BART-IT: An Efficient Sequence-to-Sequence Model for Italian Text Summarization

future internet

MDPI

Article

BART-IT: An Efficient Sequence-to-Sequence Model for Italian Text Summarization

Moreno La Quatra * and Luca Cagliero

Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi, 24-10129 Torino, Italy

* Correspondence: moreno.laquatra@polito.it; Tel.: +39-011-090-7179

Abstract: The emergence of attention-based architectures has led to significant improvements in the performance of neural sequence-to-sequence models for text summarization. Although these models have proved to be effective in summarizing English-written documents, their portability to other languages is limited thus leaving plenty of room for improvement. In this paper, we present BART-IT, a sequence-to-sequence model, based on the BART architecture that is specifically tailored to the Italian language. The model is pre-trained on a large corpus of Italian-written pieces of text to learn language-specific features and then fine-tuned on several benchmark datasets established for abstractive summarization. The experimental results show that BART-IT outperforms other state-of-the-art models in terms of ROUGE scores in spite of a significantly smaller number of parameters. The use of BART-IT can foster the development of interesting NLP applications for the Italian language. Beyond releasing the model to the research community to foster further research and applications, we also discuss the ethical implications behind the use of abstractive summarization models.

Keywords: sequence-to-sequence models; Italian language; text summarization

Citation: La Quatra, M.; Cagliero, L. BART-IT: An Efficient Sequence-to-Sequence Model for Italian Text Summarization. *Future Internet* **2023**, *15*, 15. https://doi.org/10.3390/fi15010015

Academic Editors: Massimo Cafaro, Italo Epicoco and Marco Pulimeno

Received: 30 November 2022
Revised: 19 December 2022
Accepted: 20 December 2022
Published: 27 December 2022

1. Introduction

Automatic Text Summarization (ATS) is a natural language processing task that consists of creating a shorter version of a text document, which is coherent and maintains the most relevant information of the original text [1]. Text summarization techniques are usually divided into two main categories: *extractive* and *abstractive summarization*. In extractive summarization, the summary is created by selecting a subset of the original text and thus the output text is composed of sentences that are present in the original document. Abstractive summarization, on the other hand, generates the output summary, which does not necessarily include sentences in the original document. Abstractive summarization is usually considered more challenging than extractive summarization as it entails creating new text containing coherent and summarized content. Due to the inherent complexity of the abstractive summarization task, it is often applied on top of an extractive summarization process to refine the previously selected content [1].

In recent years, ATS has received a lot of attention as it can be applied to a wide range of applications such as the extraction of highlights from scientific papers [2], the generation of summaries of news articles [3], and the creation of multimodal summaries of audio podcasts [4]. The summarization task can be also instrumental for other NLP tasks, e.g., by reducing the size of large documents to make them more suitable for downstream tasks [5].

Transformer-based architectures have shown to be effective in modeling long-range dependencies and have already been applied to several NLP tasks such as machine translation [6] and question answering [7]. BART [8] is a sequence-to-sequence model based on the Transformer architecture, which is trained using a denoising objective to learn effective representations of the input text that can be used for a wide range of downstream tasks. The model takes as input a sequence of tokens and generates a sequence of tokens as output. It

can be exploited to solve several tasks including machine translation, abstractive question answering, and text summarization. The effectiveness of BART has been demonstrated both in English and in multilingual scenarios. However, limited efforts have been devoted to exploring its potential in language-specific, non-English scenarios.

In this paper, we present *BART-IT*, a sequence-to-sequence model, based on the BART architecture that is specifically tailored to the Italian language. To learn effective language representations, BART-IT is first trained on a large corpus of Italian documents and then fine-tuned on several datasets for text summarization. To evaluate BART-IT, we consider both single-language and multilingual models that have previously been applied to the Italian language [9–11]. The experimental results show that the BART-IT summarization performance is superior to that achieved by models with a comparable (or even higher) number of parameters. Performance comparable to that of larger multilingual models makes BART-IT competitive, especially in resource-constrained scenarios [12].

To foster further research in the field of sequence-to-sequence models for the Italian language, we release the pre-trained and fine-tuned BART-IT models as well as the code used to train and evaluate BART-IT (details on the public repository are given in the *Data availability Statement*.). Since the unconscious use of deep learning models for text abstraction is potentially subject to bias and unfairness, in this paper, we also discuss the ethical implications of using deep learning models for generating text summaries.

The main contributions of this work is the release of a new Italian language model, which can be used to solve a range of NLP tasks, its fine-tuning for abstractive text summarization, and testing on benchmark data. More specifically,

- We present BART-IT, a sequence-to-sequence BART-based model specifically designed for the Italian language (see Section 3);
- We release the model pre-trained on a large Italian corpus. It can be used to solve various NLP tasks on Italian documents (see the *Data Availability Statement*);
- We fine-tune BART-IT for the abstractive text summarization task (see Section 4.3);
- We assess model performance on Italian benchmark data, showing that BART-IT achieves the best balancing between efficiency and effectiveness compared to various baselines (see Figure 1);
- We discuss the ethical aspects behind the practical use of the BART-IT model (see Section 5).

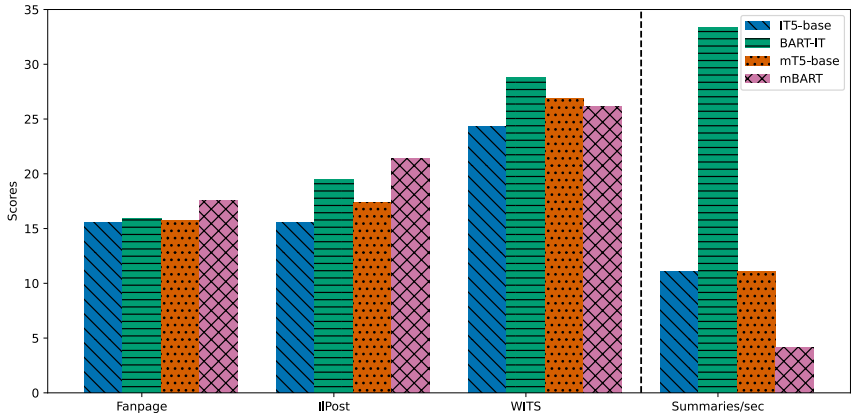

Figure 1. ROUGE-2 comparison between IT5, BART-IT, mT5, and mBART on benchmark datasets. The last four bars represent the performance of the models in terms of summaries/second on a single NVIDIA A6000 GPU.

2. Related Works

Abstractive text summarization aims at generating fluent and coherent summaries that are able to accurately convey the main ideas of the original text. Unlike extractive text summarization [13], in which the summary is generated by selecting relevant sentences

from the original text, abstractive summarization entails understanding the meaning of the raw input text and generating a new summary that is grammatically consistent and semantically coherent.

In recent years, deep learning methods have made significant progress, achieving state-of-the-art results in several NLP tasks, including text summarization [14,15]. The generalization capabilities of deep learning models and their ability to automatically extract high-level features from text data have allowed them to outperform traditional methods in many NLP tasks. In the field of abstractive text summarization, sequence-to-sequence models [16] have gained significant popularity. Thanks to their ability to encode the input text into internal representations, sequence-to-sequence models can be used to condition the generation process to produce a coherent summary.

Recurrent Neural Networks (RNNs) [17,18] have been exploited to solve sequence-to-sequence tasks since the early days of neural networks. These architectures have shown to be effective in modeling sequential data and successfully applied to abstractive summarization [19,20]. However, the training of RNNs can become very slow, especially while coping with long sequences because RNNs must be unfolded to model the temporal dependencies in the sequence. In addition, the long-term dependencies can be modeled very poorly due to the vanishing gradient problem [21].

Transformer-based models [6] mitigate the aforesaid issues by using the self-attention mechanism. It allows the network to explicitly attend to all the words in the sequence, regardless of their position in the sequence. They not only substantially improve the quality of sequential data models but also gain efficiency thanks to parallel computation. Transformers have shown to be particularly effective in addressing the abstractive summarization task [8,14] and are thus the backbone of state-of-the-art models.

Most transformer-based models for abstractive text summarization rely on the established encoder–decoder architecture. The encoder is a bidirectional transformer and is responsible for acquiring contextual information from the raw input text. The decoder is also a transformer and is trained to generate the summary by attending both to the encoder output and to the previously generated summary tokens. BART [8] is among the best-performing transformer-based abstractive summarization models. It is trained using a denoising autoencoder objective, where the input sequence is corrupted using different transformations, such as token permutations and token deletions. The training objective forces the model to learn to reconstruct the original sequence by attending to the contextual information and by generating the correct sequences.

Alternative transformer-based models for abstractive text summarization adopt similar architectures. For example, T5 [22] is a transformer-based model that is trained using multi-task learning. The key idea is to train the model on multiple tasks at once, thus transferring useful knowledge from one task to another. This is particularly beneficial for achieving a high level of generality of the trained models, as they are exposed to different text types. PEGASUS [14] is a transformer-based model trained relying on the *Gap Sentences Generation* and *Masked Language Modeling* objectives. By learning to generate sequences and predict masked tokens, PEGASUS acquires multiple skills that are deemed as beneficial in the training phase.

Italian Language Modeling

Most of the existing transformer-based language models are available only for the English language. This strongly limits the portability to multilingual scenarios. To overcome the above-mentioned limitation, prior works have attempted to train transformer-based models trained in languages other than English such as French [23], Vietnamese [24], or Chinese [25]. Limited research efforts have been devoted to the Italian language, i.e., [9,26–28]. Refs. [26–28] are encoder-only models. They encode pieces of text according to an internal representation that can be used for discriminative tasks such as text classification. However, they cannot be applied to solve generative tasks such as abstractive text summarization. IT5 [9] is a sequence-to-sequence model. Similar to BART, it can be used to generate

text conditioned on the input text. However, IT5 is trained using a multi-task learning objective, which offers a different training objective than BART and thus may show different performance on the text summarization task. In this work, we aim at developing a transformer-based model specifically designed for the Italian language and trained using the same training objective as BART [8]. We also carry out an empirical comparison between BART-IT and IT5 to analyze model performance on Italian documents.

3. BART-IT

BART-IT is a sequence-to-sequence transformer-based model based on BART [8]. It is specifically designed for the Italian language and exploits the same denoising objectives used for the training of BART.

3.1. Denoising Objectives

To learn effective representation for the summarization task, the original BART model employs multiple denoising objectives. The input text is first corrupted using different transformations and then the model is trained to reconstruct the original text. The corruption transformations used in BART are the following:

- *Document rotation:* the input document is first divided into sentences using full stops as separators. Then, one sentence is randomly selected as the first sentence and the remaining sentences follow the selected one using the original order of the input document.
- *Sentence permutation:* Similar to the previous case, the input document is first divided into sentences using full stops as separators. Then, the sentences are randomly shuffled and the resulting sequence is used as input for the model. In this case, no sentence of the corrupted sequence is forced to follow the order of the original document.
- *Token infilling:* a span length L is randomly selected using a Poisson distribution with lambda parameter $\lambda = 3$. Then, each span of length L is replaced by a single `[MASK]` token with a probability of 0.15. The remaining spans are left unchanged.
- *Token masking:* similar to the original BERT [7] pretraining objective, each token is replaced by a `[MASK]` token with a probability of 0.15. In contrast to *Token infilling*, only one token at a time is replaced by a `[MASK]` token. Similar to *Token infilling*, the remaining tokens are not modified.
- *Token deletion:* each token is randomly deleted (with a probability of 0.15) from the input sequence. The remaining tokens remain unchanged.

Table 1 shows examples of the different denoising objectives. Each corruption is applied to the original input sequence and the associated corrupted sequence is shown. The choice of the corruption transformations is random, and the model is trained to reconstruct the original sequence using the corrupted sequence as input. While each corruption is applied independently and only one corruption is applied at a time, during training, the model is exposed to all possible corruptions of the input sequence, thus allowing the model to learn to generalize on different noise patterns. The use of several corruptions having different granularities allows the model to better learn the different aspects of the input text, such as the token, sentence, and document structure.

Table 1. Examples of the different denoising objectives. Capital letters represent sentences and numbers represent tokens. Each corruption is applied to the original sequence, and the resulting corrupted sequence is shown. The model is trained to reconstruct the original sequence using the corrupted sequence as input. Selected tokens or sentences are highlighted using a yellow background.

Corruption	Original Sequence	Corrupted Sequence
Document Rotation	A. B. C. D.	C. D. A. B.
Sentence permutation	A. B. C. D.	C. A. D. B.
Token infilling	1 2 3 4 5 6 7	1 [MASK] 5 6 7
Token masking	1 2 3 4 5	1 2 3 [MASK] 5
Token deletion	1 2 3 4 5	1 2 3 5

3.2. Model Architecture

BART-IT model follows the same architecture as BART [8]. BART-IT is trained from scratch in the Italian language and uses a language-specific tokenizer created using the Byte–Pair Encoding (BPE) algorithm [29]. We use the base architecture of BART to create an efficient transformer-based model that is both able to learn effective representations and can be trained on a single GPU with a reasonable amount of memory and time. BART-IT is a sequence-to-sequence model composed of an encoder and a decoder. Both the encoder and the decoder are composed of 12 layers, the number of attention heads is 12, and the hidden size of the internal representation is 768. To effectively learn language-specific representations, we also train a tokenizer for the Italian language using the Byte–Pair Encoding (BPE) [29] algorithm. The tokenizer is trained on the same data collection used for the training of BART-IT, and it is used to tokenize the input text before feeding it to the model. Table 2 reports the full list of parameters used for the training of BART-IT and the corresponding tokenizer. The resulting model has a total of 140 million parameters, which is comparable to the number of parameters of the base version of BART [8].

Table 2. Training parameters used for BART-IT and the corresponding tokenizer.

Parameter	Value
Model parameters	
# encoder layers	12
# decoder layers	12
# attention heads	12
Hidden size	768
Feed-forward size	3072
Tokenizer parameters	
Vocab size	52,000
Min frequency	10
Special tokens	<s>, </s>, <pad>, <unk>, <mask>

3.3. Training Data Collection

Modern transformer-based models are generally pre-trained using large corpora to learn the syntactic and semantic properties of the language. With the goal of learning effective representations of the Italian language, we train BART-IT using a large collection of Italian documents. Specifically, we use the *Clean Italian mC4 Corpus* used for training IT5 [9]. It is a cleaned version of the multilingual Colossal Clean Crawled Corpus (mC4) [11] containing only Italian documents. The dataset is cleaned to remove noisy or corrupted text that could negatively affect the training of sequence-to-sequence models. Using the same data collection allows us to compare the performance of BART-IT with the performance of IT5 [9] on the abstractive summarization task. The final data collection consists of approximately 103 million documents and 41 billion words.

4. Experiments

In this section, we present the datasets used for fine-tuning BART-IT for the abstractive summarization task, the evaluation metrics, the experimental setup, and describe the achieved results.

4.1. Fine-Tuning Datasets

Abstractive summarization is a challenging natural language processing task because it entails conveying the key information contained in a long piece of text into a short summary. Most of the benchmarks available for this task consist of a collection of text documents and the corresponding manually written summaries. In our experiments, we fine-tune BART-IT on three different Italian summarization datasets to evaluate its performance on the abstractive summarization task.

- **FanPage** [30] is a dataset of Italian news articles from the online newspaper Fanpage.it (https://www.fanpage.it/, latest access: 24 December 2022). It includes 84,365 news articles as well as their corresponding summaries. The dataset is split into a training set of 67,492 documents, a validation set of 8436 documents, and a test set of 8437 documents. The average length of the documents is approximately 312 words and the average length of the summaries is approximately 43 words.
- **IlPost** [30], similar to FanPage, is a dataset of Italian news articles from the online newspaper IlPost.it (https://www.ilpost.it/, latest access: 24 December 2022). It contains a total of 44,001 article-summary pairs divided into a training set of 35,201 documents, a validation set of 4,400 documents, and a test set of 4400 documents. The average length of the documents is shorter than FanPage, with an average length of approximately 174 words, while the average length of the summaries is approximately 26 words.
- **WITS** [31] is a dataset of Wikipedia articles and their corresponding summaries. The dataset is automatically generated by crawling the Italian Wikipedia (https://it.wikipedia.org/, latest access: 24 December 2022) and extracting the leading section of each Wikipedia article and using it as the summary. The dataset is the largest among the three, containing a total of 700,000 article-summary pairs. Analogously to the original authors, we randomly select 10,000 articles as the test set, 10,000 articles as the validation set, and the remaining articles are used for training. Given the different nature of the articles, the average length of the documents is 956.66 words, and the average length of the summaries is 70.93 words.

Document type, documents' average length, and summaries' average length are rather diversified across the considered benchmarks, making them suitable for evaluating the performance of BART-IT under multiple aspects.

We use the same fine-tuning procedure for all the datasets, using a maximum sequence length of 1024 tokens for the input documents (i.e., longer documents are truncated) and a maximum sequence length of 128 tokens for the summaries (i.e., longer summaries are truncated).

4.2. Evaluation Metrics

The evaluation of abstractive summarization models is usually performed by comparing the generated summaries with the corresponding reference summaries. The most common evaluation metric is the *ROUGE* [32] score, which is a set of metrics that measure the similarity between the generated and the reference summaries by comparing the number of n-grams (i.e., sequences of n words) that they have in common. In this work, we use the ROUGE-1, ROUGE-2, and ROUGE-L scores, which measure the similarity between the generated and reference summaries by comparing the number of unigrams, bigrams, and the longest common subsequence, respectively.

The ROUGE score cannot capture *semantic similarity* between the generated and reference summaries. To overcome this limitation, we also use the *BERTScore* [33] metric, which is a state-of-the-art evaluator that measures the token-level semantic similarity between the

generated and the reference summaries. Unlike the ROUGE score, the BERTScore metric computes the cosine similarity between the embeddings of the tokens in the generated and the reference summaries. These embeddings are usually generated using a pre-trained encoder model, i.e., in our case, we use the multilingual cased version of BERT [7] as suggested by the authors of the BERTScore metric. The use of this metric allows us to capture the semantic similarity between the generated and the reference summaries and to better evaluate the quality of summaries that are not extracted from the input documents but are generated and thus may contain words that do not appear in the input documents.

4.3. Experimental Setup

The pre-training of large transformer-based models is a computationally expensive task that requires both a large amount of text data and ample computation resources. To train the BART-IT model, we use a machine with the following characteristics:

- **CPU**: Intel® Core™ i9-10980XE CPU @ 3.00 GHz;
- **GPU**: 2 x NVIDIA® RTX A6000 GPU, with 48 GB of VRAM each
- **RAM**: 128 GB.

The training of the model is performed using the `transformers` library [34] and leveraging the `PyTorch` framework for deep learning [35].

Pre-training phase

For the pre-training phase of BART-IT, we use a total batch size of 64, a maximum sequence length of 1024 tokens for both the input and the output sequences, and the AdamW optimizer [36] with a maximum learning rate of 10^{-4} and a weight decay of 10^{-2}. The pre-training phase is performed for 1.7 million steps (i.e., roughly 1 epoch as suggested in recent studies [37]) with 17,000 warmup steps. We use a linear scheduler for the learning rate, which means that the learning rate starts at 0 and linearly increases to reach the value of 10^{-4} over the first 17,000 training steps. After the warmup phase is complete, the learning rate starts to decrease according to the decay factor. This means that the optimizer will take smaller and smaller steps as training progresses, which can help to improve the convergence of the training process and the performance of the trained model. To reduce both training time and memory usage, we also use floating-point 16-bit precision during model pre-training.

Fine-tuning phase

To ensure consistency and fairness in our comparison of model performance, fine-tune BART-IT using the same set of hyperparameters for all the datasets. This allowed us to directly compare the results and evaluate the effectiveness of our model without introducing any potential biases or inconsistencies. Specifically, we use a batch size equal to 32, a maximum sequence length of 1024 tokens for the input documents, and a maximum sequence length of 128 tokens for the summaries. The fine-tuning step is performed using AdamW optimizer [36] for a maximum of 10 epochs, with a maximum learning rate of 10^{-5}, 500 warmup steps, and a weight decay of 10^{-2}. We also use floating-point 16-bit precision during model fine-tuning. These parameters were chosen to fit the computational constraints of our experimental setup and are used for all datasets. The best model is selected using the ROUGE-2 score on the validation set.

4.4. Baseline Models

To evaluate the effectiveness of BART-IT, we compare it against the following strong baselines for *abstractive* summarization:

- **IT5** [9] is a state-of-the-art sequence-to-sequence model that relies on the same architecture proposed by T5 [22] but is trained on the Italian language. This model is trained on the same dataset used for the pre-training of BART-IT and is fine-tuned on the same summarization datasets used for its evaluation. The model is available in

three different sizes: *small*, *base*, and *large*. We use the *base* version of the model since it is the most similar in terms of the number of parameters compared to the proposed model (i.e., 220 million parameters);

- **mBART** [10] is a multilingual sequence-to-sequence model that uses the same architecture of the original BART model. It is trained on a multilingual corpus of 25 languages. By construction, the model size for the base model is more than four times larger than the model size of BART-IT (i.e., 610 million parameters). Even in this case, the model is fine-tuned on the same summarization datasets used for the evaluation of BART-IT;
- **mT5** [11] is a multilingual sequence-to-sequence model that uses the same architecture of T5 [22] and is trained on a multilingual corpus of 101 languages. The model is available in five different sizes: *small*, *base*, *large*, *xlarge*, and *xxlarge*. We use the *base* version of the model that has 390 million parameters (e.g., more than 2.5 times larger than the model size of BART-IT). Similar to the previous models, the model is fine-tuned on the same datasets used for the evaluation of BART-IT.

The summarization models are also compared against well-established *extractive* summarization baselines to evaluate the effectiveness of the abstractive models in generating summaries that are more fluent and coherent than the extractive summaries.

- **Lead-K** is a baseline method that extracts the first k sentences of the input document. In our experiments, we set k to 2;
- **LexRank** [38] is an established summarization method that extracts the most important sentences from the input document by first modeling the input document as a graph and then computing the PageRank score [39] of each sentence. The similarity between two sentences is computed using the IDF-weighted cosine similarity between the TF-IDF vectors of the sentences;
- **TextRank** [40] is a baseline method that, similar to LexRank, extracts the most important sentences of the input document by modeling the input document as a graph and computing the PageRank score [39] of each sentence. The pairwise sentence similarity is computed by exploiting the word-level overlap between each pair of sentences.

All the models are evaluated using both the ROUGE and BERTScore metrics to evaluate the effectiveness of the models in generating fluent and coherent summaries.

4.5. Experimental Results

This section outlines the main experimental results achieved on the analyzed datasets used during the evaluation phase. To analyze both model efficiency and effectiveness, we report the result obtained on the test set and the number of parameters separately for each model. The model size can be used as a proxy for the complexity of the model and as a way to gauge the performance of models of different sizes. Tables 3–5 separately report the results of both extractive and abstractive summarization models on the three different datasets using ROUGE and BERTScore metrics.

Table 3. Summarization evaluation results on the FanPage dataset. The # parameters column is empty for unsupervised baseline models since they do not include any trainable parameters.

Model	# Parameters	R1	R2	RL	BERTScore
LEAD-2	×	31.88	14.11	21.68	70.84
TextRank	×	26.39	9.06	16.81	68.63
LexRank	×	29.85	11.69	19.58	69.9
IT5-base	220 M	33.99	15.59	24.91	70.3
BART-IT	140 M	35.42	15.88	25.12	73.24
mT5	390 M	34.13	15.76	24.84	72.77
mBART	610 M	36.52	17.52	26.14	73.4

Table 4. Summarization evaluation results on the IlPost dataset. The # parameters column is empty for unsupervised baseline models since they do not include any trainable parameters.

Model	# Parameters	R1	R2	RL	BERTScore
LEAD-2	×	27.72	11.66	19.62	70.25
TextRank	×	22.63	8.0	15.48	68.4
LexRank	×	26.96	10.94	18.94	69.94
IT5-base	220 M	32.88	15.53	26.7	71.06
BART-IT	140 M	37.31	19.44	30.41	75.36
mT5	390 M	35.04	17.41	28.68	74.69
mBART	610 M	38.91	21.41	32.08	75.86

Table 5. Summarization evaluation results on the WITS dataset. The # parameters column is empty for unsupervised baseline models since they do not include any trainable parameters.

Model	# Parameters	R1	R2	RL	BERTScore
LEAD-2	×	15.63	3.32	10.51	63.01
TextRank	×	15.35	3.04	9.84	62.12
LexRank	×	15.96	3.3	10.48	62.81
IT5-base	220 M	37.98	24.32	34.94	77.14
BART-IT	140 M	42.32	28.83	38.84	79.28
mT5	390 M	40.6	26.9	37.43	80.73
mBART	610 M	39.32	26.18	35.9	78.65

News summarization datasets

FanPage and IlPost [30] are two datasets containing Italian news articles and their corresponding summaries. Both datasets can be classified as *medium-sized* datasets since they contain less than 100,000 of documents in the training, validation, and test sets. BART-IT significantly outperforms all extractive baseline models and IT5 with comparable model sizes on both datasets. Tables 3 and 4 show the comparison between BART-IT and the other models on the two datasets. Regarding ROUGE-1 and ROUGE-2 scores, BART-IT outperforms IT5 by a margin of 1.43 and 0.29 points (i.e., 4.2% and 1.8% relative improvement) on FanPage and 4.43 and 3.91 points (i.e., 13.4% and 25.17% relative improvement) on IlPost. In terms of BERTScore, BART-IT outperforms IT5 by a margin of 2.93 (4.2%) and 4.3 (6.11%) points on FanPage and IlPost, respectively. When comparing BART-IT with multilingual models, we observe that BART-IT can perform better than mT5, but it is outperformed by mBART on both datasets at the cost of significantly more computationally intensive model training. More specifically, both mBART and mT5 models are trained on a multilingual corpus and contain significantly more parameters than BART-IT (e.g., 610 M parameters for mBART and 390 M parameters for mT5 compared to 140M parameters for BART-IT), which can be a reason for the performance gap between the models.

Wikipedia Summarization Dataset

WITS [31] is a large dataset of Wikipedia articles and their corresponding summaries. The dataset contains roughly 700,000 documents and has been used in previous works to evaluate the performance of abstractive summarization models [31]. Despite the fact that having a large number of documents can be beneficial for training accurate models, the Wikipedia articles are rather diversified in length and topic, making the task of summarization significantly more challenging. Analyzing the results on the WITS dataset reported in Table 5, we observe that BART-IT performs significantly better than all the other models, including the multilingual ones. Even though the number of parameters of BART-IT is significantly fewer than both mBART and mT5, the model is able to outperform them in terms of ROUGE metrics. Considering the BERTScore metric, mT5 performs slightly better than BART-IT. The results indicate that multilingual models are able to learn the mapping

between source and target languages, but they might be less effective in learning to produce a summary of long and diverse documents.

The empirical analyses reported in this section show that BART-IT is competitive yet efficient in tackling the Italian document summarization task, especially when compared with language-specific abstractive models. The model also outperforms the IT5 sequence-to-sequence model with a similar number of parameters and is competitive against multilingual models having significantly larger model sizes.

Figure 1 compares the performance of the models in terms of a combined evaluation score mixing effectiveness (evaluated using the syntactic ROUGE-2 score) and efficiency (defined by the number of summaries generated per second on a single NVIDIA A6000 GPU). BART-IT achieves the best performance mix, addressing summary generation more efficiently than all the other tested models, outperforming IT5 and mT5 in terms of ROUGE-2 scores, and being competitive against mBART. In view of the achieved results, BART-IT can be deemed as an efficient yet effective solution for tackling the abstractive summarization task for the Italian language.

5. Discussions of Model Limitations and Ethical Issues

In this section, we address the limitations of the proposed approach and the ethical aspects related to the use of abstractive models. Sequence-to-sequence models are able to generate fluent and coherent text conditioned on the input document. Despite their characteristics being particularly suitable for addressing tasks such as summarization or machine translation, abstractive models are also prone to generate hallucinated text that may contain both non-factual information or offensive content [41]. In this work, we focused on the summarization task and assess the performance of BART-IT using automatic evaluation metrics. The adopted metric provides an estimate of the quality of the generated summaries, which neither demonstrates a factual correctness of the generated summaries nor prevents the inclusion of offensive/inappropriate content. Ensuring the factual correctness of the generated summaries is a challenging task that has been investigated by the NLP community [42] and is well beyond the scope of this work to the current work. The Italian NLP community would benefit from a pre-trained BART model on the Italian language to address several tasks, including but not limited to the summarization task. Hence, it is crucial to be aware of the ethical implications of using such models.

We provide BART-IT as an open-source model in a single model size, which is significantly smaller than the multilingual models (e.g., 140 million parameters vs. 610 million parameters for mBART). Researchers and practitioners can experiment with this model and evaluate its performance on different tasks. The model size allows them to fine-tune the model even on a single consumer GPU, which can be a significant advantage for researchers and practitioners that do not have access to large computational resources. However, although the quality of the generated summaries is competitive with models having significantly more parameters, it would be interesting to investigate if the performance of BART-IT can be further improved by training a larger version of the model. Training a larger version of BART-IT can be easily accomplished by using the same training procedure and leveraging the code used in the current work (see Section 3). However, the higher number of parameters would require a larger amount of computational resources. For example, to pre-train the large version of the model proposed by the original BART authors [8], on the same dataset as in the current work and in a reasonable amount of time (e.g., less than 30 days), it would require a system configuration having at least double the number of GPUs and amount of associated memory. The rest of the system configuration would be similar to the one used in the current work. This is beyond the scope of the current work.

6. Conclusions

In this work, we presented BART-IT, an efficient and effective sequence-to-sequence model for the Italian language. We pre-train BART-IT on a large Italian corpus and evalu-

ated its performance on the summarization task. The results show that BART-IT is able to outperform the language-specific abstractive models achieving competitive performance against multilingual models with a significantly larger number of parameters.

We present and release the BART-IT pre-trained model (as well as the code needed to retrain it) to foster the Italian Natural Language Processing community to develop new applications tailored to the Italian language. In spite of the fact that it is focused on the summarization task, we believe that the proposed model can be used for other tasks such as question answering and machine translation. Investigating the performance of BART-IT on other tasks will be addressed as future work.

Author Contributions: Conceptualization, M.L.Q. and L.C.; methodology, M.L.Q.; software, M.L.Q.; validation, M.L.Q.; investigation, M.L.Q.; resources, L.C.; data curation, M.L.Q.; writing—original draft preparation, M.L.Q.; writing—review and editing, M.L.Q. and L.C.; supervision, L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: https://github.com/MorenoLaQuatra/bart-it (latest access: 24 December 2022). The repository contains the code and details on how to reproduce the results of the paper.

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

References

1. El-Kassas, W.S.; Salama, C.R.; Rafea, A.A.; Mohamed, H.K. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.* **2021**, *165*, 113679.
2. La Quatra, M.; Cagliero, L. Transformer-based highlights extraction from scientific papers. *Knowl.-Based Syst.* **2022**, *252*, 109382.
3. Duan, Z.; Lu, L.; Yang, W.; Wang, J.; Wang, Y. An Abstract Summarization Method Combining Global Topics. *Appl. Sci.* **2022**, *12*, 10378.
4. Vaiani, L.; La Quatra, M.; Cagliero, L.; Garza, P. Leveraging multimodal content for podcast summarization. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, Virtual, 25–29 April 2022; pp. 863–870.
5. Inoue, N.; Trivedi, H.; Sinha, S.; Balasubramanian, N.; Inui, K. Summarize-then-Answer: Generating Concise Explanations for Multi-hop Reading Comprehension. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics: Online and Punta Cana, Dominican Republic, Virtual, 7–11 November 2021; pp. 6064–6080. [CrossRef]
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*; pp. 6000—6010.
7. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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), Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2019; pp. 4171–4186. [CrossRef]
8. 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, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2020; pp. 7871–7880. [CrossRef]
9. Sarti, G.; Nissim, M. IT5: Large-scale Text-to-text Pretraining for Italian Language Understanding and Generation. *arXiv* **2022**, arXiv:2203.03759.
10. Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual Denoising Pre-training for Neural Machine Translation. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 726–742. [CrossRef]
11. Xue, L.; Constant, N.; Roberts, A.; Kale, M.; Al-Rfou, R.; Siddhant, A.; Barua, A.; Raffel, C. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2021; pp. 483–498. [CrossRef]
12. Li, Z.; Wang, Z.; Tan, M.; Nallapati, R.; Bhatia, P.; Arnold, A.; Xiang, B.; Roth, D. DQ-BART: Efficient Sequence-to-Sequence Model via Joint Distillation and Quantization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022. [CrossRef]
13. Abdel-Salam, S.; Rafea, A. Performance Study on Extractive Text Summarization Using BERT Models. *Information* **2022**, *13*, 67.

14. Zhang, J.; Zhao, Y.; Saleh, M.; Liu, P. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the International Conference on Machine Learning. PMLR, Virtual, 13–18 July 2020; pp. 11328–11339.
15. Xiao, W.; Beltagy, I.; Carenini, G.; Cohan, A. PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2022; pp. 5245–5263. [CrossRef]
16. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*; pp. 3104—3112.
17. Rumelhart, D.; Hinton, G.; Williams, R. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*; MIT Press: Cambridge, MA, USA, 1986; pp. 318–362.
18. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
19. Nallapati, R.; Zhai, F.; Zhou, B. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
20. See, A.; Liu, P.J.; Manning, C.D. Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2017; pp. 1073–1083. [CrossRef]
21. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.* **1998**, *6*, 107–116.
22. 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*, 5485–5551.
23. Kamal Eddine, M.; Tixier, A.; Vazirgiannis, M. BARThez: a Skilled Pretrained French Sequence-to-Sequence Model. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2021; pp. 9369–9390. [CrossRef]
24. Tran, N.L.; Le, D.M.; Nguyen, D.Q. BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese. In Proceedings of the 23rd Annual Conference of the International Speech Communication Association, Incheon, South Korea, 18–22 September 2022.
25. Shao, Y.; Geng, Z.; Liu, Y.; Dai, J.; Yang, F.; Zhe, L.; Bao, H.; Qiu, X. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv* **2021**, arXiv:2109.05729.
26. Schweter, S. Italian BERT and ELECTRA models. *Zenodo* **2020**. Available online: https://doi.org/10.5281/zenodo.4263142 (accessed on 29 November 2022).
27. Polignano, M.; Basile, P.; De Gemmis, M.; Semeraro, G.; Basile, V. Alberto: Italian BERT language understanding model for NLP challenging tasks based on tweets. In Proceedings of the 6th Italian Conference on Computational Linguistics, CLiC-it 2019, CEUR, Bari, Italy, 13–15 November 2019; Volume 2481, pp. 1–6.
28. Guarasci, R.; Minutolo, A.; Damiano, E.; De Pietro, G.; Fujita, H.; Esposito, M. ELECTRA for neural coreference resolution in Italian. *IEEE Access* **2021**, *9*, 115643–115654.
29. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2016; pp. 1715–1725. [CrossRef]
30. Landro, N.; Gallo, I.; La Grassa, R.; Federici, E. Two New Datasets for Italian-Language Abstractive Text Summarization. *Information* **2022**, *13*, 228.
31. Casola, S.; Lavelli, A. WITS: Wikipedia for Italian Text Summarization. In Proceedings of the CLiC-it, Milan, Italy, 26–28 January 2022.
32. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of Summaries. In Proceedings of the Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2004; pp. 74–81.
33. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
34. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2020; pp. 38–45. [CrossRef]
35. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*; pp 8026-–8037.
36. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
37. Hernandez, D.; Brown, T.; Conerly, T.; DasSarma, N.; Drain, D.; El-Showk, S.; Elhage, N.; Hatfield-Dodds, Z.; Henighan, T.; Hume, T.; et al. Scaling Laws and Interpretability of Learning from Repeated Data. *arXiv* **2022**, arXiv:2205.10487.
38. Erkan, G.; Radev, D.R. LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization. *J. Artif. Int. Res.* **2004**, *22*, 457–479.

39. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report; Stanford InfoLab: Stanford, CA, USA, 1999.
40. Mihalcea, R.; Tarau, P. TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2004; pp. 404–411.
41. Cao, M.; Dong, Y.; Cheung, J. Hallucinated but Factual! Inspecting the Factuality of Hallucinations in 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; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2022; pp. 3340–3354. [CrossRef]
42. Zhou, C.; Neubig, G.; Gu, J.; Diab, M.; Guzmán, F.; Zettlemoyer, L.; Ghazvininejad, M. Detecting Hallucinated Content in Conditional Neural Sequence Generation. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online Event, 1–6 August 2021; Association for Computational Linguistics: Stroudsburg, Pennsylvania, PA, USA, 2021; pp. 1393–1404. [CrossRef]