Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1640–1650
Marseille, 20-25 June 2022

Generating Extended and Multilingual Summaries with Pre-trained Transformers

Rémi Calizzano, Malte Ostendorff, Qian Ruan, Georg Rehm
DFKI GmbH, Alt-Moabit 91c, 10559 Berlin, Germany
{remi.calizzano, malte.ostendorff, qian.ruan, georg.rehm}@dfki.de

Abstract

Almost all summarisation methods and datasets focus on a single language and short summaries. We introduce a new dataset called WikinewsSum for English, German, French, Spanish, Portuguese, Polish, and Italian summarisation tailored for extended summaries of approx. 11 sentences. The dataset comprises 39,626 summaries which are news articles from Wikinews and their sources. We compare three multilingual transformer models for extractive summarisation and three training scenarios on which we fine-tune mT5 to perform abstractive summarisation. This results in strong baselines for both extractive and abstractive summarisation on WikinewsSum. We also show how the combination of an extractive model with an abstractive one can be used to create extended abstractive summaries from long input documents. Our results show that fine-tuning mT5 on all the languages combined significantly improves the summarisation performance on low-resource languages.

Keywords: Summarisation, Multilingualism, Extended summarisation, Dataset

1. Introduction

Summarisation is a well-known Natural Language Processing (NLP) task. Recently, the quality of automatically generated summaries increased with the appearance of the transformer architecture (Vaswani et al., 2017) and the development of pre-trained sentence-to-sentence models such as BART (Lewis et al., 2019), T5 (Raffel et al., 2019), and Pegasus (Zhang et al., 2020a). These models improve the state-of-the-art in various tasks including summarisation on the CNN/Daily Mail (CNN-DM) dataset (Hermann et al., 2015), for example. However, different types of input content, and different desired types and lengths of the output summary, call for different methods altogether, or different parameter settings when using the same method.

This paper focuses on generating relatively long summaries for multiple languages. In contrast to popular datasets for English summarisation (Hermann et al., 2015; Over and Liggett, 2002; Narayan et al., 2018; Graff et al., 2003; Grusky et al., 2018) or multilingual summarisation (Scialom et al., 2020; Hasan et al., 2021; Ladhak et al., 2020), in which the average length of summaries ranges between one and five sentences (Dernoncourt et al., 2018), we aim to generate longer summaries of around 11 sentences. Throughout our paper, we will use the term *extended summaries* to refer to summaries with a length of approximately eleven sentences. Thus, we call the task multilingual, extended text summarisation. Only a few summarisation datasets are tailored to generate summaries of this length. Considering our task and focus, Multi-News (Fabbri et al., 2019) and WikiSum (Liu et al., 2018) are the most suitable datasets. However, to the best of our knowledge, there is currently no public non-English dataset that is suitable for our purpose. To fill this gap, we make use of Wikinews to create the first multilingual summarisation dataset tailored to extended summaries.

Our contributions are as follows. First, we make available the WikinewsSum dataset, covering English, German, French, Spanish, Portuguese, Polish, and Italian, designed for generating extended summaries. Second, we provide strong baselines for the dataset for both extractive and abstractive summarisation. We compare three multilingual transformer models (mBERT (Devlin et al., 2019), DistilmBERT (Sanh et al., 2019), and XLM-RoBERTa (Conneau et al., 2020)) for extractive summarisation using the method by Miller (2019). We also compare three training scenarios inspired by Hu et al. (2020) on which we fine-tune mT5 (Xue et al., 2021), a multilingual sequence-to-sequence transformer model, to perform abstractive summarisation. As the input documents of WikinewsSum are too long to perform abstractive summarisation on them directly, we use a hybrid two-stage text summarisation system which consists of firstly extracting the most relevant 512 tokens from the input documents using extractive summarisation and secondly perform abstractive summarisation using these 512 tokens as input of the abstractive model. The research presented in this paper was carried out under the umbrella of the project QURATOR (Rehm et al., 2020), in which novel AI-based technologies for the automated curation of digital content are being developed.

2. Related Work

2.1. Related Datasets

CNN-DM (Hermann et al., 2015) is the most used summarisation dataset (See et al., 2017; Paulus et al., 2018; Liu and Lapata, 2019b). The dataset is composed of news articles from CNN and Daily-Mail where the content of the article is the document to summarise and the highlight of the article is the target summary. In Zhang and Wan (2017), the authors use the 100 longest Wikinews articles to create a multi-document summarisation dataset. XSum (Narayan et al., 2018) is another news summarisation dataset but with very short summaries of one sentence. WikiSum (Liu et al., 2018) is one of the first multi-document summarisation dataset on which neural models can be trained. It is based on Wikipedia[1] articles and uses the headline as the summary and the references of the article, in addition to Google's top results with the title, as the documents to summarise. WikiSum has 10^6 examples with extended summaries and a high level of abstractiveness. Multi-News (Fabbri et al., 2019) is another multi-document dataset based on Newser[2], a news aggregation website. Summaries are handwritten articles and input documents are the sources of the article.

With regard to multilingual summarisation datasets, their emergence is relatively recent and came with the growing interest in the multilingual topics in NLP with the release of various multilingual models. Ladhak et al. (2020) introduces WikiLingua, a dataset for cross-lingual and multilingual abstractive summarisation. The authors use WikiHow to create the abstractive summarisation dataset for 17 languages and then align the non-English samples to the English ones to create the cross-lingual dataset. MLSUM (Scialom et al., 2020) has been created using online newspapers in French, German, Spanish, Russian, and Turkish. It is very comparable to CNN-DM due to the usage of online newspapers to create the dataset, and also due to its articles and summaries lengths. It contains a large number of samples with more than 1.5M article/summary pairs. XL-Sum Hasan et al. (2021) uses BBC[3] news to create a large multilingual dataset with more than 1M samples covering 44 languages. It focuses on summaries of one or two sentences. These datasets focus on short summaries, and to the best of our knowledge, there is no multilingual summarisation dataset that is suitable to use in our extended summary setup.

2.2. Related Methods

In extractive summarisation, the task is to rank and select the set of sentences maximising some specific metric, where usually ROUGE is used (metric detailed in Section 5.5). Various methods exist to perform extractive summarisation. The more recent ones use pre-trained transformer models in different ways. One approach proposed by Liu (2019) consists of adding a summarisation layer on top of BERT (Devlin et al., 2019) to classify each input sentence as part of the extractive summary or not. This approach is not suitable when the input text to summarise is longer than the maximum input length of the transformer model used. In Miller (2019), the author proposes another approach which consists of representing each sentence separately using a transformer model and then applying a clustering method to create the extractive summary. The main advantage of this method is that it is independent of the length of the input text and can therefore be used for very long input documents. There are also other approaches like the one presented in Ruan et al. (2022) which consists of taking advantage of the hierarchical structure of the input document to perform extractive summarisation.

In abstractive summarisation, the task is to generate the sentences that end up in the summary (as opposed to extracting them as they are from the source document). Sentence-to-sentence models have shown their use in this task (Liu and Lapata, 2019a; Parida and Motlicek, 2019; Nallapati et al., 2016), where especially those having a transformer architecture have been proven to be successful. As Liu et al. (2018) showed, a modified transformer decoder architecture can generate Wikipedia articles by summarising long sequences better than an LSTM encoder-decoder with attention (Bahdanau et al., 2015). Furthermore, pre-trained transformer models became popular since the release of BERT (Devlin et al., 2019). Encoder-decoder transformer models which have been pre-trained on several tasks like BART (Lewis et al., 2019), T5 (Raffel et al., 2019), and Pegasus (Zhang et al., 2020a), obtained state-of-the-art results on the summarisation tasks on various datasets. These models are especially efficient because of their ability to be fine-tuned with relatively few examples. mBART (Liu et al., 2020) and mT5 (Xue et al., 2021), the multilingual versions of BART and T5, have recently been released and already proved their efficiency on multilingual summarisation datasets (Ladhak et al., 2020; Hasan et al., 2021). However, these abstractive models are not able to process long input texts (more than 1024 or 512 tokens) which makes them not suitable in our extended summary setup. To tackle this issue, Liu et al. (2018) and Liu and Lapata (2019a) propose to combine an extractive model with an abstractive one. In this setup, the extractive model selects the important sentences that the abstractive model then uses to generate the summary from. This method is also known as hybrid text summarisation. Another method to tackle long input text in summarization has been proposed in Beltagy et al. (2020; Zaheer et al. (2020). Longformer (Beltagy et al., 2020) and Big Bird (Zaheer et al., 2020) are two transformer models that accept thousands of tokens or longer and up to 4096 tokens respectively. The two models replaced the full attention mechanism of the original transformer architecture. Longformer employs an attention pattern that combines local and global information and Big Bird uses a sparse attention mechanism. Longformer and Big Bird obtain good results on the arXiv summarization dataset (Cohan et al., 2018) which focuses on long document summarization in the scientific domain. However, both models have been pre-trained only on English texts and no multilingual version of these models exists.

[1] https://en.wikipedia.org

[2] https://www.newser.com

[3] https://www.bbc.com

3. WikinewsSum

We introduce a new multilingual extended summarisation dataset based on Wikinews.[4] Wikinews is a collaborative news website and part of the Wikimedia Foundation.[5] Each news article on the website is handwritten by a member of the community, thereby respecting the neutral point of view policy.[6] We interpret the articles as summaries and the source texts as input texts for those summaries. This approach is not completely new and extends the work initiated in Zhang and Wan (2017) in two ways. First, we use the English, German, French, Spanish, Portuguese, Polish, and Italian versions of Wikinews to create a multilingual dataset. Second, we extract a number of articles that is large enough for the training of neural models instead of only using the dataset for evaluation. The creation of a summarisation dataset using Wikinews is also inspired by Liu et al. (2018), who used Wikipedia instead of Wikinews articles. As a result, we have the WikinewsSum dataset consisting of article titles, article texts which are the summaries, and source texts which are the documents to be summarised. We construct the dataset using the English, German, French, Spanish, Portuguese, Polish, and Italian versions of Wikinews and are considering extending it to other languages of Wikinews as well. To create the dataset, we obtain all news articles from the Wikinews dump, extract the corresponding source links and obtain the source text from the link. If the source website is unavailable, we use the Wayback Machine from the Web archive.[7] The data is available online in the European Language Grid.[8] The code to reproduce the datasets is available on GitHub.[9]

The WikinewsSum dataset is described in Table 1. The number of samples (39,626) is much lower than for other summarisation datasets like XL-Sum (Hasan et al., 2021) (1,005,292) or CNN-DM (Hermann et al., 2015) (311,971) due to the limited number of articles in Wikinews and the creation process. The creation process removes many samples because many sources of the articles are not available anymore, and because of the strict sample selection policy we applied. For example, a sample needs to contain input documents with at least 1.5 more characters than the summary, and the summary and the input documents need to not be too short or too long. All the filters used are available on GitHub.[10] This results in fewer samples in the WikinewsSum dataset than the number of articles in Wikinews for the respective languages. This is especially the case for the Italian language where we only have 95 samples while there are more than 11,000 Italian Wikinews articles.

We perform ROUGE statistics (metric detailed in Section 5.5) of WikinewsSum (see Table 2). The ROUGE 1 and 2 recall scores show the percentages of the uni- and bi-grams of the summary contained in respectively the input documents, and the two pre-abstractive extractive steps (methods described Section 4.1). This evaluates the difficulty to reproduce the summary based on the respective inputs. We can compare the ROUGE recall scores of WikinewsSum with the ones of the CNN-DM and the MultiNews datasets which respectively have a ROUGE 1 recall score of 80.5 and 82.24 and a ROUGE 2 recall score of 43.12 and 42.9. We notice that the scores for CNN-DM and MultiNews are higher than for WikinewsSum which means that WikinewsSum's summaries contain less uni- and bi-grams from the input documents than CNN-DM's and MultiNews's summaries, and therefore than the summarisation task is more difficult for WikinewsSum.

4. Methodology

We use three extractive models and one abstractive model. One downside of the abstractive model we use is, that it has a maximum input length of 512 tokens due to the positional encoding in the underlying language models. Because WikinewsSum has longer input texts, we take inspiration from Liu et al. (2018) and Liu and Lapata (2019a), and experiment with a hybrid combination of an extractive model with an abstractive model. The extractive model is used to perform a pre-abstractive extractive step which consists of selecting 512 tokens from the input documents. These 512 tokens are used, in a second step, as input of the abstractive model which generates the abstractive summary. We use this hybrid text summarisation method during the training and the evaluation of the abstractive model.

4.1. Extractive Models

Extractive summarisation methods generate a summary as an ordered set of the most important input sentences. Multiple methods exist to perform extractive summarisation (see Section 2.2). We use the method by Miller (2019) which has the advantages of working regardless of the input length and of being easy to use due to the provided implementation.[11] The method uses a transformer model to obtain a representation of each sentence from the input documents and creates the extractive summary using K-Means clustering to identify the sentences closest to the centroid. Therefore, we can compare different transformer models to see which one creates the best representations to create the summary. We compare three multilingual transformer models:

[4] https://www.wikinews.org

[5] https://wikimediafoundation.org

[6] https://en.wikinews.org/wiki/Wikinews:Neutral_point_of_view

[7] https://web.archive.org

[8] https://live.european-language-grid.eu/catalogue/corpus/18633

[9] https://github.com/airKlizz/WikinewsSum

[10] https://github.com/airKlizz/WikinewsSum/blob/main/src/filter.py

[11] https://github.com/dmmiller612/bert-extractive-summariser

Languages	# samples	# cross-lingual samples	Input Documents		Summaries	
			# words	# sentences	# words	# sentences
English	11,616	641 (5.5%)	1,466	57	300	13
German	8,126	2,796 (34.4%)	1,179	58	241	13
French	6,200	1,932 (31.2%)	884	29	176	7
Spanish	6,116	2,137 (34.9%)	1,215	42	276	10
Portuguese	3,843	1,971 (51.3%)	1,037	38	221	8
Polish	3,630	1,214 (33.4%)	734	35	173	10
Italian	95	46 (48.4%)	1,021	35	224	8
All languages	39,626	10,737 (27.1%)	1,168	47	245	11

Table 1: Comparison of each language in the WikinewsSum dataset with regard to the number of samples, to the number of cross-lingual samples, and to the length of the input documents and the summaries.

Languages	Input Documents		Oracle pre-Abstractive Extractive step		mBERT pre-Abstractive Extractive step	
Metrics	R-1 R	R-2 R	R-1 R	R-2 R	R-1 R	R-2 R
English	74.98	36.75	59.62	30.47	50.96	18.43
German	59.06	22.38	46.11	19.25	37.98	11.42
French	61.3	29.56	51.56	25.68	45.64	19.13
Spanish	62.39	30.79	50.78	24.79	44.41	16.57
Portuguese	57.12	32.45	48.36	28.39	41.61	19.85
Polish	49.92	24.67	42.68	22.12	35.85	15.26
Italian	66.98	28.28	53.82	24.31	47.58	16.53
All languages	63.58	30.32	51.56	25.64	44.17	16.76

Table 2: Comparison of the Input Documents, and the two pre-Abstractive Extractive steps (Oracle and mBERT) with regard to their ROUGE recall scores between them and the summaries.

mBERT Multilingual BERT (Devlin et al., 2019) is pre-trained on the Wikipedias of 104 languages using masked language modeling and next sentence prediction. mBERT is a multi-layer bidirectional transformer encoder that does not include the decoder part.

DistilmBERT The multilingual version of DistilBERT (Sanh et al., 2019) is the distilled version of mBERT instead of BERT for the original DistilBERT. However, it keeps the same principles. DistilmBERT has less parameters and is therefore more computational efficient than mBERT. DistilmBERT is pre-trained on the same corpus in a self-supervised fashion.

XLM-RoBERTa XLM-RoBERTa (Conneau et al., 2020) is the multilingual version of RoBERTa (Liu et al., 2019). It was trained on Common Crawl in 100 languages using masked language modeling.

In addition, we use another extractive method called Oracle which estimates the upper boundary of the performance. It uses the gold summary to extract the N more relevant sentences of the input documents by maximising the ROUGE-2 recall score between the gold summary and the extracted sentences that produces the extractive summary. We use this method during the fine-tuning of the abstractive models and for evaluating the performance of the abstractive models with an ideal extractive step. This method corresponds to the cheating method as described in Liu et al. (2018).

4.2. Abstractive Models

The abstractive step allows us to create an abstractive summary – a summary with new sentences and words – from the sentences selected by the extractive step before. In comparison to extractive summarisation, abstractive summarisation usually increases the structure and coherence in the created summary since its output is not just a sequence of independent sentences.

We use mT5 (Xue et al., 2021) to perform abstractive summarisation. mT5 is a multilingual variant of T5 (Raffel et al., 2020) covering 101 languages. It uses the same architecture as T5, an encoder-decoder transformer model. As T5, mT5 exists in five sizes: Small, Base, Large, XL, XXL. The XXL version of mT5 performs the best on many multilingual benchmarks; due to computational limits, we use the mT5 Base version.

We fine-tune mT5 in three different scenarios specific to a multilingual dataset. These scenarios are extracted from Hu et al. (2020). Due to the nature of WikinewsSum which contains different samples for each language, some of the training scenarios of the XTREME paper (Hu et al., 2020) were not reproducible.

Cross-lingual zero-shot transfer We fine-tune mT5 on the English samples only and evaluate the resulting model on all languages. Our goal is to see if the model is able to transfer from English to the other languages without being explicitly trained on the other languages.

In-language multi-task We fine-tune mT5 on all available samples, which results in training mT5 on the English, German, French, Spanish, Portuguese, Polish, and Italian samples, all shuffled. We want to investigate if this multilingual training improves the performance of the model compared to in-language training.

In-language We fine-tune mT5 on each language separately. Hence, we train one model per language, which yields seven models for WikinewsSum in total. The model for English corresponds to the model trained in the cross-lingual zero-shot transfer scenario.

5. Experiments

5.1. Dataset

We split the WikinewsSum dataset into a train, a validation, and a test portions. We use 70% of the original dataset for the fine-tuning of mT5, 20% for the evaluation during the training, and 10% for testing the models and obtaining the final results.

5.2. Model Implementation

For the extractive step, we use the bert-extractive-summariser library[12] which generalizes the implementation of the method descibed in Miller (2019). We replace BERT as the embedding model with the three extractive models presented in Section 4.1. To use and fine-tune mT5, we rely on the Hugginface's library transformers (Wolf et al., 2019) which provides an easy-to-use implementation of the pre-trained mT5 model. We use the Base version of all four models.

5.3. Fine-tuning

The extractive models do not require fine-tuning since only their embeddings are used.

We fine-tune the abstractive model mT5 according to the three training scenarios presented in the section 4.2. As mT5 has a maximum input length of 512 tokens and as the input documents exceed this limit in WikinewsSum, we extract the most relevant 512 tokens in terms of ROUGE scores using the Oracle extractive method following the hybrid text summarisation setup. These 512 tokens are used as input during the fine-tuning of mT5 in the different training scenarios. For all the trainings, we use the same training parameters: the cross entropy loss, a batch size of 8, 8 epochs, a learning rate of $5.6 * 10^{-5}$ with a weight decay of 0.01. We make our code publicly available on GitHub.[13]

5.4. Evaluation

During the evaluation, the summaries generated by the extractive models contain 11 sentences which is the average number of sentences in summaries in WikinewsSum (see Table 1). During the text generation with abstractive models, we use beam search with a beam size of 5, and remove duplicated tri-grams. Minimum and maximum output lengths are set to 200 and 512 which fits more than 75% of the WikinewsSum's summaries. We conduct three evaluations. First, we evaluate our extractive models (*Extractive Summarisation*; Section 6.1). Secondly, we evaluate the abstractive models with the pre-abstractive step performed by the Oracle extractive method (*Abstractive Summarisation after Oracle Pre-Abstractive Extractive Step*; Section 6.2). This scenario is not applicable in a real world use-case because it uses Oracle which relies on the gold standard to extract the most relevant 512 tokens from the input documents. However, it provides an estimate about the theoretical best possible scores. Also, we evaluate the abstractive models independently of the extractive methods, i.e. with an ideal extractive method. Thirdly, we evaluate the abstractive models as before but with using the mBERT model to perform the extractive step (*Abstractive Summarisation after mBERT Pre-Abstractive Extractive Step*; Section 6.3). We use mBERT because it is the extractive model that obtains the best results (see Table 3). Table 2 shows the difference between the ROUGE recall scores of the Oracle and the mBERT extractive methods. This evaluation gives abstractive summarisation results that can be obtained in a real world use-case compared to the previous evaluation. All the evaluations are performed for the all WikinewsSum dataset but also for each language separately. Indeed we want to see if the results differ from one language to the others.

5.5. Metric

We use ROUGE (Lin, 2004) to evaluate our results. This metric measures the quality of a summary based on the number of over-lapping uni-grams (ROUGE-1 – R-1), bi-grams (ROUGE-2 – R-2), and the longest common sub-sequence (ROUGE-L – R-L) between the generated summary and the gold summary. We use the implementation of Huggingface.[14] As ROUGE has known limitations for abstractive summarisation evaluation (Schluter, 2017), we will also provide a manual evaluation of a few examples. [15]

6. Results

6.1. Extractive Summarisation

The three extractive models obtained similar results (see Table 3). mBERT is slightly better than DistilmBERT and XLM-RoBERTa but by a very small margin (+0.09 R-1, +0.17 R-2, +0.06 R-L). The ROUGE scores are relatively good compared to the Oracle method which is the theoretical optimum but not usable in practice extractive method. There is still progress that can be achieved but the extractive method provided

[12] https://github.com/dmmiller612/bert-extractive-summariser

[13] https://github.com/airKlizz/mdmls

[14] https://github.com/huggingface/nlp/blob/master/metrics/rouge/rouge.py

[15] We also applied the BERTScore metric; the results are coherent with the ROUGE scores, cf. Table 6.

Methods	Metrics	English	German	French	Spanish	Portuguese	Polish	Italian	All Languages
Extractive Summarisation									
DistilmBERT	R-1 F	41.37	29.37	29.80	29.70	29.62	24.83	35.18	33.51
	R-2 F	14.35	8.42	12.57	12.52	14.33	10.48	12.59	12.34
	R-L F	19.66	13.65	17.10	17.07	18.75	15.03	18.43	17.30
mBERT	R-1 F	41.37	29.74	29.74	35.50	29.66	24.82	34.93	**33.60**
	R-2 F	14.48	8.70	12.62	13.31	14.51	10.55	12.68	**12.51**
	R-L F	19.63	13.83	17.13	18.10	18.86	15.07	18.86	**17.36**
XLM-RoBERTa	R-1 F	40.92	29.00	29.70	35.40	29.39	24.74	35.68	33.27
	R-2 F	14.22	8.33	12.52	13.03	14.13	10.49	12.54	12.26
	R-L F	19.66	13.54	17.07	18.05	18.43	15.03	19.54	17.26
Oracle	R-1 F	49.50	37.21	34.41	42.24	35.32	29.89	41.85	40.29
	R-2 F	25.72	15.77	17.31	20.89	21.40	15.72	19.94	20.35
	R-L F	22.67	15.93	17.38	20.54	19.19	15.33	18.61	19.16
Abstractive Summarisation after Oracle Pre-Abstractive Extractive Step									
mT5 Cross-lingual zero-shot transfer	R-1 F	44.26	9.13	9.63	11.23	10.77	6.93	9.71	19.99
	R-2 F	21.73	2.85	2.52	3.71	3.26	1.76	2.48	8.53
	R-L F	24.25	6.31	6.32	7.81	7.51	5.05	6.53	11.92
mT5 In-language multi-task	R-1 F	43.19	33.14	36.92	37.69	34.54	27.95	37.00	37.05
	R-2 F	21.33	13.47	17.40	17.46	18.05	13.65	13.87	17.51
	R-L F	23.70	17.00	21.44	21.33	21.44	16.98	19.01	20.78
mT5 In-language	R-1 F	44.26	35.06	39.41	43.81	41.00	32.26	4.27	**40.04**
	R-2 F	21.73	13.63	17.76	19.29	20.22	14.34	0.58	**18.23**
	R-L F	24.25	17.53	22.03	23.76	24.44	18.67	3.06	**21.93**
Abstractive Summarisation after mBERT Pre-Abstractive Extractive Step									
mT5 Cross-lingual zero-shot transfer	R-1 F	37.24	7.19	9.14	10.02	9.56	6.30	12.40	17.08
	R-2 F	13.00	1.68	1.87	2.48	2.27	1.30	2.82	5.25
	R-L F	19.68	5.08	5.97	6.89	6.74	4.58	7.37	10.00
mT5 In-language multi-task	R-1 F	35.56	27.05	32.59	32.94	30.01	23.53	32.90	31.30
	R-2 F	12.28	7.84	13.06	11.65	13.14	9.37	10.24	11.24
	R-L F	18.70	13.71	18.93	18.16	18.82	14.22	16.93	17.25
mT5 In-language	R-1 F	37.24	29.65	36.02	39.79	37.21	28.47	4.32	**35.03**
	R-2 F	13.00	8.32	14.08	13.86	15.46	10.66	0.10	**12.37**
	R-L F	19.68	14.76	20.08	21.17	13.20	16.65	2.80	**18.04**

Table 3: ROUGE F-measure results of the three evaluations presented Section 5.4 on WikinewsSum. We compare the extractive models, and mT5 in the three training scenarios and with two different pre-abstractive extractive steps (Oracle and mBERT) for each language of the WikinewsSum dataset in addiction to the all dataset. Bold values are the best scores obtained for each evaluation on the all WikinewsSum dataset (Oracle method excluded).

by Miller (2019) combined with multilingual transformer models stands as a strong extractive baseline for the WikinewsSum dataset.

6.2. Abstractive Summarisation after Oracle Pre-Abstractive Extractive Step

With *Abstractive Summarisation after Oracle Pre-Abstractive Extractive Step*, we evaluate the three abstractive training scenarios with the theoretical optimum extractive method. First of all, we observe that mT5 trained in the cross-lingual zero-shot transfer training scenario obtains ROUGE scores 4.0 times worse than mT5 trained in the in-language multi-task scenario and 4.3 times worse than mT5 trained in the in-language scenario on the non-English samples (see Table 3). This is due to the fact that the summaries generated by the mT5 cross-lingual zero-shot transfer model are always generated in English even if the source text is not in English. Secondly, Table 3 shows that the in-language training scenario yields better ROUGE scores than the in-language multi-task one except for Italian. Indeed for Italian, the mT5 in-language model is not able to generate correct sentences which explains the ROUGE scores obtained. When we manually evaluate the models, we observe that the models trained each language separately (in-language training scenario) produce summaries in the language they were trained one, while the model trained on all the languages once (in-language multi-task training scenario)

Methods	Metrics	English	German	French	Spanish	Portuguese	Polish	Italian	All Languages
Abstractive Summarisation after Oracle Pre-Abstractive Extractive step									
mT5	R-1 F	44.67	9.48	10.25	12.31	14.31	7.43	9.26	23.90
Cross-lingual	R-2 F	22.15	3.00	2.59	4.36	5.00	2.07	1.99	10.73
zero-shot transfer	R-L F	24.54	6.57	6.68	8.48	9.99	5.46	6.09	14.04
mT5	R-1 F	44.12	37.14	41.44	45.56	48.04	34.79	44.98	42.14
In-language	R-2 F	21.88	15.54	19.09	22.10	28.81	17.89	19.90	**20.46**
multi-task	R-L F	24.21	19.06	23.23	25.43	30.34	21.04	23.09	**23.44**
mT5 In-language	R-1 F	44.67	36.70	41.17	46.24	48.01	34.66	3.91	**42.26**
	R-2 F	22.15	15.03	18.71	22.01	28.97	17.59	0.51	20.38
	R-L F	24.54	18.53	22.56	25.36	30.39	20.67	3.19	23.33

Table 4: ROUGE F-measure results of mT5 in the three training scenarios after the Oracle pre-abstractive extractive step. In this table, the cross-lingual samples are excluded and mT5 is evaluated only on the samples where all input documents are in the same language as the target summary.

produces summaries in the same language as the input texts. However WikinewsSum contains many cross-lingual samples (see Table 1) which advantages models trained in the in-language training scenario as they always produce the summary in the correct language. To remove this bias in the results, we evaluate the abstractive models after the Oracle pre-abstractive extractive step on the WikinewsSum dataset without the cross-lingual samples. The results are shown Table 4. In these conditions, we see that the models trained on all the languages simultaneously perform as well or even better for certain languages such as German, French, Polish, and Italian than the models trained on each language separately. The difference in the ROUGE scores is very small except again for Italian.

6.3. Abstractive Summarisation after mBERT Pre-Abstractive Extractive Step

The *Abstractive Summarisation after mBERT pre-Abstractive Extractive Step* evaluation shows the ROUGE scores obtained by the abstractive models in a real-world scenario The results show that the ROUGE scores of the abstractive models after the mBERT extractive step are in average 22% worse than the scores after the Oracle extractive step. This is expected as the Oracle extractive step is in average 32% better than the mBERT one. Secondly, we notice that the ROUGE scores obtained by mT5 after mBERT are worse compared to the scores obtained by mBERT only (see Table 3). To understand this result, we compare in the Table 5 the ROUGE scores (the ROUGE F-measure scores but also the ROUGE precision and recall scores) and the length of the produced summaries of mBERT and mT5 trained on all the languages. We remark that the abstractive summaries contain more than two times fewer words than the extractive summaries. Furthermore, the summaries generated by mT5 obtain better precision scores but worse recall scores compared to mBERT's extractive summaries.

Methods	Metrics	All Languages
Extractive Summarisation		
mBERT	R-1 P	30.60
	R-1 R	**43.34**
	R-1 F	**33.60**
	R-2 P	11.24
	R-2 R	**16.40**
	R-2 F	**12.51**
	R-L P	15.57
	R-L R	**23.08**
	R-L F	**17.36**
	# words	315
	# sents	11
Abstractive Summarisation after mBERT		
mT5 In-language multi-task	R-1 P	**39.95**
	R-1 R	29.31
	R-1 F	31.30
	R-2 P	**14.36**
	R-2 R	10.61
	R-2 F	11.24
	R-L P	**21.90**
	R-L R	16.40
	R-L F	17.25
	# words	151
	# sents	6

Table 5: Comparison of the summaries produce using the mBERT extractive model and the mT5 abstractive model trained in the in-language multi-task training scenario with regard to the ROUGE metrics and the length of the produced summaries.

7. Discussion

Our experimental findings suggest that using a pre-abstractive extractive step is a valid approach to perform abstractive summarisation on long input documents such as the ones from our WikinewsSum dataset. The results presented in Section 6 show that the hybrid summarisation approach allows abstractive summaries with high ROUGE scores. However, the mBERT extractive step still decreases the performance of the abstractive models compared to the Oracle extractive step. One solution to solve this issue would be to use an abstractive model with a greater maximum input length to remove the extractive step like Longformer or Big Bird, presented in Section 2.2, which have a maximum input length of thousands of tokens and 4096 tokens respectively. Another approach could be to try to improve the extractive step with a better extractive method. In this work, we provide initial experiments with extractive and abstractive baselines as starting point for future research.

In Section 6, we show that the abstractive models trained on one language constantly produce the summary in the same language even if the source is in another language. This can be explained because the mT5 model is trained to produce summaries in only one language independently of the input text language due to the cross-lingual samples. Moreover, at inference, the textual input given to the mT5 model does not contain any indicator for the target language. On the opposite, when trained on all the languages combined (in-language multi-task training scenario), mT5 produces summaries in the languages of the input texts. Again, this can be attributed to the fact that no indication is provided to the model to indicate the language in which to generate the summary. Therefore, mT5 reproduces the data from its training: summaries in the same language as the input text. We hypothesise that by adding a language prompt to the abstractive model, we could force mT5 to generate the summary in the correct target language. This would greatly improve the results of mT5 trained in the cross-lingual zero-shot transfer. Moreover, this language prompt could also solve the issues of the model trained on all the languages simultaneously with the cross-lingual samples (see Table 4), and therefore improve the ROUGE scores of the mT5 in-language multi-task model. We will explore different approaches to specify the language of the generated summary in future work.

We also show in Section 6 that for the Italian language, mT5 trained in the in-language training scenario obtains poor ROUGE scores. We hypothesise that the too low number of training samples (61 Italian training samples) does not allow mT5 to converge during the fine-tuning. As a result, mT5 trained on the Italian samples only is not capable to perform abstractive summarization. In the in-language multi-task training scenario, mT5 is trained on all the available samples whatever their language which solves the number of training samples issue. The produced abstractive model obtains ROUGE scores for Italian similar or even better to the ROUGE scores obtained for the other languages which have at least 30 times more training samples (see Table 1). We think that mT5 trained in all the languages understands that the summaries need to be generated in the languages of the input texts. In this training scenario, mT5 can therefore fully take advantage of its multilingual pre-training by performing what we can call cross-multilingual few-shot transfer, i.e. mT5 transfers what it learned from the English, German, French, Spanish, Portuguese, and Polish samples to Italian. This result opens new research questions. For example, would the cross-multilingual transfer have any positive effect in a zero-shot scenario where the model has never been trained on the language it is evaluated on, or do all the languages have the same influence on the performance as the in-language multi-task mT5 model on Italian samples.

Finally, we see in the Table 3 that the mBERT extractive model obtains better ROUGE F-measure scores than the abstractive models after the mBERT pre-abstractive extractive step. With regard to the results shown in Table 5, we hypothesise that the abstractive summaries are shorter, more precise, but contain less information from the gold summaries than the extractive ones.

8. Conclusion

We utilise Wikinews for the creation of a summarisation dataset. We release WikinewsSum as the first multilingual dataset for extended summaries, supporting English, German, French, Spanish, Portuguese, Polish, and Italian. Wikinews, supporting more than 20 languages with more than 1000 articles, is a good source for further populating the WikinewsSum dataset by increasing the number of languages. We consider this an important first step for future work.

Furthermore, we provide strong baselines for extractive and abstractive summarisation. In particular, we show that pre-trained multilingual transformer models can be used without fine-tuning to perform extractive summarisation with good results using the method presented in Miller (2019). We compare three multilingual training scenarios inspired by Hu et al. (2020).

First, we show that the cross-lingual zero-shot transfer does not work out of the box for the summarisation task. The produced model would need a language indication to know in which language to generate the summary. This will be explored in future work.

Secondly, we show that the mT5 models trained on each language separately obtain similar results to the mT5 model trained on all the languages combined, if we exclude the cross-lingual samples. Depending on the use case, the two training scenarios can be used if the target language has enough training samples. Indeed for low resource languages like Italian in the WikinewsSum dataset, the in-language multi-task training scenario allows the model to converge during the fine-tuning and the resulting model obtains much better results than mT5 trained on Italian only. Therefore, we consider the in-language multi-task training scenario very interesting to explore to perform abstractive summarisation for low resource languages.

Finally, we show the importance of the pre-abstractive extractive step to generate abstractive summaries in our extended summary setup. Despite the extractive method we used obtains good results, improvements are still possible compared to the Oracle method.

Acknowledgements

The research presented in this article is funded by the German Federal Ministry of Education and Research (BMBF) through the project QURATOR (Unternehmen Region, Wachstumskern, no. 03WKDA1A) and by the project PANQURA (no. 03COV03E). We would like to thank the anonymous reviewers for their helpful comments.

References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv:2004.05150*.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana, June. Association for Computational Linguistics.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *ACL*.

Dernoncourt, F., Ghassemi, M., and Chang, W. (2018). A repository of corpora for summarization. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Fabbri, A. R., Li, I., She, T., Li, S., and Radev, D. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084.

Graff, D., Kong, J., Chen, K., and Maeda, K. (2003). English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Grusky, M., Naaman, M., and Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719.

Hasan, T., Bhattacharjee, A., Islam, M. S., Mubasshir, K., Li, Y.-F., Kang, Y.-B., Rahman, M. S., and Shahriyar, R. (2021). XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online, August. Association for Computational Linguistics.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Ladhak, F., Durmus, E., Cardie, C., and McKeown, K. (2020). WikiLingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048, Online, November. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv*, pages arXiv–1910.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Liu, Y. and Lapata, M. (2019a). Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.

Liu, Y. and Lapata, M. (2019b). 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)*, pages 3721–3731.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *International Conference on Learning Representations*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692, July.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation.

Liu, Y. (2019). Fine-tune bert for extractive summarization. *arXiv e-prints*, pages arXiv–1903.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures. *arXiv e-prints*, pages arXiv–1906.

Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Narayan, S., Cohen, S. B., and Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Over, P. and Liggett, W. (2002). Introduction to duc-2002: an intrinsic evaluation of generic news text. *Document Understanding Conference*.

Parida, S. and Motlicek, P. (2019). Abstract text summarization: A low resource challenge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5996–6000.

Paulus, R., Xiong, C., and Socher, R. (2018). A deep reinforced model for abstractive summarization. In *Int. Conf. on Learning Representations*.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv*, pages arXiv–1910.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rehm, G., Bourgonje, P., Hegele, S., Kintzel, F., Schneider, J. M., Ostendorff, M., Zaczynska, K., Berger, A., Grill, S., Räuchle, S., Rauenbusch, J., Rutenburg, L., Schmidt, A., Wild, M., Hoffmann, H., Fink, J., Schulz, S., Seva, J., Quantz, J., Böttger, J., Matthey, J., Fricke, R., Thomsen, J., Paschke, A., Qundus, J. A., Hoppe, T., Karam, N., Weichhardt, F., Fillies, C., Neudecker, C., Gerber, M., Labusch, K., Rezanezhad, V., Schaefer, R., Zellhöfer, D., Siewert, D., Bunk, P., Pintscher, L., Aleynikova, E., and Heine, F. (2020). QURATOR: Innovative Technologies for Content and Data Curation. In Adrian Paschke, et al., editors, *Proceedings of QURATOR 2020 – The conference for intelligent content solutions*, Berlin, Germany, 02. CEUR Workshop Proceedings, Volume 2535. 20/21 January 2020.

Ruan, Q., Ostendorff, M., and Rehm, G. (2022). HiStruct+: Improving Extractive Text Summarization with Hierarchical Structure Information. In *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 05. Accepted for publication. 22-27 May 2022.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Schluter, N. (2017). The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45.

Scialom, T., Dray, P.-A., Lamprier, S., Piwowarski, B., and Staiano, J. (2020). MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). 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)*, pages 1073–1083.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Inf. Proc. Systems*, pages 5998–6008.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). 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*, pages 483–498, Online, June. Association for Computational Linguistics.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Zhang, J. and Wan, X. (2017). Towards automatic construction of news overview articles by news synthesis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2111–2116.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. (2020a). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Int. Conf. on Machine Learning*, pages 11328–11339. PMLR.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020b). Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Appendix

Methods	Metrics	English	German	French	Spanish	Portuguese	Polish	Italian	All Languages
Extractive Summarisation									
DistilmBERT	B-S P	0.6920	0.6669	0.6357	0.6807	0.6680	0.6455	0.6706	0.6697
	B-S R	0.7196	0.6890	0.6846	0.7104	0.7084	0.6834	0.7068	0.7021
	B-S F	0.7052	0.6774	0.6585	0.6949	0.6869	0.6633	0.6879	0.6850
mBERT	B-S P	0.6908	0.6679	0.6354	0.6810	0.6673	0.6456	0.6618	0.6695
	B-S R	0.7215	0.6931	0.6855	0.7124	0.7084	0.6848	0.7033	0.7041
	B-S F	0.7055	0.6799	0.6587	0.6960	0.6865	0.6640	0.6816	0.6859
XLM-RoBERTa	B-S P	0.6900	0.6658	0.6351	0.6794	0.6660	0.6451	0.6752	0.6684
	B-S R	0.7173	0.6878	0.6834	0.7087	0.7061	0.6831	0.7099	0.7005
	B-S F	0.7031	0.6762	0.6576	0.6934	0.6848	0.6629	0.6917	0.6836
Oracle	B-S P	0.7238	0.6947	0.6528	0.7058	0.6930	0.6638	0.6919	0.6955
	B-S R	0.7436	0.7144	0.6967	0.7228	0.7266	0.7024	0.7190	0.7217
	B-S F	0.7332	0.7039	0.6731	0.7138	0.7087	0.6818	0.7047	0.7077
Abstractive Summarisation after Oracle Pre-Abstractive Extractive Step									
mT5	B-S P	0.7526	0.6814	0.6687	0.7014	0.6864	0.6468	0.6820	0.7009
Cross-lingual	B-S R	0.7199	0.6431	0.6579	0.6650	0.6641	0.6218	0.6480	0.6717
zero-shot transfer	B-S F	0.7354	0.6614	0.6627	0.6824	0.6746	0.6337	0.6644	0.6855
mT5	B-S P	0.7494	0.7219	0.7130	0.7306	0.7274	0.6887	0.7203	0.7274
In-language	B-S R	0.7190	0.6937	0.7174	0.7030	0.7140	0.6847	0.6942	0.7074
multi-task	B-S F	0.7334	0.7070	0.7138	0.7161	0.7197	0.6857	0.7066	0.7165
mT5 In-language	B-S P	0.7526	0.7264	0.7164	0.7374	0.7381	0.6974	0.4603	0.7321
	B-S R	0.7199	0.6939	0.7179	0.7073	0.7194	0.6908	0.5261	0.7092
	B-S F	0.7354	0.7093	0.7153	0.7216	0.7277	0.6931	0.4905	0.7196
Abstractive Summarisation after mBERT Pre-Abstractive Extractive Step									
mT5	B-S P	0.7202	0.6680	0.6571	0.6858	0.6757	0.6412	0.6693	0.6828
Cross-lingual	B-S R	0.7004	0.6363	0.6517	0.6576	0.6586	0.6180	0.6459	0.6615
zero-shot transfer	B-S F	0.7098	0.6515	0.6538	0.6712	0.6666	0.6290	0.6572	0.6716
mT5	B-S P	0.7157	0.6958	0.6953	0.7069	0.7094	0.6700	0.7045	0.7022
In-language	B-S R	0.6981	0.6774	0.7033	0.6891	0.7011	0.6702	0.6869	0.6910
multi-task	B-S F	0.7065	0.6861	0.6982	0.6976	0.7046	0.6693	0.6952	0.6960
mT5 In-language	B-S P	0.7202	0.7043	0.7020	0.7151	0.7186	0.6836	0.4495	0.7091
	B-S R	0.7004	0.6807	0.7069	0.6948	0.7064	0.6803	0.5213	0.6949
	B-S F	0.7098	0.6919	0.7026	0.7044	0.7116	0.6811	0.4822	0.7012

Table 6: BERTScore (Zhang et al., 2020b) precision (B-S P), recall (B-S R), and F1 (B-S F) results of the three evaluations presented Section 5.4 on WikinewsSum. We compare the extractive models, and mT5 in the three training scenarios and with two different pre-abstractive extractive steps (Oracle and mBERT) for each language of the WikinewsSum dataset in addiction to the all dataset. Hash code for the BERTScore metric: bert-base-multilingual-cased_L9_no-idf_version=0.3.11(hug_trans=4.13.0)_fast-tokenizer