This is, to the best of our knowledge, the closest approach to ours. Our **tailored distractors** are inspired by the distractor selection process in multiple choice question systems, and correspond to Sato et al. (2020)'s well-chosen false candidates, but, while they only use them for testing, we use them to **train** our model. Furthermore, we select them by taking into account the **similarity** between the whole sentences, whereas they only take the content words into account.

3 Tailored Distractors

In this section we describe the techniques considered to select distractors: noisy, using a search engine, semantic similarity and top ranking.

3.1 Noisy

The first approach consists on creating noisy distractors. This approach's goal is not to improve the performance metrics, but to study if retrieval models effectively use the distractors, or if their performance does not change when these are replaced with noisy data.

The noisy distractors are generated as random strings, so that they do not make sense syntactically nor semantically.

Given a corpus, the approach consists on replacing all the distractors by noisy distractors, while keeping the gold reply intact.

3.2 Search Engine

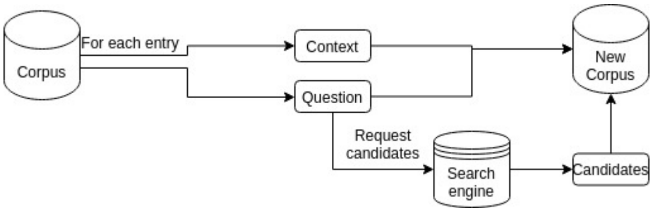

Figure 1. Search Engine approach

When there is a large number of candidates, response selection systems use search engines, such as Whoosh, to retrieve candidates, who are ranked according to that system's heuristic. To retrieve candidates from Whoosh, it is necessary to have question-answer pairs, which are used to create indexes. Then, given a query and a number of hits, *n*, the search engine finds the *n* candidates that better matches that query, and returns their answers, ordered by their level of matching.

Given a corpus, *c*, the first step is to create indexes: first, preprocess *c* to only consider question-answer pairs, namely, for each entry, the last utterance in the history and the gold response. Having created the indexes, a new version of *c* is created: for each history, we give Whoosh the last utterance and request *k + 1* candidates, depending on the number of distractors, *k*, wanted. From those, the first retrieved candidate will be the gold reply and the remaining *k* candidates are shuffled and used as distractors. The resulting dataset requires an additional processing step, which is to delete entries with less than *k + 1* candidates, which can happen because, occasionally, the search engine does not match the question sent with the requested number of hits.

The process of creating a dataset with a search engine retrieved distractors is illustrated in Figure 1.

3.3 Semantic similarity

Another approach on distractor selection is based on its semantic similarity with the gold reply. As previously seen, in the multiple choice question generation task, the selected distractors have a high degree of similarity with the gold reply, enough to make it difficult for students to select the correct answer without minimal domain knowledge, but are not paraphrases of the gold reply.

Our approach consists on selecting distractors that are semantically similar to the gold reply, without being paraphrases. To do this, using a natural language inference corpus, the average semantic similarity for the `paraphrase` relation is computed. Then, a set of random utterances is sampled from the corpus, and the similarity between each of them and the gold reply is calculated. The ones selected as distractors are those with higher similarity below the paraphrase threshold.

Thus, to build a corpus with this method, it is necessary to have a **previous corpus**, *c*, with gold replies, a **NLI corpus**, *c_nli*, in the same language as *c*, and a **method to compute semantic similarity**, *m*.

3.4 Top-ranking

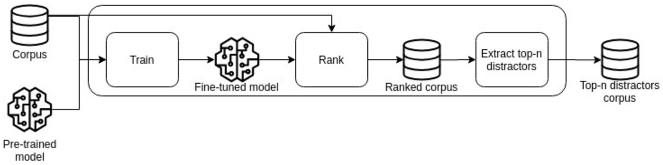

Figure 2. Top-ranking approach

This approach requires a ranking pre-trained model, *m*, and a corpus, *c*, with a set of entries consisting on a conversation history, a gold reply, and a set of distractors. We assume that *c* contains *n* distractors by entry, but, due to memory limitations, only *d*, randomly chosen, are used during training, where $d <= c$.

The approach consists on using a ranking model to rank candidates according to their probability of being the gold reply. Then, the *d* candidates with a higher probability, excluding the one that is the gold reply, are used as distractors in a new corpus, where each entry has the same conversation history and gold reply as the original corpus, the distractors being the only difference. Thus, the distractors in the tailored corpus are the ones that the model had higher difficulties to tell apart from the gold reply, and can then be used to train the model from scratch. This is more similar to a real world setting where the model will have to select a gold reply from a set of strong distractors. This can be seen in Figure 2.

By doing this, we are training our model with the highest ranked distractors, which is more similar to a real world setting where the model will have to select a gold reply from a set of strong distractors.

4 Evaluation setup

To evaluate the impact of our distractors, we use a system that does both retrieval and generation. In this section, we describe that system, the scenarios used to construct the models with the different distractors, along with the metrics used to evaluate them.

4.1 DialoGP3T

To study the impact of tailored distractors, we use an adaptation of the TransferTransfo (Wolf et al., 2019) model that is trained on a multi-task setting with two goals: minimize the *language model* loss, in order to generate plausible responses, and minimize the *multiple choice* loss, in order to correctly classify a gold response among a set of distractors. It needs a pre-trained GPT-2 model and a dataset.

For our experiments, we use the pre-trained model `microsoft/DialoGPT-small`[2] (Zhang et al., 2020b). It is a neural model for response generation, trained on Reddit dialogue data. Since we use an adaptation TransferTransfo with DialoGPT, we call this model DialoGP3T.

As training data, we use the *PersonaChat* dataset[3], which contains 17898 entries, where each entry contains a personality (a few sentences describing the agent), and a set of utterances, with each containing a set of candidates, where the last one is the gold reply and the others are distractors, and a conversation history.

4.2 Scenarios

In order to perform our experiments with distractors, we use two scenarios, one for Portuguese and one for English. Since this work was developed under the scope of project MAIA: Multilingual AI Agent Assistants[4], whose goal is to develop a platform where AI agents perform customer support, we also make an experiment using our tailored distractors and a customer support dataset.

4.2.1 Portuguese. For the experiments in Portuguese, `Bert for next sentence prediction` is used, with the `bert-base-multilingual-cased-sentence`[5] model.

Since, to the best of our knowledge, there is not structured dialogue data for this language, we translated 5000 dialogues from the Cornell Movie Dialogue Corpus[6].

- **Noisy** – As we will see further, using noisy distractors significantly decreased the retrieval metrics, therefore we decided not to test them for the Portuguese language.
- **Search Engine** – The search engine we use is Whoosh. Since the model used for the Portuguese language is `BertForNextSentencePrediction`, to fine-tune it we need, for each utterance, one positive and one negative example. Thus, in this setting, only one distractor is retrieved from Whoosh. For this experiment, we used the Whoosh indexes that were already created for the Subtle corpus. For each utterance of our corpus, 3 candidates were retrieved by Whoosh, and the 3rd one was used as a distractor.
  To select whether we will fine-tune our `BertForNextSentencePrediction` model with 2, 3 or 4 epochs, we make experiments with 4 epochs of train and then select the one with higher average accuracy. Since the candidates retrieved from Whoosh are deterministically chosen, instead of creating five different datasets, as in the previous experiment, we create one dataset and randomly split it five times into training and testing set, and fine-tune the model with the training one, computing the accuracy at the end of each epoch. We also repeat this four times for each, in order to obtain enough results to measure if they have statistical significance.
  We repeat the aforementioned process for datasets created with *random* distractors, and testing the models with a Whoosh validation set, to see if selecting Whoosh distractors in training improves the results in testing, compared to selecting them randomly.
- **Semantic Similarity** – To evaluate this approach, it is necessary to select a threshold and a similarity method. Two natural language inference corpus for the Portuguese language were joined: SICK_BR (Real et al., 2018) and ASSIN-1[7].
  The resulting dataset contained 60.8% of the sentence pairs `NEUTRAL`, 9.6% `CONTRADICTION`, 18.4% `ENTAILMENT` and 11.2% `PARAPHRASE`.
  To compute the semantic similarity between a gold reply and a given utterance, we tested two approaches: Spacy and BERT.
  Spacy has a `similarity`[8] method that computes a cosine similarity over word vectors. There are three models available for the Portuguese language: small (`pt_core_news_sm`), medium (`pt_core_news_md`) and large (`pt_core_news_lg`). Both medium and large models include word vectors trained using FastText CBOW on Wikipedia and OSCAR, while the small one does not include this feature. The difference between them is the number of unique vectors: the medium model

[2] https://huggingface.co/microsoft/DialoGPT-small(Last accessed on: 27/11/2020)

[3] https://s3.amazonaws.com/datasets.huggingface.co/personachat/personachat_self_original.json (Last accessed on: 19/10/2020)

[4] https://resources.unbabel.com/maia-unbabel-research (Last accessed on: 23/11/2020)

[5] https://huggingface.co/DeepPavlov/bert-base-multilingual-cased-sentence (Last accessed on: 21/12/2020)

[6] https://www.cs.cornell.edu/ cristian/Cornell$_{M}ovie - Dialogs_{C}orpus.html(Lastaccessedon : 23/11/2020)$

[7] http://propor2016.di.fc.ul.pt/?page$_{i}d = 381(Lastaccessedon :$ 12/12/2020)

[8] https://spacy.io/usage/vectors-similarity (Last accessed on: 30/11/2020)

has 20000 and the large one has 500000 unique vectors. Thus, two settings were tested using spacy: with the medium (`spacy md`) and the large (`spacy lg`) models. Regarding the BERT model, its output consists of the word embedding of each token, plus an embedding for an extra token, `[CLS]`, representing the whole sentence and used for sentence classification purposes. To test BERT to compute the semantic similarity between two sentences, two approaches were used: compute the cosine similarity over the `[CLS]` tokens of the two sentences (`BERT cls`), and compute the same similarity over the average of the embeddings of the words from each sentence (`BERT avg`).

Each setting was ran on the described inference dataset, and then averaged over each label. The results are shown in Table 2.

Setting	E	N	P	C
Spacy md	0.7907	0.6904	0.8582	0.7802
Spacy lg	0.7891	0.6874	0.8571	0.7773
BERT cls	0.9698	0.9599	0.9817	0.9776
BERT avg	0.8528	0.7819	0.9009	0.8709

Table 2. Similarity by label

Given the settings results, we chose the one that better differentiated the labels. The `BERT cls` setting has very high and close results, so we excluded it. From the remaining, our intuition is that sim(NEUTRAL) < sim(CONTRADICTION) < sim(ENTAILMENT) < sim(PARAPHRASE). The `BERT avg` setting has sim(CONTRADICTION) > sim(ENTAILMENT), so we excluded it. Between the remaining, `Spacy md` and `Spacy lg`, since no significant difference was seen between them, `Spacy md` was chosen for being a smaller model.

As in the previous approach, we make experiments with 4 epochs of train and then select the one with higher average accuracy. The experiments consist on creating five different datasets, which will always be different because the distractor chosen for each gold reply is the one most similar to the gold reply but not too similar from a set of 100 randomly sampled utterances. Then, for each dataset, we split it into training and validation sets, and fine-tune the model with the training set, computing the accuracy at the end of each epoch. We repeat this four times for each, in order to obtain enough results to measure if they have statistical significance.

We repeat the aforementioned process for datasets created with *random* distractors, and testing the models with tailored distractors, to see if selecting tailored distractors in training improves the results in testing, compared to selecting them randomly.

- **Top ranking** – To select distractors using the top ranking approach, a ranking model is needed. For the English scenario, the DialoGP3T model is used, that does both ranking and generation, allowing us to observe the impact of our distractors in both. As previously seen, this model requires a pre-trained GPT-2 model, namely DialoGPT for the English language. However, there is no multilingual DialoGPT model, nor one for the Portuguese language; the only one available is `pierreguillou/gpt2-small-portuguese`[9]. Since this model was not fine-tuned for dialogue, we decided not to use it and only test the top ranking approach with English data.

4.2.2 English. For the experiments for the English language, we use the previously introduced DialoGP3T model, and the PersonaChat dataset.

Regarding the Semantic Similarity technique, for Portuguese it was done using SICK_BR and Assin data and the spacy similarity method. However, for the English language, using the MultiNLI corpus (Williams et al., 2018), labeled with `entailment`, `neutral` and `contradiction` relations, and the MSR paraphrase corpus [10], containing sentence-pairs labeled as `paraphrases`, the tested method for computing semantic similarity, spacy, with the English medium model, did not differentiate the labels as expected, with a `neutral` average of 0.78725 ± 0.087, `entailment` of 0.7746 ± 0.0925, `contradiction` of 0.7811 ± 0.0910, and `paraphrase` of 0.95675 ± 0.03, when we expected a more significant difference between the first three. Furthermore, when computing the average similarity of PersonaChat's responses with random distractors from the corpus, the result was 0.93, which, from the attained similarity averages, would suggest that all the utterances from the corpus were paraphrases, which does not make sense. Therefore, we decided not to use the semantic similarity approach for the English language.

The other techniques, noisy, search engine and top ranking, were used as previously described using this model and dataset.

4.3 Evaluation Metrics

4.3.1 Ranking. To evaluate the impact of tailored distractors on ranking, in our DialoGP3T model, we use the metric Hits@k, k in [1,5,10], which represents the correct answer in the top k hits. When k = 1, the result is the accuracy, which is the metric used for the Portuguese experiments, with `Bert for next sentence prediction`.

[9] https://huggingface.co/pierreguillou/gpt2-small-portuguese (Last accessed on: 12/12/2020)

[10] https://www.microsoft.com/en-us/download/details.aspx?id=52398 (Last accessed on: 12/12/2020)

4.3.2 Generation.

To evaluate the impact of tailored distractors on generation, in our DialoGP3T model, the following metrics are used:

- BLEU (Papineni et al., 2002) – evaluates machine translation by measuring how many words overlap on a translation and a reference translation
- TER Score (Snover et al., 2006) – evaluates machine translation by measuring how much a translator would have to edit a translation so that it would match a reference translation
- BertScore (Zhang et al., 2020a)– evaluates text generation by measuring the similarity between a candidate and a reference sentence

5 Results

In this section, we present the results obtained. In Section 5.1, we show the results obtained in testing models trained with our different tailored distractors. Then, in Section 5.2, we select the best technique and test it in a customer support dataset. Finally, in Section 5.3, we show if our DialoGP3T model is sensitive to perturbations in the context.

5.1 Comparison of distractor selection methods

5.1.1 Portuguese.

- Search engine – Table 3 shows the results, averaged over all the results obtained as described above, and their statistical significance. We observe that training the model with the Whoosh distractors improves, on average, 5% compared to training with the random ones. The original results can be consulted in the thesis.
- Semantic similarity – Table 4 shows the results, averaged over all the results obtained as described above, and their statistical significance. We observe that training the model with the tailored distractors improves, on average, 3% to 4% compared to training with the random ones. The original results can be consulted in the thesis.

5.1.2 English.

Four different datasets are used to train DialoGP3T: one with random distractors, used as a baseline, and the others built using the previously seen techniques, resulting in four different models: R (random), N (noisy), W (search engine – Whoosh) and T (top ranking).

To have results with statistical significance, we trained the aforementioned models with five different seeds. We tested them with testing data made of random (Table 5) and tailored distractors (both T and W, Table 6). The original tables with values across all seeds can be consulted in the thesis. Note that these testing sets have the same gold replies and history; only the distractors are different. Namely, the random testing set has 19 distractors, and both the tailored ones only have 4. Thus, since distractor selection does not affect generative results, only the ranking results change and, since in tailored test sets there are only 5 candidates, the Hits@5 and Hits@10 metrics are always 1, which explains why only the Hits@1 results are shown in Table 6.

We observe that, regarding the Hits@1 metric, the model trained with the top-rank distractors has the best results. Regarding the Hits@5 and Hits@10 metrics, the model trained with random distractors shows the best results. Regarding the generation metrics, the model trained with noisy distractors surprisingly shows the best results, the BLEU and BertScore metrics.

Looking at the tailored testing results (Table 6), we observe that the setting with best results is the one whose training set contains distractors selected the same way as in the testing set, both for top-rank and Whoosh.

To assess the significance of these results, for each metric, we gather the five different results, one by seed, of each model. Then, to compare two models, we calculate the *p-value* using their corresponding results. If *p-value* < 0.05, we consider the result to be *significant*. We do this to assess if the following hypothesis are true: the T model has best Hits@1 result for R testing; the N model has best BLEU and BertScore results for R testing; the T model has best Hits@1 result for T testing; and the W model has best Hits@1 result for W testing. The results are shown in Table 7, where X > Y for Z stands for the hypothesis that model X performs better than model Y on test set Z, using the metric specified in column Metric.

The only hypothesis that is not statistical significant is *the* N *model has best BertScore results for* R *testing*, which means that the improvement observed may be by chance. All the other hypothesis are statistically significant, namely:

1. the T model has best Hits@1 results than the R model for R testing;
2. the N model has best BLEU results than the R model for R testing;
3. the T model has best Hits@1 results than the R model for T testing;
4. the W model has best Hits@1 results than the R and T models for W testing.

From 3. and 4., we conclude that **for scenarios with strong distractors, training a model using strong distractors *generated using the same heuristic* is a better option than using random ones**.

5.2 Customer Support

Tables 8 and 9 show the average results across five different seeds obtained for a test set with, respectively, randomly and top-rank chosen distractors. Since the number of candidates was 10 for the random test set and 5 for the tailored test set, we omit the Hits@10 metric for the first and also the Hits@5 metric for the latter, which are always 1. Also, as mentioned in previous experiments, the generative scores are independent of the distractors, thus are not shown in the

Epoch	Random	Whoosh	*p-value*	Significant
1	0.49454 ± 0.00953	**0.55024** ± 0.00262	9.68417e-5	Yes
2	0.49176 ± 0.00356	**0.5541** ± 0.00470	2.77729e-8	Yes
3	0.49468 ± 0.00696	**0.55556** ± 0.00322	3.62604e-6	Yes
4	0.49452 ± 0.00400	**0.55678** ± 0.00402	8.10505e-9	Yes

Table 3. Context agent random and Whoosh train, Whoosh test

Epoch	Random	Tailored	*p-value*	Significant
1	0.50314 ± 0.00634	**0.53662** ± 0.00593	0.00003	Yes
2	0.50296 ± 0.00438	**0.54258** ± 0.00784	0.00005	Yes
3	0.50452 ± 0.00497	**0.54462** ± 0.00883	0.00009	Yes
4	0.50626 ± 0.00544	**0.54578** ± 0.00950	0.00014	Yes

Table 4. Context agent random and tailored train, tailored test

Metric	R	N	W	T
Hits@1	0.81882 ± 0.00513	0.05223 ± 0.01683	0.75448 ± 0.04487	**0.83476** ± 0.00455
Hits@5	**0.97736** ± 0.00126	0.27192 ± 0.06292	0.96324 ± 0.00881	0.97702 ± 0.00125
Hits@10	**0.99644** ± 0.00036	0.52522 ± 0.07984	0.99248 ± 0.00160	0.99514 ± 0.00055
BLEU	2.62674 ± 0.15890	**2.90554** ± 0.11438	2.70748 ± 0.15361	2.67988 ± 0.10051
TER	1.035 ± 0.01390	1.0419 ± 0.02092	1.041 ± 0.02826	**1.02646** ± 0.00895
BertScore	0.84874 ± 0.01223	**0.85576** ± 0.00117	0.8555 ± 0.00103	0.85468 ± 0.00119

Table 5. Mean and stdev by metric and training set for *random* testing

Test set	R	N	W	T
W	0.74712 ± 0.00214	0.20520 ± 0.03418	**0.80614** ± 0.02922	0.77470 ± 0.03960
T	0.82404 ± 0.00413	0.20408 ± 0.03762	0.75388 ± 0.00444	**0.84582** ± 0.00343

Table 6. Seed variation Hits@1 results (T and W test)

Hypos	Metric	Original values	Test values	p-value	Significant
T > R for R	Hits@1	0.81882 ± 0.00513	**0.83476** ± 0.00455	0.00086	Yes
N > R for R	BLEU	2.62674 ± 0.15890	**2.90554** ± 0.11438	0.01465	Yes
N > R for R	BertScore	0.84874 ± 0.01223	**0.85576** ± 0.00117	0.26942	No
T > R for T	Hits@1	0.82404 ± 0.00413	**0.84582** ± 0.00343	0.00002	Yes
W > R for W	Hits@1	0.74712 ± 0.00214	**0.80614** ± 0.02922	0.01052	Yes
W > T for W	Hits@1	0.75388 ± 0.00444	**0.80614** ± 0.02922	0.01536	Yes

Table 7. Seed variation results (T and W test)

second table. The original results before averaging can be consulted in the thesis.

5.3 Is DialoGP3T sensitive to perturbations in the context?

To see if our DialoGP3T model is sensitive to changes in the context, we perform an ablation study where we introduce perturbations in the context to assess whether the performance metrics change. Here, we assume that the context does not include the most recent utterance, since it is the one to which the model will generate a response. Since the goal is to study the perturbations' impact independently, for each of them a new version of the corpus is produced, where the only difference is the context. The perturbations are the following:

- **No context** – delete all the utterances in the context.
- **Half context** – randomly delete half of the utterances in the context.

Metric	R	T	*p-value*	Significant
Hits@1	0.73362 ± 0.00913	**0.78172** ± 0.00927	0.00003	Yes
Hits@5	0.99134 ± 0.00205	**0.99324** ± 0.00084	0.11002	No
BLEU	**11.12594** ± 0.22760	11.00848 ± 0.23538	0.44566	No
TER	**1.0243** ± 0.01417	1.02768 ± 0.01648	0.73721	No
BertScore	0.8514 ± 0.00547	**0.85324** ± 0.00158	0.50423	No

Table 8. Xbox average results (random test)

Metric	R	T	*p-value*	Significant
Hits@1	0.73504 ± 0.00920	0.78364 ± 0.00908	0.00003	Yes

Table 9. Xbox average results (tailored test)

- **Shuffle context** – shuffle all the utterances in the context.

Using the new three corpora, we train three versions of DialoGP3T: `no context model`, `half context model` and `shuffle context model`. These models are trained through 1 epoch, and with five different seeds, to assess their results' statistical significance. They are then tested using the PersonaChat validation corpus, with the original context.

Table 10 shows the mean and standard deviation of each setting (`O` represents the `original model` (intact context), `N` the `no context model`, `H` the `half context model` and `S` the `shuffle context model`) across the different seeds (original results before averaging can be consulted in the thesis), and the *p-value* obtained by performing a *ttest* on each setting with the original setting, in order to assess if the obtained values have statistically significance. We use $alpha = 0.05$, thus a result is significant if it has $p - value <$ 0.05. We observe that the only setting with significant results across all metrics is `No Context model` (`N`), whose results are significantly worse than those of the original model, namely, with a Hits@1 of 0.6142, compared to the original's 0.8188. The two other settings also show significantly worse results for the Hits@1 metric, but only approximately less 0.01 than the original; their BLEU results are significantly better than the original ones.

This experiment has shown that DialoGP3T **takes the context of a conversation into account when selecting a response**, since its ranking accuracy decreases around 20% when the context of the conversation is removed from the dataset, thus demonstrating the model's robustness.

6 Conclusions

In conclusion, we have presented different ways to select distractors while training language models, and showed that, for scenarios with strong distractors, training a model using strong distractors generated using the same heuristic gives better results than using random ones. We also showed that DialoGP3T uses context when selecting and generating responses, since its results drastically change when trained with a dataset without the conversation history.

References

D. Ameixa, L. Coheur, and R. A. Redol. From subtitles to human interactions: introducing the subtle corpus. Technical report, Tech. rep., INESC-ID (November 2014), 2013.

J. Araki, D. Rajagopal, S. Sankaranarayanan, S. Holm, Y. Yamakawa, and T. Mitamura. Generating questions and multiple-choice answers using semantic analysis of texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1125–1136, Osaka, Japan, Dec. 2016. The COLING 2016 Organizing Committee. URL https://www.aclweb.org/anthology/C16-1107.

J. Devlin, M.-W. Chang, K. Lee, and K. 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://www.aclweb.org/anthology/N19-1423.

L. Gao, K. Gimpel, and A. Jensson. Distractor analysis and selection for multiple-choice cloze questions for second-language learners. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 102–114, Seattle, WA, USA â†' Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.bea-1.10. URL https://www.aclweb.org/anthology/2020.bea-1.10.

J.-C. Gu, Z.-H. Ling, and Q. Liu. Interactive matching network for multi-turn response selection in retrieval-based chatbots, 2019.

C. Gunasekara, J. K. Kummerfeld, L. Polymenakos, and W. Lasecki. DSTC7 task 1: Noetic end-to-end response selection. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 60–67, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4107. URL https://www.aclweb.org/anthology/W19-4107.

M. Henderson, I. Vulić, D. Gerz, I. Casanueva, P. Budzianowski, S. Coope, G. Spithourakis, T.-H. Wen, N. Mrkšić, and P.-H. Su. Training neural response selection for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5392–5404, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1536. URL https://www.aclweb.org/anthology/P19-1536.

S. Jiang and J. Lee. Distractor generation for Chinese fill-in-the-blank items. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 143–148, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-

Metric	Dataset	Mean ± StDev	p-value	Significant
Hits@1	O	0.8188 ± 0.00513	-	-
	N	0.6142 ± 0.00894	3.4876x10^{-9}	Yes
	H	0.80632 ± 0.00581	0.00712	Yes
	S	0.8075 ± 0.00158	0.0060	Yes
Hits@5	O	0.97736 ± 0.00126	-	-
	N	0.88014 ± 0.00867	1.0944x10^{-5}	Yes
	H	0.97406 ± 0.00197	0.0167	Yes
	S	0.97502 ± 0.00227	0.0888	No
Hits@10	O	0.99644 ± 0.00036	-	-
	N	0.96332 ± 0.00382	3.885x10^{-5}	Yes
	H	0.99574 ± 0.00068	0.0892	No
	S	0.99566 ± 0.00082	0.1033	No
BLEU	O	2.62674 ± 0.15890	-	-
	N	1.47026 ± 0.21993	2.2732x10^{-5}	Yes
	H	2.92308 ± 0.10873	0.0106	Yes
	S	2.88062 ± 0.10773	0.0211	Yes
TER	O	1.035 ± 0.01390	-	-
	N	0.96830 ± 0.00735	7.2536x10^{-5}	Yes
	H	1.04446 ± 0.00679	0.2221	No
	S	1.03870 ± 0.01023	0.6456	No
BertScore	O	0.84874 ± 0.01223	-	-
	N	0.82348 ± 0.00921	0.00699	Yes
	H	0.85588 ± 0.00100	0.2624	No
	S	0.85554 ± 0.00121	0.2826	No

Table 10. Mean, stdev and p-value across seeds (Random vs each setting)

5015. URL https://www.aclweb.org/anthology/W17-5015.

D. Jurafsky and J. H. Martin. Speech and language processing, 2019. URL https://web.stanford.edu/~jurafsky/slp3/26.pdf.

Z. Lin, D. Cai, Y. Wang, X. Liu, H. Zheng, and S. Shi. The world is not binary: Learning to rank with grayscale data for dialogue response selection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9220–9229, Online, Nov. 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-main.741.

R. Lowe, N. Pow, I. Serban, and J. Pineau. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems, 2016.

W. Ma, Y. Cui, N. Shao, S. He, W.-N. Zhang, T. Liu, S. Wang, and G. Hu. TripleNet: Triple attention network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 737–746, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1069. URL https://www.aclweb.org/anthology/K19-1069.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

R. Mitkov and L. A. Ha. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17–22, 2003. URL https://www.aclweb.org/anthology/W03-0203.

R. Mitkov, L. A. Ha, A. Varga, and L. Rello. Semantic similarity of distractors in multiple-choice tests: Extrinsic evaluation. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 49–56, Athens, Greece, Mar. 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W09-0207.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

L. Real, A. Rodrigues, and A. Vieira. Sick-br: A portuguese corpus for inference: 13th international conference, propor 2018, canela, brazil, september 24–26, 2018, proceedings, 01 2018.

S. Sato, R. Akama, H. Ouchi, J. Suzuki, and K. Inui. Evaluating dialogue generation systems via response selection, 2020.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. USA, 2006. Association for Machine Translation in the Americas. URL https://www.cs.umd.edu/~snover/pub/amta06/ter_amta.pdf.

W. L. Taylor. cloze procedure: A new tool for measuring readability, 1953.

P. University. About wordnet., 2010.

A. Williams, N. Nangia, and S. R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference, 2018.

T. Wolf, V. Sanh, J. Chaumond, and C. Delangue. Transfertransfo: A transfer learning approach for neural network based conversational agents, 2019.

Y. Wu, W. Wu, C. Xing, M. Zhou, and Z. Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots, 2017.

C. Yuan, W. Zhou, M. Li, S. Lv, F. Zhu, J. Han, and S. Hu. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 111–120, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1011. URL https://www.aclweb.org/anthology/D19-1011.

R. Zhang, H. Lee, L. Polymenakos, and D. Radev. Addressee and response selection in multi-party conversations with speaker interaction rnns, 2017.

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

Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan. Dialogpt: Large-scale generative pre-training for conversational response generation, 2020b.

X. Zhou, D. Dong, H. Wu, S. Zhao, D. Yu, H. Tian, X. Liu, and R. Yan. Multi-view response selection for human-computer conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 372–381, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1036. URL https://www.aclweb.org/anthology/D16-1036.

X. Zhou, L. Li, D. Dong, Y. Liu, Y. Chen, W. X. Zhao, D. Yu, and H. Wu. Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1103. URL https://www.aclweb.org/anthology/P18-1103.