algorithms

MDPI

Article

RoSummary: Control Tokens for Romanian News Summarization

Mihai Alexandru Niculescu [1], Stefan Ruseti [1] and Mihai Dascalu [2,*]

[1] Computer Science & Engineering Department, University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania
[2] Research Technology, 19D Soseaua Virtutii, 060782 Bucharest, Romania
* Correspondence: mihai.dascalu@upb.ro

Abstract: Significant progress has been achieved in text generation due to recent developments in neural architectures; nevertheless, this task remains challenging, especially for low-resource languages. This study is centered on developing a model for abstractive summarization in Romanian. A corresponding dataset for summarization is introduced, followed by multiple models based on the Romanian GPT-2, on top of which control tokens were considered to specify characteristics for the generated text, namely: counts of sentences and words, token ratio, and n-gram overlap. These are special tokens defined in the prompt received by the model to indicate traits for the text to be generated. The initial model without any control tokens was assessed using BERTScore (F_1 = 73.43%) and ROUGE (ROUGE-L accuracy = 34.67%). Control tokens improved the overall BERTScore to 75.42% using <LexOverlap>, while the model was influenced more by the second token specified in the prompt when performing various combinations of tokens. Six raters performed human evaluations of 45 generated summaries with different models and decoding methods. The generated texts were all grammatically correct and consistent in most cases, while the evaluations were promising in terms of main idea coverage, details, and cohesion. Paraphrasing still requires improvements as the models mostly repeat information from the reference text. In addition, we showcase an exploratory analysis of the generated summaries using one or two specific control tokens.

Keywords: RoGPT2; control tokens; summarization; text generation; human evaluation

check for updates

Citation: Niculescu, M.A.; Ruseti, S.; Dascalu, M. RoSummary: Control Tokens for Romanian News Summarization. *Algorithms* **2022**, *15*, 472. https://doi.org/10.3390/a15120472

Academic Editor: Frank Werner

Received: 31 October 2022
Accepted: 6 December 2022
Published: 11 December 2022

1. Introduction

A remarkable development in Natural Language Processing (NLP) towards creating models that understand human languages has been observed in recent years. Text generation is one of the main challenges in the field of NLP, and this task has seen an important development after the introduction of Transformers [1]. The Transformer uses an encoder–decoder architecture, self-attention, and positional encodings to facilitate parallel training. The GPT-2 model developed by OpenAI [2] was the first model with remarkable text generation capabilities. GPT-2 was trained for predicting the next token in a sequence and could easily be adjusted for specific tasks. The follow-up improving the GPT-3 model [3] is more than 10-times larger in terms of the parameters and deduces the task only from the provided prompt. There have been several open-source variations of the model, such as GPT-Neo [4] and GPT-J [5]. Other architectures consider a unified framework to cover text-to-text formats and convert text-based language problems, such as the Text-To-Text Transfer Transformer (T5) [6]. This model can perform zero-shot learning and deduce the task from the context of the prompt received as the input, even if it was not presented in the training stage.

For the Romanian language, there are not many specific resources (i.e., pre-trained models and datasets), although there has been significant progress in recent years. The most notable models for Romanian consider the BERT architecture (e.g., RoBERT [7], BERT-base-ro [8], Distil-BERT [9]) and the GPT-2 architecture (e.g., RoGPT2 [10]) and were developed in the last 2 years. Romanian has only one available benchmark, namely LiRo [11]. However, the models are small compared to their English counterparts, and there are no available datasets for common NLP tasks. Overall, Romanian remains a low-resource language with low international usage (https://www.worlddata.info/languages/romanian.php; last accessed on 20 October 2022), despite recent efforts in terms of datasets and models; as such, we argue for the necessity of our efforts to develop tools tailored to this language.

Text summarization is a task of particular importance in NLP centered on extracting critical information from the text using two approaches. First, extractive summarization involves removing the most-important phrases or sentences that include the main ideas of a text. Second, abstractive summarization considers the generation of a new summary starting from the text. One of the most popular datasets in English used for this task is *CNN/Daily Mail* [12], having a total number of 280,000 examples; the dataset was afterward extended to other languages, including French, German, Spanish, Russian, and Turkish, thus generating the large-scale multilingual corpus *MLSUM* [13]. Another dataset used in studies for abstractive summarization is Extreme Summarization (*X-Sum*) [14] to generate a short, one-sentence summary for each news article; *X-Sum* was derived from BBC news and consists of 220,000 examples. Another dataset is *Webis-TLDR-17 Corpus* [15] with approximately three million examples constructed with the support of the Reddit community. Extractive summarization in Romanian has been previously tackled by Cioaca et al. [16] and Dutulescu et al. [17] with small evaluation datasets. We now introduce the first dataset for Romanian abstractive summarization (https://huggingface.co/datasets/readerbench/ro-text-summarization; last accessed on 20 October 2022).

A wide variety of architectures has been employed for text summarization, including general Transformer-based models [6,18–20] and specific models such as BRIO [21], ProphetNet [22], or PEGASUS [23]. We aim to provide a baseline abstractive summarizer for Romanian built on top of RoGPT2 [10] and to control the characteristics of the generated text. This is an additional step to better imitate human capabilities by considering one or more specifications that improve the summary. As such, we assessed the extent to which text generation is influenced by using control tokens specified in the prompt received by the model to induce specific characteristics of a text. The idea of specifying control tokens directly in the prompt was exploited first in MUSS [24] and CONTROL PREFIXES [25]. The GPT-2 model was also used in combination with BERT [26]; however, to our knowledge, the generation task was not tackled until now in combination with control tokens to manipulate the characteristics of the generated summary.

Following the introduction of various models for text summarization, evaluating the quality of a generated text is a critical challenge, which can be even more difficult than the text generation task itself. Text evaluation is generally performed using synthetic metrics developed for machine translation, such as Bilingual Evaluation Understudy (BLEU) [27], Recall Oriented Understudy for Gisting Evaluation (ROUGE) [28], or Metric for Evaluation for Translation with Explicit Ordering (METEOR) [29]; however, these metrics are limited as they focus on the lexical overlap. Newer metrics based on Transformers, such as BERTScore [30], BARTScore [31], or Bilingual Evaluation Understudy with Representations from Transformers (BLEURT) [32], are much more accurate compared to the classical metrics. Still, they require more resources (i.e., pre-trained models and higher computing power) and have longer processing times. Besides comparing automated similarity metrics, Celikyilmaz et al. [33] argued that a human evaluation is the gold standard for evaluating a Natural Language Generation (NLG) task; nevertheless, it is the most expensive and cumbersome to accomplish.

Thus, our research objective is threefold: create a dataset for summarization in Romanian, train a model that generates coherent texts, and introduce control tokens to manipulate the output easily. Following this objective, our main contributions are the following:

- Publish a clean version of the dataset for Romanian text summarization (https://huggingface.co/datasets/readerbench/AlephNews; last accessed 20 October 2022).

- Develop and publicly release a baseline model built on top of RoGPT-2 available on HuggingFace (https://huggingface.co/readerbench/RoSummary-large; last accessed on 20 October 2022), with the corresponding code released on GitHub (https://github.com/readerbench/RoSummary; last accessed on 20 October 2022).
- Study the use of control tokens for the text characteristics in the case of our summarization task.

2. Method

This section presents the dataset created for the summarization task, the model architecture, the training method with the control tokens, as well as the methods employed to evaluate the generated text.

2.1. Corpus

The dataset for the summarization task was constructed by crawling all articles from the AlephNews website (https://alephnews.ro/; last accessed on 20 October 2022) until July 2022. The site presents a section with the news summary as bullet points with sentences representing the main ideas for most articles. This peculiarity of the site enabled the automatic creation of a reasonably qualitative dataset for abstractive summarization. The news articles that did not have a summary or were too short were eliminated by imposing a minimum limit set of 20 characters. This resulted in 42,862 collected news articles. The news and summary texts were cleaned using several heuristics: these were the repair of diacritics, the elimination of special characters, the elimination of emoticons, and fixing punctuation (if it has more points, if it has no punctuation mark, a period is added at the end of the sentence), eliminating words such as "UPDATE", "REPORT", "AUDIO", etc. The dataset was split into 3 partitions (i.e., train, dev, and test) with proportions of 90%–5%–5%. Articles with a maximum of 715 tokens based on the RoGPT2 tokenizer were selected for the test partition; out of 724 tokens, 9 were reserved for the control tokens. After analyzing the dataset and based on the limitations regarding the sequence length of a context, the maximum size was set to 724 tokens. In the case of entries from the training and dev partitions having the combined length of the article and the summary greater than 724, the article content was divided into a maximum of 3 distinct fragments, which had the last sentences removed; this was applied to approximately 10% of the entries to increase the number of examples and to keep the beginning of the news, which contains key information to be considered. We chose not to apply this augmentation technique for the entries in the test partition, as this would have altered the content of the original texts and would have generated multiple artificial test entries; moreover, we limited the text to the first 715 tokens so that control tokens could also be added when running various configurations. The total number of examples for each partition was: 47,525 for training, 132 for validation, and 2143 for testing.

2.2. RoGPT2 Model for Summarization

The model was trained to predict the next token using the previous sequence, similar to the RoGPT2 [10] training for the Romanian language. The model architecture consists of several decoder layers of architecture Transformers [1], as presented in Figure 1. There are 3 versions of the model, each with a different number of decoder layers: 12 layers were used for the base version, 24 layers for the medium version, and 36 layers for the large version.

Control tokens were used to indicate the task and the characteristics of the generated text, which are presented in the following subsections. This assumes that the model maximizes the probability of a subword depending on the context and the previously generated subwords:

$$P(w_{1...m}) = \prod_{i=1}^{m} P(w_m | w_1, w_2, w_3, ..., w_{m-1}) \tag{1}$$

Cross-entropy was the loss function for the supervised learning task:

$$L_{CE} = -\sum_{i=1}^{n} t_i log(p_i) \quad (2)$$

where t_i is the label and p_i is the probability of the ith class, or more specifically, a class is considered the *id* of a token.

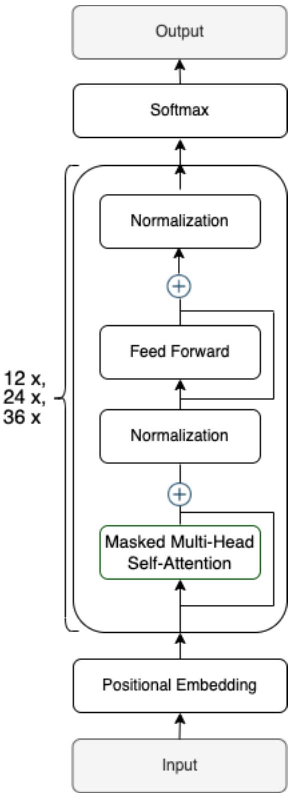

Figure 1. RoGPT2 architecture.

Due to a large number of parameters, the model was trained on TPU v3-8. The batch size was limited to fit into memory 724 tokens per entry. The Adam optimizer [34], the ReduceLROnPlateau (https://keras.io/api/callbacks/reduce_lr_on_plateau/; last accessed on 20 October 2022) and EarlyStopping (https://keras.io/api/callbacks/early_stopping/; last accessed on 20 October 2022) callbacks were used.

Three decoder methods for text generation were considered to choose the next token depending on the tokens generated up to that point and the probability distribution over the vocabulary.

Greedy search: This strategy is based on choosing a local optimum, in this case the token with the highest probability, which converges to a local maximum. First, the probability distribution is generated, and then, the next token is selected by choosing the highest probability. The procedure continues until the desired length is achieved or the token indicating the end is found. An advantage of this method is that it is efficient and intuitive, but it does not guarantee finding a global optimum for the generated sequence; this can lead to the non-exploration of some branches with a higher probability.

Beam search: Beam search [35] partially solves the maximum global problem by keeping the best beam width sequences with a higher total probability. Multiple contents are generated for each step, and the sequence with the highest probability is chosen at each step. The advantage of this method is that it obtains better results for relatively small beam widths, but it requires more memory for a larger beam width or longer sequences, whereas the text does not vary much, being quite monotone. Beam search also does not guarantee finding the global optimum. Beam search works quite well when it can approximate the generated text's length, but has issues when the corresponding length varies greatly. Holtzman et al. [36] argued that people do not choose the phrase with the highest probability as the element of unpredictability is important.

Top-p (nucleus) sampling: This method involves choosing the smallest subset of words with a probability equal to p. Based on the new probability distribution, a new token is chosen. The advantage of this method is that it achieves results quite close to human ones and does not require many resources. The disadvantage is that p is fixed and not dynamic.

2.3. Control Tokens

Starting from previous studies presented in the Introduction and related to the specifics of the summarization task, we chose to specify a set of 4 control tokens representative of various characteristics of the text, namely:

- **NoSentences** indicates the number of sentences that the summary should have;
- **NoWords** indicates the number of words to be generated within the summary;
- **RatioTokens** reflects how many times the sequence of tokens of the summary must be longer than the input;
- **LexOverlap** is the ratio of the number of 4-grams from the summary that also appears in the reference text; stop words and punctuation marks were omitted.

The first 3 control tokens are purely quantitative and reflect different use-case scenarios: a summary containing at most a specific number of sentences, a summary having an imposed number of words, or a compression ratio to be used globally. The last control token ensures a lower or higher degree of lexical overlap between the two texts.

The prompt for the summarization task was the following:

$$Text : \{article\}\ Summary : \{summary\} < |endoftext| > \tag{3}$$

The model learns that, after the control token **"Summary:"**, it must generate the summary of the text preceding that token. Control tokens are specified before the token that indicates the input (i.e., marked by the **Text** token), while the token specific to the task is placed after the end. The prompt used for an item from the dataset used for training is the following:

$$FeatureToken : \{value\}\ Text :\ \{article\}\ Summary : \{summary\} < |endoftext| > \tag{4}$$

where FeatureToken is <*NoSentences*>, <*NoWords*>, <*RatioTokens*>, or <*LexOverlap*>.

Following the initial experimentation, we noticed that the model learns best when subsequent entries have the same input text, but with different values for the control tokens and a different text to be generated; this refers to the extraction of fragments from the original summary and their use as the output. This variation is reflected in the text to be generated and was used for the <NoSentences>, <NoWords>, and <RatioToken> control tokens. The generation of multiple variations was applied if the summary text had more than 3 sentences; thus, incremental examples were generated by adding sentences and calculating the value for the control token each time. An example for a summary comprising 4 sentences s_1, s_2, s_3, s_4 and <NoWords> would consider two entries in the training dataset: the first item would consist of the first 3 sentences and the corresponding <NoWords> for this first shorter summary and a second item where the s_4 sentence would be added and <NoWords> is set at the global count of words from the summary.

Besides training the summarization model with each control token individually, we also considered combinations of 2 control tokens, namely: <NoWords>-<NoSentences>, <RatioTokens>-<NoSentences>, and <LexOverlap>-<NoWords>. The combination <NoWords>-<NoSentences> was chosen because it reflects the most straightforward manner to manually enforce the length of the summary by an end user (i.e., specify an approximate number of words and the number of sentences that the generated summary should have). <RatioTokens> presents the same idea as <NoWords>, only that it is much more difficult to learn by the model as it represents the ratio between the length of the news and that of the summary. The combination of <LexOverlap>-<NoWords> is interesting because it forces the model to generate a text with an approximate number of words. Still, the generated text must not match the one received by the model. <NoWords> indicates how many words the summary should have, while <LexOverlap> restricts the percentage of combinations of words that are present in the news and generated text by the model; a small value for <LexOverlap> indicates that the model must reformulate an idea from the news, whereas a large value makes the model extract the most important phrases within a word limit.

2.4. Evaluation Metrics

Our evaluations considered both automated and human evaluations of the generated summaries. We wanted the evaluation of the model to be a sustainable one; for this, the three evaluation metric methods used were: Recall Oriented Understudy for Gisting Evaluation (ROUGE) [28] as a classic metric, which is used in the majority of research in the field of abstract summarization, BERTScore [30], a metric that uses a pre-trained model to understand the generated text and the reference to provide a better comparison, and human evaluation. To evaluate the characteristics of the control token, the following metrics were used: Mean Absolute Error (MAE) and Mean-Squared Error (MSE) for <NoSentences> and <NoWords>, and the Pearson and Spearman coefficients were used for <RatioTokens> and <LexOverlap>.

2.4.1. BERTScore

Metrics based on Transformers [1], such as BERTScore [30], have been introduced to better capture the similarity between texts. BERTScore shows how good and realistic a text generated by a model is at the semantic level (i.e., the metric considers the meaning of the text by computing the cosine similarity between token embeddings from the generated sentences versus the tokens in the given sentences as a reference). The token embeddings are the numerical representations of subwords obtained using the BERT [37] tokenizer. The precision, recall, and F$_1$ scores are computed based on the scalar product between the embeddings in the two texts. Precision refers to the generated text and is calculated as the average value for the largest scalar product between the embeddings of the generated sentence and those of the reference sentence; in contrast, recall is centered on the reference text and is computed in an equivalent manner while considering the embedding of the reference versus the generated sentence embeddings. The original paper showed good correlations to human evaluations. Even if BERTScore is more accurate when compared to classical machine translation metrics, which account for the overlap between words using n-grams or synonyms (e.g., BLEU, ROUGE), the metric requires a language model for the targeted language. We used the implementation offered by HuggingFace (https://huggingface.co/spaces/evaluate-metric/bertscore; last accessed on 20 October 2022), which considers mBERT [37] for the Romanian language. The performance metrics are computed as follows:

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} max_{x_i \in x}(x_i^T \hat{x}_j) \tag{5}$$

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} max_{\hat{x}_j \in \hat{x}}(x_i^T \hat{x}_j) \tag{6}$$

$$F_{BERT} = 2 * \frac{P_{BERT} * R_{BERT}}{P_{BERT} + R_{BERT}} \tag{7}$$

where:

- x is the embedding for the text given as a reference;
- $\hat{x}$ is the embeddings for the text generated by the model.

2.4.2. Human Evaluation

Human evaluation is considered the gold standard in measuring the quality of generated text [33], but it is costly and difficult to achieve. For human evaluation, the most-used method is the one by which a form is created, and the respondents are asked to evaluate the generated text. In our case, correspondents were asked to assess the generated text from the point of view of five metrics: main idea (i.e., the main idea of the article is present within the summary), details (i.e., the key information is found in the generated text for irrelevant ideas), cohesion (i.e., phrases and ideas have a logic), wording/paraphrasing (i.e., the text is not the same as that of the news and the model-made changes), and language beyond the source text (i.e., there is a varied range of lexical and syntactic structures). The scores ranged from 1 to 4, the best being 4. The summary scoring rubric is based on the studies of Taylor [38] and Westley, Culatta, Lawrence, and Hall-Kenyon [39]. The raters were asked to evaluate 5 examples chosen randomly from the texts generated using the 3 decoding methods, and for 3 variants of the model; in total, 45 questions were included in the form. The Intraclass Correlation Coefficient (ICC3) [40] was calculated for each configuration and model-version-decoding method to measure the consistency of the evaluations. The form was sent to people collaborating with our research laboratory to obtain the relevant results, primarily due to the complexity of the 5 metrics used.

2.5. Experimental Setup

The Adam [34] optimizer started from a learning rate equal to 1×10^{-4} and was reduced to 4×10^{-6} using the callback ReduceLROnPlateau, for patience equal to 2 and a factor of 1/e. The patience parameter was set to 1 for combinations of control tokens due to the task's complexity and the dataset's size; the training was more aggressive, modifying the learning rate if there were no improvements after an epoch. The training was stopped if no improvements were noticed after 3 epochs for baseline summarization or 4 epochs for the control token. A context size equal to 724 was considered, and the batch size varied for each model version: 128 for the base, 24 for the medium, and 16 for the large models. Three decoding methods were used for text generation: greedy, beam-search, and top-p sampling. The experiments were performed on TPU v3.8 for training, while the NVIDIA Tesla A100 and NVIDIA Tesla P100 were used for text generation and evaluation. The model received prompts that contained the summary token and those that specified the characteristics of the text to be generated.

3. Results

This section presents the results obtained by the models for the summarization task and the experiments for control tokens. In most experiments, the same configuration was used for text generation. After training, the following generation strategies were used: greedy, beam search with a width equal to four, and top p sampling (with top k = 25 and p = 0.94). In addition, we introduced an exploratory analysis to highlight the benefits of using control tokens when generating summaries with various specificities.

3.1. News Summary

This experiment aimed to generate summaries for news articles without any particular characteristics. The model knows that it must generate text after the control token <Summary>. The evaluation of the model was performed using the metrics: ROUGE [28] score (the F1-score average was calculated for ROUGE-1, ROUGE-2, ROUGE-L) and BERTScore [30]. The results are available in Table 1. The medium version using beam search achieved the best scores (74.34% for BERTScore F_1 and 34.67% for ROUGE-L F_1), surpassing the large version with beam search by 0.1% for BERTScore.

Table 1. Results for the evaluation of news summaries (bold marks the best results).

Model	Decode Method	BERT Score			ROUGE		
		Precision (%)	Recall (%)	F_1 (%)	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-L (%)
Base	Greedy	73.35	73.99	73.58	33.60	18.62	33.33
	Beam Search	73.54	74.68	74.04	34.80	19.91	34.16
	Top-p Sampling	72.96	72.99	72.92	30.58	14.52	29.51
Medium	Greedy	73.78	74.01	73.80	34.22	19.22	33.94
	Beam Search	73.90	**74.93**	**74.34**	**35.46**	**20.61**	**34.67**
	Top-p Sampling	73.15	72.85	72.94	30.42	14.00	29.21
Large	Greedy	73.76	74.24	73.91	34.14	18.95	33.55
	Beam Search	**73.94**	74.70	74.24	34.92	19.95	33.84
	Top-p Sampling	73.11	73.01	72.99	30.51	14.18	29.31

3.2. Human Evaluations

The next experiment was to evaluate the model trained on the AlephNews dataset to generate summaries on the DigiNews test dataset introduced by Niculescu et al. [10]. As the DigiNews dataset does not have a summary for a news story, a human evaluation was performed to assess the quality of the generated text. The form was completed by six raters, and the scores from Table 2 consider the average for the five evaluated texts from each combination.

Table 2. Results for human evaluation (bold marks the best results).

Model	Decode Method	Main Idea	Details	Cohesion	Paraphrasing	Language	ICC3(1)	ICC3(k)
Base	Greedy	3.10	2.93	3.10	2.46	3.26	0.88	0.98
	Beam Search	2.73	2.86	2.86	2.03	3.40	0.93	0.99
	Top-p Sampling	2.70	2.50	2.53	1.90	3.00	0.92	0.98
Medium	Greedy	2.76	2.36	2.46	2.06	2.73	0.88	0.98
	Beam Search	3.43	**3.36**	3.30	2.00	3.56	**0.98**	**1.00**
	Top-p Sampling	2.56	2.30	3.16	2.63	3.33	0.92	0.98
Large	Greedy	**3.73**	3.06	**3.53**	2.30	**3.73**	0.92	0.99
	Beam Search	2.23	2.06	2.33	1.56	2.93	0.95	0.99
	Top-p Sampling	2.50	2.33	3.26	**2.70**	3.26	0.85	0.97

3.3. Control Tokens

For the following experiments, control tokens were used individually or in combination to indicate the characteristics of the generated text, in addition to the one indicating the task. For the more complex scenarios, we wanted to observe if the model learns a combination of several control tokens that were not reproduced in the training stage and if the order of tokes from the prompt matters. BERTScore [30] was used holistically as a means to compare different combinations; the Mean Absolute Error (MAE) and Mean-Squared Error (MSE) were considered for <NoSentences> and <NoWords>, whereas the Pearson and Spearman coefficients were used for <RatioTokens> and <LexOverlap>. Table 3 shows the best BERTScores obtained for each control token separately; the beam search and top-p sampling decoding methods were selected because they obtained the most revealing results. Detailed results for each control token are presented in Tables A1–A4. The best score was 75.42% with the <LexOverlap> control token.

Subsequently, we explored the extent to which the model succeeded in learning combinations of control tokens, having only examples for each one in the training stage. The following combinations of control tokens were chosen in line with the argumentation from the Method Section: <RatioTokens>-<NoSentences>, <NoWords>-<NoSentences>, <NoWords>-<LexOverlap>. We decided to focus only on the condensed results that consider BERTScore for the medium and large versions using beam search and the top-p sample as the decoding methods (see Table 4). Tables A5–A10 present the full results of the previous combinations. The best score was achieved by the combination of <NoWords>-<LexOverlap> using the medium version with beam search (F1 = 74.95%).

Table 3. BERTScore [30] for control tokens taken individually (bold marks the best results).

Control Token	Model	Decode Method		BERTScore	
			Precision (%)	Recall (%)	F_1 (%)
NoSentences	Base	Beam Search	73.69	73.53	73.54
		Top-p Sampling	72.52	72.24	72.32
	Medium	Beam Search	73.49	74.42	73.89
		Top-p Sampling	72.72	73.04	72.83
	Large	Beam Search	73.90	74.78	74.27
		Top-p Sampling	73.34	72.99	73.11
NoWords	Base	Beam Search	74.17	73.67	73.88
		Top-p Sampling	72.84	72.56	72.67
	Medium	Beam Search	74.71	74.45	74.55
		Top-p Sampling	73.43	73.07	73.23
	Large	Beam Search	74.90	74.67	74.75
		Top-p Sampling	73.53	73.27	73.37
RatioTokens	Base	Beam Search	74.81	72.48	73.55
		Top-p Sampling	73.22	71.59	72.32
	Medium	Beam Search	75.45	73.41	74.34
		Top-p Sampling	74.11	72.49	73.22
	Large	Beam Search	74.35	74.66	74.48
		Top-p Sampling	73.22	73.37	73.26
LexOverlap	Base	Beam Search	75.62	74.32	74.89
		Top-p Sampling	73.48	73.17	73.27
	Medium	Beam Search	75.90	**74.94**	75.36
		Top-p Sampling	73.95	73.88	73.87
	Large	Beam Search	74.37	73.83	74.05
		Top-p Sampling	**76.30**	74.66	**75.42**

Table 4. BERTScore [30] for complex control tokens (bold marks the best results).

Control Token	Model	Decode Method		BERTScore	
			Precision (%)	Recall (%)	F_1 (%)
RatioTokens-NoSentences	Medium	Beam Search	74.47	74.34	74.36
		Top-p Sampling	73.48	73.00	73.20
	Large	Beam Search	74.81	74.54	74.63
		Top-p Sampling	73.77	73.18	73.43
NoSentences-RatioTokens	Medium	Beam Search	72.67	75.28	73.91
		Top-p Sampling	71.76	73.96	72.81
	Large	Beam Search	73.25	**75.51**	74.33
		Top-p Sampling	72.48	73.99	73.19
NoWords-NoSentences	Medium	Beam Search	73.98	74.71	74.30
		Top-p Sampling	72.94	73.21	73.04
	Large	Beam Search	74.43	74.71	74.52
		Top-p Sampling	73.66	73.47	73.52
NoSentences-NoWords	Medium	Beam Search	73.91	75.33	74.58
		Top-p Sampling	72.61	73.74	73.15
	Large	Beam Search	73.46	75.34	74.35
		Top-p Sampling	72.73	74.11	73.38
LexOverlap-NoWords	Medium	Beam Search	**75.05**	74.84	**74.90**
		Top-p Sampling	73.49	73.52	73.46
	Large	Beam Search	74.89	74.60	74.69
		Top-p Sampling	73.71	73.69	73.66
NoWords-LexOverlap		Top-p Sampling	73.64	73.76	73.66
	Large	Beam Search	74.81	74.59	74.65
		Top-p Sampling	73.53	73.56	73.50

3.4. Exploratory Analysis of Generated Summaries Using Control Token

Besides assessing the performance of various configurations, our aim was also to explore the extent to which control tokens change the generated texts. As such, we generated summaries for the same news by varying the values for the control token(s), while assessing the impact on the quality of the generated summary and its resemblance to the original text. Given the previous best results, medium and large RoGPT models with beam search configurations were chosen for this experiment. We experimented with an individual control token (i.e., <NoSenentences>) that is easily explainable, as well as with a more complex scenario that forces a compression/expansion of the generated text (i.e., a combination used of <NoSentences>-<NoWords>). The range for <NoSenentences> was 2–5; there were extremely few training samples with only 1 sentence within the summary, and our model is incapable of generating such over-condensed summaries. The <NoWords> control token considered five values −50%, −25%, 0%, +25%, +50%, which signified a compression of −50% words from the reference summary, all the way to an expansion with +50% additional words. A sample of 100 news articles from the test partition was chosen, and BERTScore F_1 was calculated for each value of the control token(s); the corresponding results are presented in Figures 2 and 3. An example of text generation when only the <NoSentence> was varied is presented in Appendix C.1, whereas Appendix C.2 showcases the example for <NoSentence>-<NoWords>.

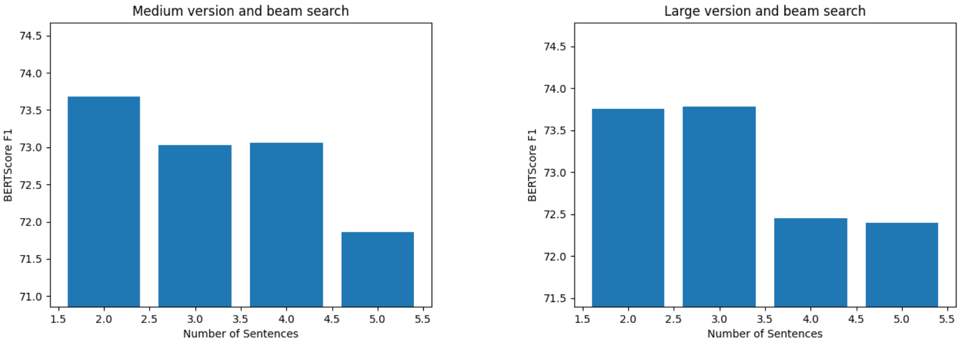

Figure 2. BERTScore for NoSentences.

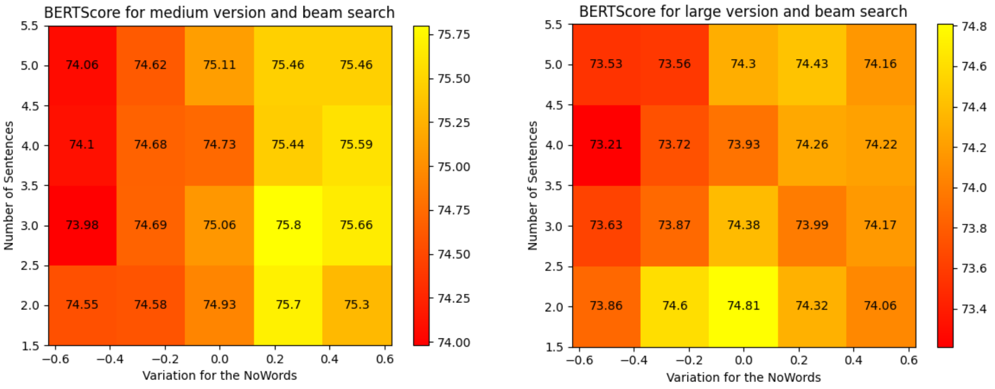

Figure 3. BERTScore for NoSentences-NoWords.

4. Discussion

The baseline model managed to achieve good results (see Table 1) for the summarization task, and the best results for ROUGE-L (34.67%) and BERTScore (74.34%) were obtained by the medium version with the beam search decoding method. It is worth noting that the best results were obtained with the beam search decoding method regardless of the considered model. Poorer results obtained by the large version are arguable, given the relatively small size of the dataset.

Results from the human evaluations (see Table 2) were also consistent, based on the obtained ICC3 score. The best score for the main idea was obtained by the large model with greedy decoding (3.73/4), followed by the medium version with beam search with a score of 3.43/4, thus arguing that the models managed to identify the main idea from the news. In terms of the provided details, the best score (3.36/4) was achieved by the medium model with beam search decoding (see Appendix A.1 for an example). The model managed to have coherent sentences with an elevated language; this was also shown in the paper that introduced RoGPT2 [10]. The large model obtained the highest overall score in terms of cohesion with greedy decoding (3.27/4), followed by the medium model with beam search with a score of 3.13/4; this lower score is justifiable since the contents of some randomly sampled news articles were challenging to summarize (see Appendix A.2 for a horoscope example). Paraphrasing was the main problem of the texts generated by the model since the models mostly repeated information from the reference text. Nevertheless, the results obtained by the model are impressive, considering that the human-evaluated news articles originated from a dataset on which the model was not trained.

The summaries using control tokens obtained better scores than the baseline summarization task (see Table 3). The small differences indicate that a winning configuration cannot be determined with certainty as the largest difference was up to 2%; however, we observed that beam search consistently obtained the best results. Despite being the most complex token, the largest improvement in BERTScore F_1 with 1.08% was obtained with the <LexOverlap> control token. The worst results for controlling text characteristics were obtained by <NoSentences>, whereas <RatioTokens> obtained a lower BERTScore than <NoWords> because it is a token more difficult to understand by the model.

Lower performance for combinations of tokens was expected because the dataset is relatively small and the task difficulty was higher. Then, comparing the performance of the models on each control token individually, we noticed that a higher performance was obtained for the second token specified in the prompt; this suggests that the model was influenced more by the second token from the prompt. The combination <NoWords>-<LexOverlap> obtained the best overall results, highlighting the benefits of complementarity between control tokens. Overall, the best decoding method was beam search.

When considering the exploratory analysis, the best results when varying the number of sentences were obtained for values of 2 and 3; this was expected as most summaries had 3 sentences. The example from Appendix C.1 highlights that the model seems to only extract sentences from the original text without paraphrasing. With <NoSentences> set at three, the model copied a central sentence and reiterated it based on a repetition present in the source text (i.e., the news article contained "Roxana Ispas este fondatoarea brandului Ronna Swimwear." and "Roxana Ispas, fondatoare Ronna Swimwear", which confused the model). Furthermore, there was a problem when setting the control token to 5 as the model failed to generate five sentences; nevertheless, it generated considerably longer sentences than the previous use case with only four sentences.

The best results for the experiment with the <NoSentences>-<NoWords> combination were obtained when the number of sentences was equal to 2 or 3 and the number of words was equal to +25% or +50% more words than the original summary. The best BERTScore was obtained for the medium version with <NoSentence> = 3 and <NoWords> = +25%, followed by a similar scenario with <NoSentences> = 2 and the same value for <NoWords>. As exemplified in Appendix C.2, the model takes into account the number of words that must be generated, i.e., there is a proportional relationship between the length of the summary and the value of the control token. Furthermore, a higher compression rate given by a smaller number of words forced the model to generate one less sentence than specified.

5. Conclusions

This paper introduced a novel dataset, a baseline model, and control tokens for manipulating text characteristics when summarizing texts in Romanian; all previous resources have been publicly released. Our model obtained overall good results (F1-scores above

0.73 in most configurations), indicating that the models learn even from limited samples. The generated texts were grammatically correct and primarily consistent, as highlighted by the human evaluation. Using control tokens led to the improvement of BERTScore [30]. The best results were obtained when using beam search as a decoding strategy, while medium and large models shared similar performances; however, the medium models are more suitable given the size of the dataset. Higher scores were obtained when only one control token was used. In contrast, the model emphasized the second token specified in the prompt when generating the text in complex scenarios.

In terms of future work, we aim to increase the quality and size of our dataset with examples originating from other news websites targeting specific fields in contrast to AlephNews, which is a generalist news site. This will ensure a higher diversity of text characteristics and introduce the possibility of new control tokens specific to the new categories. Moreover, we plan to register the summarization task in the LiRo benchmark [11] to ensure the development of robust natural-language-understanding systems for Romanian.

Author Contributions: Conceptualization, M.D. and S.R.; methodology, M.D., S.R. and M.A.N.; software, M.A.N. and S.R.; validation, M.A.N., S.R. and M.D.; formal analysis, S.R.; investigation, M.A.N. and S.R.; resources, M.A.N.; data curation, M.A.N.; writing—original draft preparation, M.A.N.; writing—review and editing, M.D. and S.R.; visualization, M.A.N.; supervision, M.D.; project administration, M.D.; funding acquisition, M.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the "Innovative Solution for Optimizing User Productivity through Multi-Modal Monitoring of Activity and Profiles – OPTIMIZE"/"Soluție Inovativă de Optimizare a Productivității Utilizatorilor prin Monitorizarea Multi-Modala a Activității și a Profilelor—OPTIMIZE" project, Contract Number 366/390042/27.09.2021, MySMIS code: 121491.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki and approved by the Ethics Committee of the Faculty of Automated Control and Computers, University Politehnica of Bucharest.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset for Romanian text summarization is freely available on HuggingFace (https://huggingface.co/datasets/readerbench/AlephNews; last accessed 20 October 2022); the models built on top of RoGPT-2 are available on HuggingFace (https://huggingface.co/readerbench/RoSummary-large; last accessed 20 October 2022); the corresponding code is released on GitHub (https://github.com/readerbench/RoSummary; last accessed 20 October 2022).

Acknowledgments: Special thanks to the TensorFlow Research Cloud (https://www.tensorflow.org/tfrc; last accessed on 20 October 2022) programs for providing us the Tensor Processing Unit (TPU) (https://cloud.google.com/tpu/; last accessed on 20 October 2022) that was used to train the models.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
ICC	Intraclass Correlation Coefficient
ROUGE	Recall Oriented Understudy for Gisting Evaluation
NLP	Natural Language Processing
MSE	Mean-Squared Error
MAE	Mean Absolute Error
NLG	Natural Language Generation
TPU	Tensor Processing Unit

Appendix A

Appendix A.1

News: "Zona Cheile Râşnoavei, sufocată de gunoaie Un telespectator Digi24 a trimis o sesizare la autorităţile locale, dar reprezentaţii primăriei ridică neputincioşi din umeri: au încercat să cureţe, dar romii din apropiere fac mizerie din nou, peste noapte. La câţiva metri de drumul care duce spre Cheile Râşnoavei, unul din cele mai frumoase locuri de vizitat din judeţ, frumuseţea peisajului este umbrită de gunoaiele aruncate pe o rază de mai bine de o sută de metri. Profitând de faptul că zona este marcată ca fiind poligon de trageri, oamenii au aruncat aici nestingheriţi saci întregi de gunoi, fără să se gândească la turiştii care trec pe aici sau la felul în care nepăsarea lor strică imaginea oraşului. Cunoscut pentru cetatea ţărănească şi pentru peisajele de poveste de pe Cheile Râşnoavei, oraşul Râşnov atrage la fiecare sfârşit de săptămână mii de turişti. Cei care ajung însă la periferie rămân dezamăgiţi: mormane întregi de sticle, hârtii, gunoi menajer şi chiar cărămizi s-au adunat aici. Iar în ultima perioadă, localnicii au început să ducă acolo chiar şI gunoaiele din grajduri. Ovidiu Drăgunoiu locuieşte în Braşov şi este un împătimit al drumeţiilor. Zilele trecute a vrut să meargă cu câinele la plimbare pe drumul spre Cheile Râşnoavei. A filmat mizeria şi a anunţat autorităţile. Luaţi la întrebări, reprezentaţii primăriei au găsit rapid o scuză. „Am făcut ordine acolo, am curăţat, am nivelat zona ca într-o săptămână zona să fie la loc. Vin cu căruţele, cu sacii, cu fel şI fel de modalităţi ei pun acolo din curte parcă intenţionat parcă sfidător să arate zona din nou murdară şi plină de mizerie", a declarat Marian Ilie, responsabil cu problemele de mediu la Primăria Râşnov. Locuitorii oraşului sunt si ei afectaţi de mormanele de gunoaie. Nu de puţine ori s-au îmbolnăvit animalele duse la păscut în zonă. Autorităţie din Râşnov spun că toate gunoaiele vor fi adunate, iar cei care nu le depozitează corespunzător vor fi amendaţi. Deocamdată însă, nimeni nu a primit nicio sancţiune."

Summary: Zona Cheile Râşnoavei, sufocată de gunoaie Un telespectator Digi24 a trimis o sesizare la autorităţile locale, dar reprezentaţii primăriei ridică neputincioşi din umeri: au încercat să cureţe, dar romii din apropiere fac mizerie din nou, peste noapte. La câţiva metri de drumul care duce spre Cheile Râşnoavei, unul din cele mai frumoase locuri de vizitat din judeţ, frumuseţea peisajului este umbrită de gunoaiele aruncate pe o rază de mai bine de o sută de metri. La câţiva metri de drumul care duce spre Cheile Râşnoavei, unul din cele mai frumoase locuri de vizitat din judeţ, frumuseţea peisajului este umbrită de gunoaiele aruncate pe o rază de mai bine de o sută de metri.

Appendix A.2

News: "HOROSCOP. Leii pleacă într-o vacanță mult-așteptată PEȘTI Atenție la cheltuieli. Nu cedați tentațiilor. BERBEC Se anunță o zi dinamică. Sunt favorizate activitățile intelectuale. TAUR Vă puteți întâlni cu persoane care vă îndeamnă la comunicare. GEMENI Sunteți nemulțumit de ritmul în care se desfășoară un proiect. Lucrurile vor reveni la normal. RAC Este posibil să primiți niște bani din colaborări mai vechi. LEU Puteți pleca într-o călătorie pe care o așteptați de mult timp. FECIOARĂ Petreceți o seară specială cu cei dragi. Primiți vizita copiilor. BALANȚĂ Sunteți foarte solicitați la birou. Aveți o serie de responsabilități. SCORPION Foarte implicați în relația de iubire, Scorpionii petrec o seară specială alături de partener. SAGETĂTOR Nu cumpărați tot ce vă iese în cale. Mai mult de jumătate dintre achiziții se vor dovedi inutile. CAPRICORN În aceste zile veți vedea rezultate concrete ale muncii dumneavoastră și veți avea ocazia să vă exprimați ideile. VĂRSĂTOR Ați putea primi o veste importantă, care vă reține la birou. Nu neglijați totuși, familia."

Summary: Berbecii pleacă într-o vacanță mult-așteptată PEȘTI Atenție la cheltuieli. Nu cedați tentațiilor.

Appendix B. Results for Control Tokens

Appendix B.1. Simple Scenarios

Table A1. Results for NoSentences (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore Recall (%)	F_1 (%)	MSE	MAE
Base	Greedy	73.15	72.87	72.92	3.630	0.920
	Beam Search	73.69	73.53	73.54	0.857	**0.661**
	Top-p Sampling	72.52	72.24	72.32	1.554	1.026
Medium	Greedy	73.54	74.08	73.74	0.996	0.814
	Beam Search	73.49	74.42	73.89	**0.813**	0.702
	Top-p Sampling	72.72	73.04	72.83	0.955	0.852
Large	Greedy	**73.96**	73.97	73.90	1.141	0.987
	Beam Search	73.90	**74.78**	**74.27**	0.989	0.870
	Top-p Sampling	73.34	72.99	73.11	1.168	1.001

Table A2. Results for NoWords (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore Recall (%)	F_1 (%)	MSE	MAE
Base	Greedy	73.73	73.24	73.43	257.15	9.28
	Beam Search	74.17	73.67	73.88	114.31	7.55
	Top-p Sampling	72.84	72.56	72.67	397.93	9.14
Medium	Greedy	74.34	74.04	74.15	529.18	8.11
	Beam Search	74.71	74.45	74.55	67.42	5.44
	Top-p Sampling	73.43	73.07	73.23	110.34	6.53
Large	Greedy	74.58	74.33	74.42	147.61	6.73
	Beam Search	**74.90**	**74.67**	**0.7475**	**51.87**	**4.96**
	Top-p Sampling	73.53	73.27	73.37	77.69	6.10

Table A3. Results for RatioTokens (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore Recall (%)	F_1 (%)	Spearman (%)	Pearson (%)
Base	Greedy	74.46	71.98	73.10	51.88	30.36
	Beam Search	74.81	72.48	73.55	59.44	58.09
	Top-p Sampling	73.22	71.59	72.32	54.48	40.58
Medium	Greedy	**75.47**	73.41	74.34	54.26	38.19
	Beam Search	75.45	73.41	74.34	62.06	63.23
	Top-p Sampling	74.11	72.49	73.22	55.08	53.41
Large	Greedy	74.12	74.37	74.21	90.03	55.80
	Beam Search	74.35	**74.66**	**74.48**	**93.17**	**88.23**
	Top-p Sampling	73.22	73.37	73.26	90.63	84.81

Table A4. Results for LexOverlap (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore Recall (%)	F_1 (%)	Spearman (%)	Pearson (%)
Base	Greedy	75.13	73.59	74.28	77.68	78.93
	Beam Search	75.62	74.32	74.89	72.74	69.37
	Top-p Sampling	73.48	73.17	73.27	80.38	84.65
Medium	Greedy	75.59	74.55	75.01	77.79	0.8074
	Beam Search	75.90	**74.94**	75.36	76.77	74.62
	Top-p Sampling	73.95	73.88	73.87	**81.78**	**86.68**
Large	Greedy	75.83	74.44	75.07	79.72	83.46
	Beam Search	74.37	73.83	74.05	79.84	80.29
	Top-p Sampling	**76.30**	74.66	**75.42**	80.74	86.11

Appendix B.2. Complex Scenarios

Table A5. Results for RatioTokens-NoSentences (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore	F_1 (%)	MSE	MAE	Spearman (%)	Pearson (%)
			Recall (%)		NoSentences	NoSentences	RatioTokens	RatioTokens
Base	Greedy	73.22	73.44	73.25	1.641	1.019	62.19	42.57
	Beam Search	73.77	74.07	73.86	1.002	0.714	66.71	66.01
	Top-p Sampling	72.56	72.98	72.72	2.678	1.316	64.29	49.99
Medium	Greedy	74.39	74.11	74.19	0.974	0.759	74.57	69.14
	Beam Search	74.47	74.34	74.36	**0.677**	**0.549**	**77.70**	**78.03**
	Top-p Sampling	73.48	73.00	73.20	1.123	0.808	77.02	68.41
Large	Greedy	74.59	74.17	74.33	0.685	0.555	74.86	72.91
	Beam Search	**74.81**	**74.54**	**74.63**	0.919	0.757	77.02	74.00
	Top-p Sampling	73.77	73.18	73.43	1.027	0.811	74.70	72.85

Table A6. Results for NoSentences-RatioTokens (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore	F_1 (%)	MSE	MAE	Spearman (%)	Pearson (%)
			Recall (%)		NoSentences	NoSentences	RatioTokens	RatioTokens
Base	Greedy	73.88	73.41	73.58	1.188	0.829	74.60	45.64
	Beam Search	**74.23**	73.89	74.01	**0.834**	**0.599**	78.80	75.22
	Top-p Sampling	72.87	72.87	72.83	1.774	1.036	76.88	73.45
Medium	Greedy	72.22	74.83	73.46	3.707	1.414	82.59	71.82
	Beam Search	72.67	75.28	73.91	1.888	1.087	86.45	78.72
	Top-p Sampling	71.76	73.96	72.81	3.408	1.537	84.84	81.96
Large	Greedy	72.84	74.77	73.75	2.368	1.309	87.39	76.25
	Beam Search	73.25	**75.51**	**74.33**	1.670	1.077	89.52	**85.69**
	Top-p Sampling	72.48	73.99	73.19	2.629	1.415	**89.53**	85.21

Table A7. Results for NoWords-NoSentences (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore	F_1 (%)	MSE	MAE	Spearman (%)	Pearson (%)
			Recall (%)		NoWords	NoWords	NoSentences	NoSentences
Base	Greedy	73.99	72.82	73.35	291.70	10.80	1.323	0.791
	Beam Search	74.28	73.46	73.82	**190.02**	**9.66**	**0.715**	**0.532**
	Top-p Sampling	73.01	72.18	72.55	196.83	10.29	1.270	0.875
Medium	Greedy	73.86	74.36	74.05	414.65	12.50	1.519	1.023
	Beam Search	73.98	**74.71**	74.30	201.53	11.03	0.905	0.714
	Top-p Sampling	72.94	73.21	73.04	232.68	11.69	1.586	1.077
Large	Greedy	74.28	74.26	74.21	294.37	12.43	1.156	0.890
	Beam Search	**74.43**	**74.71**	**74.52**	239.29	11.55	84.85	69.37
	Top-p Sampling	73.66	73.47	73.52	245.23	11.76	1.178	0.925

Table A8. Results for NoSentences-NoWords (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore Recall (%)	F_1 (%)	MSE NoWords	MAE NoWords	MSE NoSentences	MAE NoSentences
Base	Greedy	72.93	73.51	73.17	238.36	11.28	1.794	1.063
	Beam Search	73.30	74.16	73.70	160.75	9.75	**1.156**	**0.773**
	Top-p Sampling	72.10	72.86	72.44	226.73	11.79	2.284	1.224
Medium	Greedy	73.46	74.72	74.05	290.49	11.53	2.083	1.170
	Beam Search	**73.91**	75.33	**74.58**	**148.71**	**9.42**	1.263	0.837
	Top-p Sampling	72.61	73.74	73.15	229.48	11.64	2.517	1.290
Large	Greedy	73.33	74.97	74.09	383.55	14.77	4.283	1.226
	Beam Search	73.46	**75.34**	74.35	308.95	13.88	1.530	0.985
	Top-p Sampling	72.73	74.11	73.38	338.57	13.82	2.529	1.284

Table A9. Results for LexOverlap-NoWords (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore	F_1 (%)	MSE	MAE	Spearman (%)	Pearson (%)
			Recall (%)		NoWords	NoWords	LexOverlap	LexOverlap
Base	Greedy	74.06	73.86	73.89	577.91	15.65	0.548	0.578
	Beam Search	73.98	74.28	74.06	470.63	15.35	0.417	0.439
	Top-p Sampling	72.99	73.07	72.97	464.38	15.25	57.83	64.11
Medium	Greedy	74.76	74.42	74.54	257.78	11.61	**68.33**	**72.44**
	Beam Search	**75.05**	**74.84**	**74.90**	**224.48**	**1,0.99**	64.09	65.60
	Top-p Sampling	73.49	73.52	73.46	245.16	11.70	66.07	72.41
Large	Greedy	74.34	74.18	74.21	321.30	12.29	64.58	68.29
	Beam Search	74.89	74.60	74.69	398.75	12.14	60.77	61.18
	Top-p Sampling	73.71	73.69	73.66	289.60	12.36	65.06	69.99

Table A10. Results for NoWords-LexOverlap (bold marks the best results).

Model	Decode Method	Precision (%)	BERTScore	F_1 (%)	MSE	MAE	Spearman (%)	Pearson (%)
			Recall (%)		NoWords	NoWords	LexOverlap	LexOverlap
Base	Greedy	74.12	73.83	73.90	690.47	15.70	57.18	60.38
	Beam Search	74.06	74.21	74.06	629.78	15.61	42.64	44.49
	Top-p Sampling	72.85	73.02	72.88	436.36	15.30	58.94	66.40
Medium	Greedy	74.77	74.44	74.56	263.85	11.78	67.68	71.77
	Beam Search	**75.08**	**74.92**	**74.95**	**245.22**	**11.43**	64.16	65.61
	Top-p Sampling	73.64	73.76	73.66	272.39	12.23	**68.80**	**74.56**
Large	Greedy	74.25	74.10	74.13	277.87	12.25	63.87	67.61
	Beam Search	74.81	74.59	74.65	408.00	12.55	59.81	60.23
	Top-p Sampling	73.53	73.56	73.50	282.77	12.42	61.51	65.75

Appendix C. Summaries Generated While Varying Values for Control Token(s)

Appendix C.1. Summaries Generated with <NoSentences>

News: "O româncă a vândut costume de baie de lux în valoare de 2 milioane de euro în 2020. Cine a fost puțin creativ anul trecut a făcut bani frumoși. Roxana Ispas este fondatoarea brandului Ronna Swimwear.A lucrat mai mulți ani în domeniul juridic, apoi a avut un business în domeniul consultanței, iar acum s-a reprofilat. Face costume de baie de lux. A profitat de faptul că multe românce au mers anul trecut în vacanțe exotice și a făcut haine de plajă, sutiene și chiloți. Și nu îi merge rău deloc. Are comenzi din toată lumea și cliente fidele mai ales în Germania, Franța sau Dubai. În 2020 Roxana a avut încasări de 2 milioane de euro.Roxana Ispas, fondatoare Ronna Swimwear: P̈rofilul clientei–cea care vrea să arăte ca un milion dollar baby. Așa am vândut bine și în Madrid și la Monte Carlo și la Paris și Dubai. Online am vândut prima oară cel mai bine, dar anul trecut a fost jumate-jumate chiar. În 2021 sper să se dea drumul să călătorim, să fie mult mai ok". Și deja stăm foarte ok la capitolul vacanțe exotice, România se află pe locul 5 în topul țărilor care au trimis cei mai mulți turiști în Maldive. Doar în primele 10 zile din ianuarie 2021 aproape 1500 de turiști români au preferat această destinație. I-am depășit pe nemți, francezi, elvețieni și chiar pe americani."

Original: "Pandemia le-a adus celor îndrăzneți vânzări uriașe în 2020. Roxana vinde costume de baie de lux în Franța, Germania sau Dubai. Roxana a avut vânzări de 2 milioane de euro în 2020. România a trimis mai mulți turiști în Maldive decât SUA, Germania sau Elveția."

Generate for <NoSentences>=2: "Roxana Ispas este fondatoarea brandului Ronna Swimwear. Roxana a vândut costume de baie de lux în valoare de 2 milioane de euro în 2020."

Generate for <NoSentences>=3: "Roxana Ispas este fondatoarea brandului Ronna Swimwear. Roxana a vândut costume de baie de lux în valoare de 2 milioane de euro în 2020. Roxana Ispas este fondatoarea brandului Ronna Swimwear."

Generate for <NoSentences>=4: "Roxana Ispas este fondatoarea brandului Ronna Swimwear. Roxana a vândut anul trecut costume de baie de lux în valoare de 2 milioane de euro. Roxana Ispas: "Profilul clientei–cea care vrea să arăte ca un milion dollar baby. Așa am vândut bine și în Madrid și la Monte Carlo și la Paris"."

Generate for <NoSentences>=5: "Roxana Ispas este fondatoarea brandului Ronna Swimwear. Roxana a lucrat mai mulți ani în domeniul juridic, apoi a avut un business în domeniul consultanței. Roxana a vândut costume de baie de lux în valoare de 2 milioane de euro în 2020. Roxana Ispas: "Profilul clientei – cea care vrea să arăte ca un milion dollar baby. Așa am vândut bine și în Madrid și la Monte Carlo și la Paris"."

Appendix C.2. Summaries Generated with the Combination <NoSentences>-<NoWords>

News: "Studenţii de la UMF Craiova sunt voluntari în campania de vaccinare. Când se termină campania de vaccinare non-stop. Studenții de la Universitatea de Medicină și Farmacie din Craiova ajută la campania de vaccinare. Spun că doar așa se pot întoarce în sălile de curs.Ioana Deaconu e studentă în anul II. Vrea să devină chirurg, dar până acum n-a reușit deloc să facă practică în spital, din cauza restricțiilor anti pandemice. Acum, s-a înscris voluntar în campania de vaccinare non-stop. Împreună cu prietena ei, Amalia, îi ajută pe cei care vin la vaccinare să completeze formularele de triaj.Printre primii care au venit să se vaccineze au fost mai mulți străini. Ana este studentă în Spania, dar va locui o perioadă în România, unde a venit să lucreze ca voluntar la o organizație de luptă împotriva traficului de persoane. Spune că, atunci când a auzit că se poate vaccina fără programare, nu a stat pe gânduri.Peste 100 de studenți de la Medicină participă, timp de o săptămână, la campania de vaccinare non-stop de la stadionul Ion Oblemenco.Tudor Bălșanu, prodecanul Facultății de Medicină, spune că studenții nu sunt obligați să fie vaccinați pentru a putea participa la cursuri, dar riscurile scad atunci când sunt imunzați cât mai mulți.Campania de vaccinare non-stop din Craiova se va încheia pe 17 mai. Centrul are 10 fluxuri de vaccinare ziua și două noaptea."

Original: "Aproximativ 100 de studenți de la UMF Craiova sunt voluntari în campania de vaccinare. La Craiova, până în data de 17 mai, te poți vaccina non-stop, la stadionul Ion Oblemenco. Și străinii se pot vaccina, fără programare, la centrul deschis la stadionul Ion Oblemenco."

Generate for <NoSentences>=3 and <NoWords>-50%: "Studenții de la UMF Craiova ajută la campania de vaccinare non-stop. Ioana Deaconu este studentă în Spania, dar va locui o perioadă în România."

Generate for <NoSentences>=3 and <NoWords>-25%: "Studenții de la UMF Craiova sunt voluntari în campania de vaccinare non-stop. Ioana Deaconu este studentă în Spania, dar va locui o perioadă în România, unde a venit să lucreze ca voluntar la o organizație de luptă împotriva traficului de persoane."

Generate for <NoSentences>=3 and <NoWords> + 0%: "Studenții de la UMF Craiova sunt voluntari în campania de vaccinare non-stop. Ioana Deaconu, studentă în anul II, spune că doar așa se pot întoarce în sălile de curs. Peste 100 de studenți de la Medicină participă, timp de o săptămână, la campania de vaccinare non-stop de la stadionul Ion Oblemenco."

Generate for <NoSentences>=3 and <NoWords> + 25%: "Studenții de la UMF Craiova sunt voluntari în campania de vaccinare non-stop. Ioana Deaconu, studentă în anul II, spune că doar așa se pot întoarce în sălile de curs. Peste 100 de studenți de la Medicină participă, timp de o săptămână, la campania de vaccinare non-stop de la stadionul Ion Oblemenco. Campania de vaccinare non-stop din Craiova se va încheia pe 17 mai."

Generate for <NoSentences>=3 and <NoWords> + 50%: "Studenții de la Universitatea de Medicină și Farmacie din Craiova ajută la campania de vaccinare non-stop. Ioana Deaconu e studentă în anul II. Vrea să devină chirurg, dar până acum n-a reușit deloc să facă practică în spital, din cauza restricțiilor anti pandemice. Peste 100 de studenți de la Medicină participă, timp de o săptămână, la campania de vaccinare non-stop de la stadionul Ion Oblemenco."

References

1. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
2. Radford, A.; Wu, J.; Amodei, D.; Amodei, D.; Clark, J.; Brundage, M.; Sutskever, I. Better language models and their implications. *OpenAI Blog* **2019**, *1*, 2. Available online: https://openai.com/blog/better-language-models (accessed on 14 October 2022).
3. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
4. Black, S.; Gao, L.; Wang, P.; Leahy, C.; Biderman, S. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. *arXiv* **2020**, arXiv:2101.00027.
5. Wang, B.; Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. 2021. Available online: https://github.com/kingoflolz/mesh-transformer-jax (accessed on 14 October 2022).
6. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **2020**, *21*, 1–67.
7. Masala, M.; Ruseti, S.; Dascalu, M. Robert–a romanian bert model. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6626–6637.
8. Dumitrescu, S.D.; Avram, A.M.; Pyysalo, S. The birth of Romanian BERT. *arXiv* **2020**, arXiv:2009.08712.
9. Avram, A.M.; Catrina, D.; Cercel, D.C.; Dascălu, M.; Rebedea, T.; Păiş, V.; Tufiş, D. Distilling the Knowledge of Romanian BERTs Using Multiple Teachers. *arXiv* **2021**, arXiv:2112.12650.
10. Niculescu, M.A.; Ruseti, S.; Dascalu, M. RoGPT2: Romanian GPT2 for Text Generation. In Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA, 1–3 November 2021; pp. 1154–1161.
11. Dumitrescu, S.D.; Rebeja, P.; Lorincz, B.; Gaman, M.; Avram, A.; Ilie, M.; Pruteanu, A.; Stan, A.; Rosia, L.; Iacobescu, C.; et al. Liro: Benchmark and leaderboard for Romanian language tasks. In Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1), Online, 7–10 December 2021.
12. Nallapati, R.; Zhou, B.; Nogueira dos santos, C.; Gulcehre, C.; Xiang, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. *arXiv* **2016**, arXiv:1602.06023.
13. Scialom, T.; Dray, P.A.; Lamprier, S.; Piwowarski, B.; Staiano, J. MLSUM: The multilingual summarization corpus. *arXiv* **2020**, arXiv:2004.14900.
14. Narayan, S.; Cohen, S.B.; Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv* **2018**, arXiv:1808.08745.
15. Völske, M.; Potthast, M.; Syed, S.; Stein, B. Tl; dr: Mining Reddit to learn automatic summarization. In Proceedings of the Workshop on New Frontiers in Summarization, Copenhagen, Denmark, 7 September 2017; pp. 59–63.
16. Cioaca, V.; Dascalu, M.; McNamara, D.S. Extractive Summarization using Cohesion Network Analysis and Submodular Set Functions. In Proceedings of the 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2020), Timisoara, Romania, 1–4 September 2020.
17. Dutulescu, A.; Ruseti, S.; Dascalu, M. Unsupervised Extractive Summarization with BERT. In Proceedings of the 24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2022), Linz, Austria, 12–15 September 2022.
18. Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; Huang, X.J. Extractive Summarization as Text Matching. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 6–8 July 2020; pp. 6197–6208.
19. Liu, Y.; Lapata, M. Text Summarization with Pretrained Encoders. 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), Hong Kong, China, 3–7 November 2019; pp. 3730–3740.
20. Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; Zettlemoyer, L. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 6–8 July 2020; pp. 7871–7880.
21. Liu, Y.; Liu, P.; Radev, D.; Neubig, G. BRIO: Bringing Order to Abstractive Summarization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 2890–2903.
22. Qi, W.; Yan, Y.; Gong, Y.; Liu, D.; Duan, N.; Chen, J.; Zhang, R.; Zhou, M. ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 2401–2410.
23. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *Proc. Int. Conf. Mach. Learn.* **2020**, *119*, 11328–11339.
24. Martin, L.; Fan, A.; de la Clergerie, É.; Bordes, A.; Sagot, B. MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. *arXiv* **2020**, arXiv:2005.00352.
25. Clive, J.; Cao, K.; Rei, M. Control prefixes for text generation. *arXiv* **2021**, arXiv:2110.08329.
26. Kieuvongngam, V.; Tan, B.; Niu, Y. Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2. *arXiv* **2020**, arXiv:2006.01997.

27. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadelphia, PE, USA, 7–12 July 2002; pp. 311–318.
28. Lin, C. Recall-oriented understudy for gisting evaluation (rouge). *Retrieved August* **2005**, *20*, 2005.
29. Banerjee, S.; Lavie, A. 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, Ann Arbor, MI, USA, 25 June 2005; pp. 65–72.
30. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. Bertscore: Evaluating text generation with BERT. *arXiv* **2019**, arXiv:1904.09675.
31. Yuan, W.; Neubig, G.; Liu, P. Bartscore: Evaluating generated text as text generation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 27263–27277.
32. Sellam, T.; Das, D.; Parikh, A.P. BLEURT: Learning robust metrics for text generation. *arXiv* **2020**, arXiv:2004.04696.
33. Celikyilmaz, A.; Clark, E.; Gao, J. Evaluation of text generation: A survey. *arXiv* **2020**, arXiv:2006.14799.
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
35. Freitag, M.; Al-Onaizan, Y. Beam search strategies for neural machine translation. *arXiv* **2017**, arXiv:1702.01806.
36. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The curious case of neural text degeneration. *arXiv* **2019**, arXiv:1904.09751.
37. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
38. Taylor, L. Assessing Reading-Into-Writing Skills for an Academic Context: Some Theoretical and Practical Considerations. 2013. Available online: https://github.com/kingoflolz/mesh-transformer-jax (accessed on 14 October 2022).
39. Westby, C.; Culatta, B.; Lawrence, B.; Hall-Kenyon, K. Summarizing expository texts. *Top. Lang. Disord.* **2010**, *30*, 275–287. [CrossRef]
40. Koo, T.K.; Li, M.Y. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *J. Chiropr. Med.* **2016**, *15*, 155–163. [CrossRef]