Abstractive Text Summarization based on Transformer Deep Neural Networks

Vlatka Davidović
Faculty of Informatics and Digital Technology
University of Rijeka
Rijeka, Croatia
vlatka.davidovic@uniri.hr

***Abstract*—Abstractive text summarization is natural language processing task of automatically transforming an input text to a short informative summary that assembles text written by human. Text summarization nowadays relies on deep neural language models that are intensively trained on a vast amount of texts, achieving unprecedently results in language generation tasks, such as generation of short summary. Prominent deep learning architecture is based on transformer deep neural network model.**

This work briefly presents transformer architecture and its most important building block and features. Next, the neural models for abstractive summarization task built on transformer architecture are shortly overviewed and contrasted along with commonly used evaluation metrics.

Existing summarization datasets, monolingual and multilingual as important part of training, are systematized. Datasets for abstractive summarization task are mostly collected from online sources on internet. Among all datasets, English language prevails. Croatian language is underrepresented in the text summarization research arena, so this work outlines future possibilities in training Croatian abstractive summarization neural model to bridge this gap.

Keywords— abstractive text summarization, transformer models, text summarization datasets

I. INTRODUCTION

Automatic text summarization is one of the natural language processing (NLP) tasks which has the goal to convert one or more documents into short summary while preserving the main information and meaning of the original text [1]. Similarly, text summarization can be modeled as the mapping problem from the input text to shorter output text [2]. Finally, summarization can be modelled as a text generation problem [4].

Text written in natural language belongs to sequential data category where the ordering of data derives the syntax and semantics of the language [5]. Also, ordering of the words is important to derive the meaning of the whole text. Each text sequence (i.e. sentence) follows previous sequence and inherits information.

With vast and ever-increasing quantity of texts, both formal and informal, contained in documents or posted on online media, automatic summarization gains on popularity in the research community.

There are two types of automatic text summarization techniques: extractive and abstractive. Extractive summarization determines most important words, phrases and sentences in the text and use them to produce the summary [6]. While abstractive text summarization takes the semantic representation of the text and generate a new summary, that contain words and phrases that does not exist in the original text [7]. In order to be useful, the summary needs to be linguistically fluent and similar to human-written text [8]. While extractive summarization is easier to develop, abstractive text derives summaries of better quality while engaging more challenging methods. Methods require extensive computer's power capabilities (i.e. CPU, GPU, memory) especially when they are based on deep learning models [9].

Deep learning models became very complex and powerful as combine many different neural networks into complex one. Early deep learning architectures for summarization task were mostly based on recurrent neural networks (RNN) that processes text sequentially and predict next token based on previous and input tokens [10]. An extension of RNN is able to capture information from longer sequences with long short-term memory (LSTM) [11] or gated recurrent unit (GRU) variants [12]. Recently, encoder-decoder architectures based on transformer neural network has gained the popularity in many NLP tasks, especially in ones related to text generation problems - hence summarization [13] [14] [15] [16] [17]. Transformers include attention mechanism [18], that sets the focus on more valuable information in input sequences, additionally improving the quality of information to generate abstractive summaries [7] [18] [19] [20]. Finally, transformers also achieve respectable quality (e.g. in terms of fluency, coherence, consistency and relevance) of the generated text in abstractive text summarization task, but they require extensive computing resources to train the model and generate output within reasonable amount of time [9].

Nowadays, the research focus is to moved forward towards large language models (LLMs). LLMs transformer models are scaled up to larger number of parameters (over 100 billion) and huge training data. LLMs show the ability of learning "in context", with very little ("few-shot") or without ("zero-shot") input data provided [9] [21]. Although they resonate quickly and have strong summarization ability [21], there are still some unsolved challenges: datasets for low resourced languages are still missing, models do not resonate well and can derive (hallucinate) false information, and automatic evaluation of generated text needs to be improved.

This work gives an overview of deep neural models based on transformer architecture, that are used for abstractive summarization task. Training deep neural models require huge dataset, so brief information about characteristics of each dataset, where models were trained, is extracted from original papers.

Major focus of this work is to:

- determine if abstractive summarization model based on transformer architecture for Croatian language exist,
- determine how much is Croatian language represented in existing datasets, that are used in abstractive summarization tasks,
- examine which neural model can be best as starting point for the text summarization training for the Croatian, and which characteristics of Croatian summarization dataset are needed.

After introduction, structure of the paper is as follows. The second section presents overview of related works. Description of original transformer architecture is in the third section. While, deep neural models based on transformer architecture, and modified for abstractive summarization task are presented in fourth section. Fifth section is related to the evaluation metrics used to assess the quality of generated summary. Available summarization datasets are overviewed in the sixth section. The seventh section presents the results of summarization task on machine translated Croatian dataset. Finally, paper conclude with research plan on Croatian summarization task.

II. Related work

Nowadays, abstractive summarization uses deep neural models to generate better summaries of the input text. Deep learning has developed from artificial neural network (ANN) that consists of input and output nodes, connections between them with weight parameters and activation functions. Simplest ANN is feedforward neural network which process inputs through layers: input layer, hidden layers and output layer. The predominate neural architecture is based on transformer models, so the scope of related work in this paper is limited to the overview of transformer networks.

Encoder-decoder architecture [12] [22] with attention mechanism [18] achieved prominent results in abstractive summarization [2] [19]. Encoder reads the input sequence of variable length, encode it into a single fixed-length context vector representation and generate sequence of hidden states. Attention mechanism learns which parts of the input sequence are valuable and provides decoder with information which parts to attend more. Decoder decodes the representation and produce the sequence of variable length as the output text.

In 2017. transformer architecture is introduced in the paper Attention is All You Need by Vaswani et al. [23]. Architecture consists of encoder and decoder stacks of multiple neural networks layers. Important novel characteristics of transformer architecture are self-attention mechanism and parallel processing of sequences. Self-attention is a special case of attention mechanism [18] that attend to different positions of a single sequence in order to compute a representation of that sequence. Parallel processing enabled speed-up of the learning process which enabled huge number of parameters trained in the transformer models, and open a new era in training of large language models (LLMs).

The latest models are based on transformer architecture. They can be divided into encoder only, encoder-decoder and decoder only models.

- **Encoder-only transformers** or **autoencoder transformers** (BERT [24], ERNIE [25], RoBERTa [26], ALBERT [27], ELECTRA [28], DeBERTa [29]) are trained using masked language model (MLM). It randomly masks some of the tokens from input and then tries to predict the original token based only on its context. Model tries to reconstruct the original sentence.
- **Decoder-only transformers** or **autoregressive transformers** (GPT-1 [30], GPT-2 [31], GPT-3 [32], GPT-4 [33], XLNet [34], OPT [35], OPT-ILM [36], BLOOM [37], BLOOMZ [37], GPT-Neo-X-20B [38], GPT-J-6B [39], YaLM-100B [40], GLM [41], Galactica [42], LLaMa [43], ChatGPT [44], PaLM [45], LaMDa [46], Falcon [47]) are trained on prediction of next token based on previously predicted tokens. Mask of next tokens in the input is used so that attention cannot access to later part of sentence, that need to be learned. Still, autoregressive learning is based only on predicting the next token in the sequence.
- **Encoder-decoder transformers** or **sequence-to-sequence transformer** (BART [48], mBART [49], T5 [50], mT5 [51], Switch [52], T0 [53], Tk-Instruct [54], FlanT5 [55], UL2 [56], FlanUL2 [56], Pegasus [57]) use encoder and decoder of original transformer.

For text processing we need the good representation of the data [10]. Historically data representation was based on bag-of word (i.e. TF-IDF) while recently has been substituted with embeddings [23] [58]. Embedding based data representation can be learnt as a side-effect of training the neural model, transforming the representation in low-dimensional vector space, where has been shown that embedding vectors representing words of similar meaning (i.e. share common context in the corpus) are in a proximity [59]. After training, obtained embedding vectors as data representations, can be transferred to the new NLP tasks, usually including some transfer learning. Transfer learning involves pre-training on one task where model learns good representation, and then using trained representations for fine-tuning of the model on new dataset and on similar, yet new, task [58] [60] [24]. Training of large models require lot of computing power and resources, so in order to reduce the cost of the training, transfer learning has become widely and predominantly used in the NLP field.

Pre-training can be unsupervised (UniLM [74], GPT-3 [32]), supervised (Galactica [42]) or self-supervised (T5 [50], UL2 [56], Pegasus [57], ProphetNet [61]). Transformer models are usually pretrained on unlabeled and huge text quantity to better learn general language representations that can be fine-tuned for downstream NLP tasks. Pretraining on one or multiple tasks with subsequent fine-tuning using both unsupervised and self-supervised setup achieved state-of-the-art performance for many NLP challenges, at the same time enabling better generalization of the language model.

Better language models are trained on larger set of data [62], so to accelerate the process, along the hardware, models are also scaled up. Transformer models work with very large number of parameters: 65 million in original transformer, 340 million in BERT large [24], 400 million in BART large [48], 770 million in T5 large [50], 1.5 billion in GPT-2 [31], 540 billion in PaLM [45], 1.76 trillion in GPT-4 [33]. Parameters

were rapidly scaling up (Figure 1) and trend is shifted towards developing the large language models (LLMs) [63]: GPT-3 [32] and GPT-4 [33], LLaMA [43], PaLM [45], PaLM2 [64]. Only a few companies and organizations can follow that trend: OpenAI, Meta, Google, Microsoft, AI2. Among non-profit organization that can still play in large language model training arena are EleutherAI and BigScience.

LLMs can generalize well and pre-trained model can be ported to new tasks via training that involves only a few examples. It is called few-shot learning or in-context learning [65]. Tendency is going to the making communication with LLMs similar to communication with human, employing one-shot or zero-shot learning. While in one-shot learning only one example is presented and other examples are clustered around the same point in representation space, in zero-shot learning no examples are provided. Nowadays, few-shot learning is predominantly used as a prompting technique to interact with the LLMs [66].

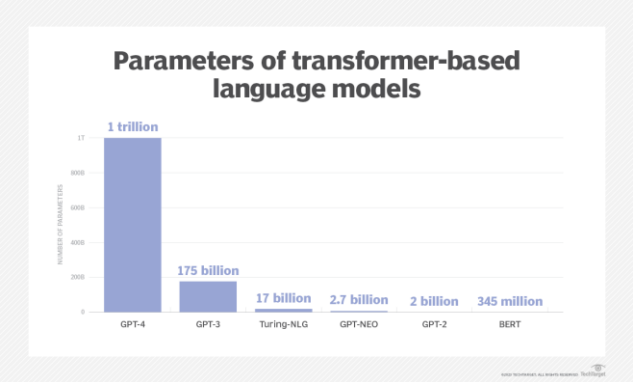

Fig. 1 Number of parameters in LLMs [68]

LLMs for summarization task can be divided roughly into two sets**: encoder-decoder** (or sequence-to-sequence) and **autoregressive language models** [13], and architecture shifts toward **instruction-tuned models** (Figure 2). Instruction-tuned models are LLMs pre-trained on huge amount of data, for diverse mixture of tasks, so model start to generalize better. Model is then fine-tuned with steps of written instructions given in prompted form, or instruction-tuning [63] [46] [53] [67].

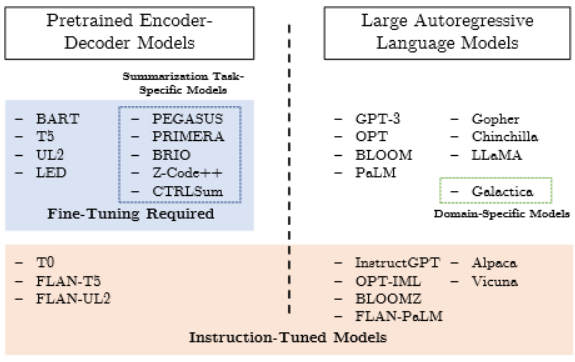

Fig. 2 LLMs for summarization task divided into pre-trained encode-decoder and large autoregressive models [13]

Although LLMs are mostly built upon general NLP tasks (i.e. autoregressive learning of the next token or simultaneously learning several tasks), there are also models pre-trained especially for abstractive summarization task (Pegasus [57], Longformer ED [73], Z-Code++ [16]) or fine-tuned for it (CTRLSum [69], Primera [70], BRIO [71]). Fine-tuning with few-shot learning settings has been used to the opinion summarization [72], summarization of medical dialogues [14] and on standard benchmark news datasets [15].

English language is the most represented in reported summarization work. Other languages than English, for creating and training the models are usually included in **multilingual models**: mBART [49], mT5 [51], XL-Sum [17], PaLM [45], BLOOM [37], Z-Code++ [16]. mBART is pretrained on 25 languages [49], mT5 and Z-Code++ are pretrained on 101 languages [51] [16], XL-Sum on 44 languages [17], PaLM on 124 languages (78% English) [45], BLOOM contains 46 languages [37].

Croatian language is presented as a small part of multilingual PaLM dataset [45]. Dataset consists of 780 billion of tokens, mixture of data collected through internet, like multilingual Wikipedia, filtered multilingual webpages, books, news articles, social code and social media conversations. Croatian words appeared only in 0.027% of dataset [45].

Due to very high skewness toward English language, multilingual models are more successful doing NLP tasks in English.

It is worth noticing that several monolingual summarization models are pre-trained: for Macedonian language - Macedonizer [75], for Slovenian language [76], Italian [77] and GPT-3 is fine-tuned for Russian summarization [78].

III. Transformer Architecture

Original transformer architecture is structured from two building blocks: encoder and decoder. Encoder takes input sequence of word representation and pass it through a six encoder blocks to output. The encoder's output, a sequence of continuous representations, is passed to decoder, which generates output sequence of words one token at a time.

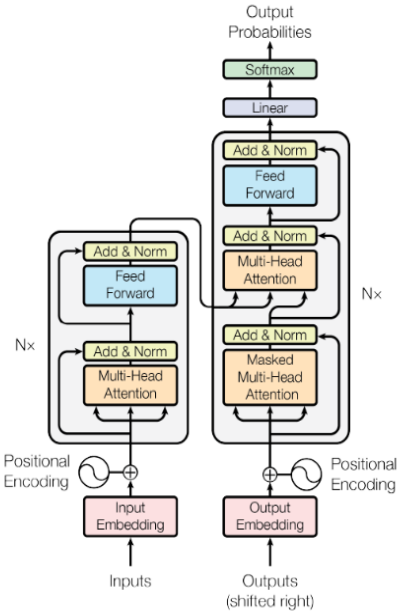

Fig. 3 . Transformer architecture [23]

The encoder block is composed of several network layers, where each layer has two parts: multi-head self-attention layer and position-wise fully connected feed-forward network. Every part of the layer has residual connection and layer normalization.

Residual connection pass information through the layer, adds a skip connection to pass information around the layer, and then data is combined. Dropout is part of regularization technique applied to the output of each sub-layer. "Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error" [10]. After that, output is normalized and passed to new encoder layer.

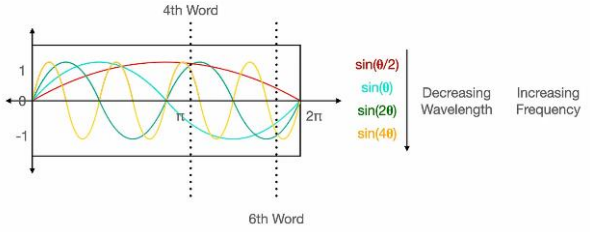

Fig. 4 Positional encoding in transformer [79]

Before passing the sequence to encoder, the sequence is tokenized and converted into word embeddings. To keep ordering of the word in the sequence, positional encoding is added after input embedding. Positional encoding is calculated as sine (1) and cosine (2) functions of different frequencies.

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (1)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \quad (2)$$

Self-attention or intra-attention mechanism is a primary building block of transformer. It computes a context of the word in the sentence, so related words get a high score value.

Attention calculates the scalar product between embedding vectors and query W^Q, key W^K and value W^V weight matrices to make some linear projections and create a key, query and a value vector. The query a representation of the current word and it is multiplied with every key value. Key vectors are the labels that match against search, and are divided by square root of keys dimension. Then is applied a softmax function. Softmax function calculates which relationships between words and which words are significant. Result is multiplied with each value vector.

Value vectors are actual word representations. Matrix of outputs is (3).

$$Attention\ (Q,K,V) = softmax\left(\frac{Q\,K^T}{\sqrt{d_k}}\right)V \quad (3)$$

Multi-head self-attention mechanism is attention function that linearly project many different sets of key, query and value with a linear projection to dimensions, which are performed in parallel, multiple times. It allows the model to jointly attend to information from different representation. It has eight attention heads, with randomly initialized weight matrices. Result of multi-head attention are matrices that are concatenated (4) (5) and normalized and sent to feed-forward neural network (6).

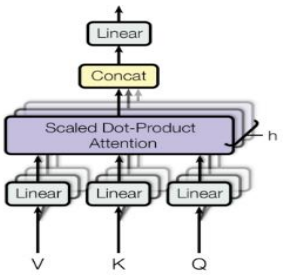

Fig. 5 Multi-head self-attention [23]

$$MultiHead\ (Q,K,V) = Concat\ (head_1,\dots,head_h)\ W^O \quad (4)$$

where each head is:

$$head_i = Attention\left(QW_i^Q,\ KW_i^K, VW_i^V\right) \quad (5)$$

Each layer in encoder and decoder has position-wise feed-forward networks

$$FFN(x) = max\ (0;\ xW1 + b1)\ W2 + b2 \quad (6)$$

Decoder stack has the same number of layers as encoder stack. Each layer has multi-head self-attention layer, fully connected feed-forward layer and masked multi-head attention over the output of the encoder stack. Also, every sublayer has residual connections and normalization around each of them. Masked multi-head attention ensures that future known output is masked, so that the prediction can depend only on the previous known outputs.

The last linear layer is a simple fully connected neural network that projects the vector produced by the stack of decoders, into a larger vector. The softmax layer then calculates those scores into probabilities and produce the word according to the highest probability.

IV. Transformer Architecture

Models based on transformer architecture are modified to have

- encoder only block,
- decoder only block,
- encoder-decoder block like original transformer.

Decoder-only (autoregressive) and **encoder-decoder (sequence-to-sequence)** models put focus on regenerating text sequences and are particularly successful in natural language generation (NLG) tasks such as machine translation, abstractive summarization and question-answering. Hence, they are predominantly used for abstractive summarization task. The overview of transformer summarization models is reported in table A in Appendix.

UniLM [74] is multilayered transformer, pre-trained on large amounts of text. It uses three types of unsupervised language modeling tasks: unidirectional, bidirectional, and

sequence-to-sequence prediction and can be fine-tuned for NLU and NLG tasks.

BertSum [80] apply BERT in summarization task with 2-stage fine-tuning: first on extractive summarization task, then on abstractive task. Extractive summarization selects important sentences from the text. Model on the second stage perform abstractive summarization, but with already pretrained encoder.

BART (Bidirectional and Auto-Regressive Transformers) [48] is denoising autoencoder that combines BERT-like bidirectional encoder with GPT-like decoder. It pretrains sequence-to-sequence model in two stages. As first step, text is corrupted with an arbitrary noising function (masking token, token deletion, text infilling, sentence permutation, document rotation, sentence shuffling, text infilling + sentence shuffling). In second step, a sequence-to-sequence model is trained to reconstruct the original text with left-to-right autoregressive decoder. Fine-tuning can work well with text generation and comprehension tasks. BART is trained on CNN/DailyMail and XSum datasets.

T5 (Text-to-Text Transformer) [50] is unified framework that converts all text-based language problems into a text-to-text problem, i.e. taking text as input and producing new text as output format. Model is pre-trained on unlabeled Colossal Clean Crawled Corpus (C4) using unsupervised learning. Authors compare the effectiveness of different transfer learning objectives, unlabeled data sets, and other factors, while exploring the limits of transfer learning for NLP by scaling up models and datasets. Model is based on original encoder-decoder transformer.

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence) [57] is large transformer-based encoder-decoder model is pre-trained on massive text corpora. Model simultaneously apply GSG (Gap Sentence Generation) and MLM (Masked Language Model) in the self-supervised pretraining. GSG selects and masks whole sentences from a document and concatenate the gap-sentences into pseudo-summary. In remaining sentences, some tokens are masked by MLM. Model is pretrained to predict these sentences. Results are validate using human evaluation.

GPT-3 (Generative Pre-trained Transformers) [32] is large autoregressive language model (LLM) with 175 billion of parameters that is trained on 300 billion tokens of text (570GB in total Common Crawl dataset) and tested in the few-shot setting. Architecture of GPT language models is based on transformer decoder blocks. Models are autoregressive, meaning that model predicts one token at a time, based on the past values. Predicted token is added to the sequence of inputs and pass to the model in its next step. GPT uses masked self-attention - it masks future tokens by interfering in the self-attention. It calculates blocking information from tokens that are to the right of the calculated position.

GSum (Guided Neural Abstractive Summarization) [81] is general and extensible guided summarization framework that can effectively take 4 external guidance as input: 1. highlighted sentences in the source document, 2. keywords, 3. salient relational triples (subject, relation, object), and 4. retrieved summaries. Experiments are performed across several different varieties, show how different type of guidance effect on quality of summary and how to generate more faithful summaries.

ProphetNet [61] is self-supervised sequence-to-sequence pre-training model with n-stream self-attention mechanism. ProphetNet learns n-step ahead prediction that predicts the next n tokens simultaneously based on previous context tokens at each time step. This future n-gram prediction is served as extra guidance that explicitly encourages the model to plan for future tokens and prevents overfitting on strong local correlations. There are two goals: (a) the model should be able to simultaneously predict the future n-gram at each time step in an efficient way during the training phase, and (b) the model can be easily converted to predict the next token only as original encoder-decoder model for inference or fine-tuning phase.

Longformer (Long-Document Transformer) [73] can process long sequences with new attention mechanism that replace the standard self-attention. Also, combine local windowed attention with a global attention. Their attention scales linearly with sequence length, enabling model to process documents of thousands of tokens or longer. Longformer Encoder-Decoder (LED) is a Longformer variant.

OPT (Open Pre-Trained Transformer Language Models) [35] is LLM with 175 billion parameters, similar to GPT-3, but with less carbon footprint then GPT-3. OPT made three benefits for research community. The full release includes: pre-trained language models of numerous sizes, a code base for training and deploying these models, and log books that detail the model development process. OPT is decoder-only architecture trained on massive dataset of unlabeled text data of English sentences. OPT-175B is evaluated over 16 standard, prompting-based NLP tasks, and is evaluated in both zero-shot and one/few-shot regimes.

SimCLS (A Simple Framework for Contrastive Learning of Abstractive Summarization) [82] is framework for abstractive summarization that bridge the gap between the learning objective and evaluation metrics. It uses contrastive learning to formulate text generation as a reference-free evaluation problem (i.e., quality estimation).

BRIO (Bringing Order to Abstractive Summarization) [71] is abstractive model with dual role: as a generation model, it generates the output summaries in an autoregressive way; as an evaluation model, it can be used to score the quality of candidate summaries by estimating a probability distribution over candidate outputs. It is built on the pretrained BART or PEGASUS but training paradigm may be extended to any encoder-decoder model.

PRIMERA (Pyramid-based Masked Sentence Pre-training for Multi-document Summarization) [70] is pre-trained model for multi-document representation with a focus on summarization that reduces the need for dataset-specific architectures and large amounts of fine-tuning labeled data. For pretraining is used a large resource where each instance is a set of related documents without any ground-truth summaries. The Newshead dataset is a relatively large dataset, where every news event is associated with multiple news articles. Approach is evaluated on wide variety of multi-document summarization datasets plus one single document dataset from various domains (News, Wikipedia, and Scientific literature) [83]. The underlying transformer model is pretrained on an unlabeled multi-document dataset.

BLOOM (BigScience Large Open-science Open-access Multilingual Language Model) [37] development was coordinated by BigScience, an open research collaboration

network. The goal was to publicly release of an open-access LLM to research community. BLOOM is open-access multilingual language model pre-trained on 176 billion of parameters, on the ROOTS corpus, a composite collection of 498 Hugging Face datasets amounting to 1.61 terabytes of text that span 46 natural languages and 13 programming languages. After pretraining BLOOM, the same massively multitask fine-tuning recipe was applied to BLOOMZ with multilingual zero-shot task generalization abilities. To train BLOOMZ, P3 is extended to include new datasets in languages other than English and new tasks, such as translation. This resulted in xP3, a collection of prompts for 83 datasets covering 46 languages and 16 tasks. Croatian language is not found in this dataset.

T0 [53] demonstrated that language models finetuned on a multitask mixture of prompted datasets have strong zero-shot task generalization abilities. T0 was trained on a subset of the Public Pool of Prompts (P3), a collection of prompts for various existing and open-source English natural language datasets.

PaLM (Pathways Language Model) [45] has 540-billion parameter, uses a standard transformer model architecture in a decoder-only setup, with some modifications: SwiGLU activation function instead of ReLU (Rectified Linear Unit) or GELU (Gaussian Error Linear Unit) that are mostly used in transformers, parallel layers, multi-query attention, RoPE (Rotary Position Embedding) embeddings that have better performance on long sequence, vocabulary use SentencePiece [84] and other. The PaLM pretraining dataset consists of a high-quality diverse corpus of 780 billion tokens.

Gopher [85] is LLM with 280 billion parameters. It is autoregressive transformer architecture with two modifications: RMSNorm is used instead of LayerNorm from original transformer, and the relative positional encoding scheme rather than absolute positional encodings. RMSNorm (Root Mean Square Layer Normalization) is simplified regularization technique that stabilize the layer activation. Relative encodings permit to evaluate on longer sequences than it is trained on, which improves the modelling of articles and books. Text is tokenized using SentencePiece [84]. The Gopheris trained on MassiveText, a collection of large English-language text datasets from multiple sources: web pages, books, news articles, and code. MassiveText contains 2.35 billion documents, or about 10.5 TB of text. Dataset does not contain Croatian words.

Chinchilla [86] is used for compute-optimal training, the model size and the number of training tokens should be scaled equally: for every doubling of model size the number of training tokens should also be doubled. Hypothesis is that by training a predicted compute-optimal model, Chinchilla, that uses the same compute budget as Gopher but with 70B parameters and 4x more data.

Galactica [42] is a large language model that can store, combine and reason about scientific knowledge. It is trained on a large scientific corpus of papers, reference material, knowledge bases and many other sources. Galactica also performs well on reasoning, Galactica was used to help write papers, including recommending missing citations, topics to discuss in the introduction and related work, recommending further work, and helping write the abstract and conclusion.

LLaMA [43] is collection of foundation language models ranging from 7B to 65B parameters trained on trillions of tokens, and show that it is possible to train state-of-the-art models using publicly available datasets exclusively, without resorting to proprietary and inaccessible datasets.

UL2 (Unifying Language Learning Paradigms) [56] is based on vanilla T5 transformer with GLU layers [87], and T5-style relative attention. GLU (Gated Linear Units) is used instead of ReLU activation function and has stable and better control of passing information. UL2 works with two key ideas: first is Mixture-of-Denoisers (MoD) pretraining objective that combines diverse pre-training tasks and mix them together, and second is mode switching that associates fine-tuning task with pre-training schemes. Dynamic mode switching works via discrete prompting. Model achieve strong results at in-context learning and works well with chain-of-thought prompting and reasoning tasks.

Big Bird [88] is transformer-based model with a sparse attention mechanism that reduce computational complexity in self-attention layer to linear approximation. Quadratic dependency is related to self-attention matrix where complexity is $O(n^2*d)$, n is number of tokens and d is the representation dimension. The proposed sparse attention can handle sequences of length up to 8x and consequence is the capability to handle longer context. As a result, Big Bird improves performance on various NLP tasks such as question answering and summarization.

Z-Code++ [16] is pretrained encoder-decoder model that is using three techniques: (1) two-phase pre-training process to improve model's performance on low-resource summarization tasks, (2) using disentangled attention layers that replace self-attention layers in the encoder, (3) fusion-in-encoder, a simple yet effective method of encoding long sequences in a hierarchical manner. In two-phase pre-training first pre-trained using text corpora for language understanding, and then is continually pre-trained on summarization corpora for grounded text generation. Z-Code large has 710M parameters and is pre-trained on English data and multi-lingual data across 5 languages. Croatian is not included.

V. Evaluation metrics

Summarization evaluation metrics can be divided into manual or automatic. Manual relies on human judgment: candidate summary is evaluated based on evaluators subjective merit. Automatic evaluation can be reference-based (candidate summary is compared to the "perfect" or "gold" summary) or reference-free (directly or indirectly define a model of document content salience and evaluate the content of candidate summary against this salience).

According to [89], there are four different dimensions of **summary quality**:

- **Coherence**: The collective quality of how well the summary is structured and organize,
- **Consistency**: The extent to which the summary contains information which is factually supported by the input document,
- **Fluency**: The grammatical correctness of the sentences,
- **Relevance**: How well the summary selects important content from the source document.

Among automatic evaluation methods, the most commonly used metric is **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) [90]. ROUGE is reference-based metric which measure quality of candidate summary comparing it to the reference summary. Reference summary is mostly created by humans.

ROUGE-n counts the number of overlapping n-grams between candidate and reference summaries (9). Precision (7) is percent of n-grams in the reference summary that are also present in the candidate summary. Recall (8) is percent of n-grams in the reference summary that are also present in the reference summary.

$$P_{rouge_n} = \frac{|\text{n\_grams}_{candidate} \cap reference|}{|\text{n\_grams}_{candidate}|} \quad (7)$$

$$R_{rouge_n} = \frac{|\text{n\_grams}_{candidate} \cap reference|}{|\text{n\_grams}_{reference}|} \quad (8)$$

$$Rouge_n F1 = 2 * \frac{\text{P}_{rouge_n} * \text{R}_{rouge_n}}{\text{P}_{rouge_n} + \text{R}_{rouge_n}} \quad (9)$$

ROUGE-L calculates the longest common subsequence between two summaries (LCS) (12). Longer shared sequence indicates more similarity between summaries. LCS reflect sentence-level word order but does not require consecutive matches. Length of X is |m|, and length of Y is |n|. Precision (10) and recall (11) are calculated prior to F1-score (12).

$$P_{rouge\_lcs} = \frac{\text{LCS (X,Y)}}{|m|} \quad (10)$$

$$R_{rouge\_lcs} = \frac{\text{LCS(X,Y)}}{|\text{n}|} \quad (11)$$

$$Rouge_{LCS} F1 = \frac{(1+\beta^2)\text{R}_{LCS} * \text{P}_{LCS}}{\text{R}_{LCS} + \beta^2 \text{P}_{LCS}} \quad (12)$$

ROUGE-W [91] is a weighted longest common subsequence (WLCS) that measure length of consecutive matches and put more weights (score) on consecutive matches. The weighting function f has properties:

f (x + y) > f(x) + f(y) for any positive x and y, and

f (k) = α*k - β, k >=0 and α, β>0.

The function charges a gap penalty of -β for each non-consecutive n-gram sequences. For X sequence of length m, Y sequence of length n, WLCS precision (13), recall (14) and F1-score (15) metrics are calculated.

$$P_{rouge\_WLCS} = f^{-1}\left(\frac{\text{WLCS (X,Y)}}{f(m)}\right) \quad (13)$$

$$R_{rouge\_WLCS} = f^{-1}\left(\frac{\text{WLCS (X,Y)}}{f(n)}\right) \quad (14)$$

$$Rouge_{WLCS} F1 = \frac{(1+\beta^2) R_{WLCS} * P_{WLCS}}{R_{WLCS} + \beta^2 P_{WLCS}} \quad (15)$$

The **BERTScore** [92] is evaluation metric based on pre-trained BERT contextual embeddings [24]. Each token is represented by its BERT embedding. All tokens embeddings in one summary are aligned to some token embedding in another summary based on cosine similarity. Final score is a normalized sum of all the alignment weights. For reference sentence x = (x_1, …, x_k) and candidate x' = (x'_1, …, x'_m) cosine similarity of reference token and candidate token is:

$$cos(x_i, x_j') = \frac{x_i^T x_j'}{\|x_i\| \|x_j'\|} \quad (16)$$

To compute precision, each token in x' is matched to a token in x (17) and vice versa to compute recall (18). Similarity score is maximized with matching most similar tokens. F1 score is given in (19).

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in X} x_i^T \hat{x}_j \quad (17)$$

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in X} \max_{\hat{x}_j \in \hat{x}} x_i^T \hat{x}_j \quad (18)$$

$$F1_{BERT} = 2 * \frac{P_{BERT} * R_{BERT}}{\text{P}_{BERT} + \text{R}_{BERT}} \quad (19)$$

VI. Datasets for Summarization

Transformers require massive datasets for training and summarization datasets are usually limited in size since they consist of pairs: source text and related summary. Generally, summarization datasets can be categorized into monolingual, multilingual and cross-lingual. Cross-lingual summarization generate a summary in one language, if input document(s) is given in other language. It incorporates two tasks: translation and summarization together [93]. Among published monolingual summarization datasets, most representative is English language[1] as the most resourceful language on the Internet.

Monolingual datasets are mostly domain specific:

CNN/DailyMail [2] [3], XSum [94], Newsroom (CORNELL newsroom) [95], New York Times [96], Gigaword [97] [7] contain news articles; arXiv [98], ScisummNet [99], Pubmed [100] contain scientific papers; BigPatent [101] contains patent documents; WikihowQA [102] contain question-answering dataset with summary; Reddit TIFU [103], SAMSum [104], DialogSum [105], AESLC [106] contain conversation texts; WikiHow [107] and BookSum [108] contains knowledge base documents and literature.

Less datasets are available for other languages, like Chinese (LCSTS [109] short texts from microblogging website), Italian (IlPost [110] news from Fanpage), Indonesian (Liputan6 [111] and IndoSum [112] from news portals), German (Klexikon [113] children's lexicon), Spanish (DACSA [114] from Catalan and Spanish newspaper) and some other languages.

Low resourced languages can mostly be found in multi-lingual and cross-lingual summarization datasets. Both, multilingual and cross-lingual datasets contain documents in

[1] https://paperswithcode.com/datasets?task=text-summarization

several languages. While documents in multilingual datasets are not necessarily aligned, cross-lingual datasets are prepared for cross-lingual summarization task that summary can be produced in a different language from a source [115].

Multilingual datasets contain article/summary pairs in different languages: ML-SUM consists of 5 different languages [116], XL-Sum 44 different languages ranging from low to high-resource [17], multilingual Common Crawl (mC4) [51] include over 100 languages, Infiniset [46] contains dialog data from public web documents with only 6.25% of non-English documents, PaLM [45] dataset contains mixture of data in 124 languages collected through internet. Among multilingual datasets, Croatian language is included only with 0.027% in PaLM dataset [45].

Cross-lingual datasets WikiLingua consists of 18 languages [117], WikiMulti 15 languages [118], xP3 46 languages [119], XWikis 4 languages [115], EUR-Lex-Sum 24 European languages [120]. Among cross-lingual datasets, Croatian language is included only in EUR-Lex-Sum, which consists of European law documents, human translated into 24 languages [120].

Croatian language is underrepresented in any of available datasets.

Most of datasets belongs to **single document summarization**, where every document has related summary. **Multi-document summarization** task creates summary from multiple documents. Dataset Multi-News is large-scale multi-document summarization (MDS) of news articles and human-written summaries of these articles [121].

The overview of summarization datasets is in Table B in Appendix.

Level of abstractedness and compression ratio between text and summary have substantial impact on the generated summary. **Abstractedness** of the dataset is measure of unique n-grams in the reference summary which are not in the text. **Compression ratio** is defined by sizes of text and summary. Small number of words/tokens in text and summary can give better results. The higher is the compression ratio, abstractive summarization is more challenging [107].

VII. Croatian summarization task

Currently the trained neural model for Croatian abstractive summarization is still missing. Although number of deep learning models work with multilingual data, Croatian language underrepresented. In huge multilingual dataset built for PaLM model, Croatian words appeared only in 0.027% of dataset [45]. To temporarily bridge the gap, we have translated well known news summarization dataset CNN/DailyMail [2] [3] to Croatian language, using Google machine translation [122].

TABLE I. The average number of words in the text and summary

CNN/DailyMail datasets	Avg. num. of words in the document		Vocabulary size	
	text	*summary*	*text*	*summary*
English	659	49.9	136,927	30,963
Croatian MT	579.8	45.7	305,163	64,445

Texts analysis are presented in the Table I. Models are trained on both datasets for comparison of results. Both datasets are preprocessed and tokenized. Training is made on 2 different models: LSTM-based and Bi-LSTM-based networks with attention mechanism placed in encoder-decoder architecture. Models are trained on English and Croatian MT. Table II shows training phase: number of parameters and epochs, batch size and training time in hours.

TABLE II. Comparison of training data in different models and datasets (ES=early stopping)

Dataset and model	Num. of parameters	Batch size	Train. time	Num. of epochs
EN-LSTM	55,011,863	400	122h	33 (ES)
HR-LSTM	115,486,155	250	148h	26 (ES)
HR-Bi-LSTM	135,392,655	250	303h	50

Table III shows evaluation results, using ROUGE and BERTScore metrics. Results show some differences in favor of English dataset. Overall, generated summaries were not generating correct content, but most topics were well captured.

TABLE III. Evaluation using ROUGE and BERTScore metrics

Dataset and model	Rouge 1 (F1)	Rouge 2 (F1)	Rouge L (F1)	BERTScore (F1)
EN-LSTM	20.29%	3.41%	15.10%	82.42%
HR-LSTM	16.91%	2.75%	12.58%	64.94%
HR - Bi-LSTM	18.71%	3.17%	13.22%	66.17%

This initial attempt was served as the shortcut for training to get initial results of the abstractive text summarization model for Croatian language.

VIII. Conclusion and discussion

Main goal of this work is to present different transformer architectures for abstractive text summarization. Two type of this architecture are mainly used for summarization task: sequence-to-sequence and autoregressive transformers. In a last few years language models scaled up rapidly from millions to billions and lately trillions of parameters and using a huge amount of data for training. This, so called large language models (LLMs), currently consist of over 100 billion of parameters and require immense computational resources. As the results, they enable a better generalization in NLP tasks that can be fine-tuned with zero or few-shot learning.

Transformer neural language models require huge datasets for training processes, due to the extensive number of parameters that need to be set.

Most summarization models are trained on English datasets, with only sporadic models reported for other languages. So, there is still open questions in creation of the summarization models for languages other than English. Moreover, we are still missing studies if transformer-based summarization models will perform as good as for English in other languages.

The first question addressed in this work was to determine if abstractive summarization model based on transformer architecture for Croatian language exist. Among all summarization models based on transformer architecture there is not reported any model designed especially for the Croatian language.

The second question was to determine how much is Croatian language represented in existing multilingual datasets, that are used in abstractive summarization tasks. Among publicly available summarization dataset, there is no reported Croatian dataset for summarization task. Only a few transformer models are trained on multilingual or cross-lingual datasets, where Croatian is represented only as a part of one cross-lingual dataset. Specifically, Croatian words are under 1% of the data used for training. Hence, Croatian language is underrepresented in all existing models and datasets for summarization task, and this should be addressed in the future.

The third question was to examine which neural model can be the best as starting point for the abstractive state-of-the-art summarization training for the Croatian, and which characteristics of Croatian summarization dataset are needed. According to reported neural models for other languages, the reasonable starting point is to select neural model that requires smaller scale of data to train the parameters. Hence, the preferred model is one with limited number of parameters and needed computing resources. Given that, large language models will not be considered for training in next stage of the PhD work. The reason is restricted access to computational resources for training the LLMs. Croatian dataset for summarization task still needs to be large, collected from different resources and contain different text domains for better generalization. That dataset will be used for training of one of the smaller summarizations model's architecture. Transformer models can achieve reasonably good results even on the smaller scales of parameters: BART-base has only 140 million of parameters, T5-small has 60 million of parameters, Longformer-small has 41 million of parameters. Hence, the future work will be focused in training of these models.

Finally, the first steps in PhD research will be construction of the Croatian dataset followed by training of the Croatian model for abstractive text summarization task based on transformer architecture.

REFERENCES

[1] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with Pointer-Generator Networks," Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017.

[2] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, 2016.

[3] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching Machines to Read and Comprehend," Neural Information Processing Systems, 2015.

[4] A. Gatt and E. Krahmer, "Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation," Journal of Artificial Intelligence Research, vol. 61, pp. 65–170, Jan. 2018.

[5] Y. Goldberg, Neural network methods in natural language processing. Morgan & Claypool Publishers, 2017.

[6] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E.D. Trippe, J.B. Gutierrez and K. Kochut, "Text Summarization Techniques: A Brief Survey." International Journal of Advanced Computer Science and Applications, 8 (10), pp. 397-405, 2017.

[7] S. Chopra, M. Auli, and A. M. Rush. "Abstractive sentence summarization with attentive recurrent neural networks." In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 93-98. 2016.

[8] C. F. Greenbacker, "Towards a Framework for Abstractive Summarization of Multimodal Documents", Proceedings of the ACL-HLT 2011 Student Session, pages 75–80, Portland, OR, USA 19-24 June 2011.

[9] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein et al. "On the opportunities and risks of foundation models." arXiv preprint arXiv:2108.07258, 2021.

[10] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997.

[12] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." In Conference on Empirical Methods in Natural Language Processing (EMNLP 2014). 2014.

[13] F. Retkowski, "The Current State of Summarization." arXiv e-prints arXiv:2305.04853, 2023.

[14] D. F. Navarro, M. Dras, and S. Berkovsky, "Few-shot fine-tuning SOTA summarization models for medical dialogues". In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pp. 254–266, Hybrid: Seattle, Washington + Online. 2022.

[15] Q. Bi, H. Li, and H. Yang. "Boosting Few-shot Abstractive Summarization with Auxiliary Tasks." In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2888-2893. 2021.

[16] P. He, "Z-Code++: A Pre-trained Language Model Optimized for Abstractive Summarization", arXiv e-prints, doi:10.48550/arXiv.2208.09770, 2022.

[17] T. Hasan, A. Bhattacharjee, M. S. Islam, K. Mubasshir, Y.-F. Li, Y.-B. Kang, et al., "XL-sum: Large-scale multilingual abstractive summarization for 44 languages", Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP), pp. 4693-4703, 2021.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate". In Proceedings of the 3rd International Conference on Learning Representations (ICLR'15), http://arxiv.org/abs/1409.0473, 2015.

[19] S. Chopra, M. Auli, and A. M. Rush. "Abstractive sentence summarization with attentive recurrent neural networks." In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 93-98., 2016.

[20] R. Paulus, C. Xiong, and R. Socher. "A Deep Reinforced Model for Abstractive Summarization." In International Conference on Learning Representations. 2018.

[21] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKewn, T. B. Hashimoto, "Benchmarking Large Language Models for News Summarization", arXiv preprint arXiv:2301.13848, 2023.

[22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks." In Advances in neural information processing systems, 27. pp 3104-3112. arXiv preprint arXiv:1409.3215, 2014.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser and I. Polosukhin, "Attention is all you need." In Advances in Neural Information Processing Systems, volume 30., 2017.

[24] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding." In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, pp 4171–4186. arXiv preprint arXiv:1810.04805, 2018.

[25] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. "ERNIE: Enhanced Language Representation with Informative Entities." In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1441-1451. 2019.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, … and V. Stoyanov. "RoBERTa: a robustly optimized BERT pretraining approach.", arXiv preprint arXiv:1907.11692, 2019.

[27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." Published as a conference paper at ICLR 2020.

[28] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators,” in International Conference on Learning Representations, 2020.

[29] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” CoRR, vol. abs/2006.03654, 2020.

[30] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training”, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018.

[31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. “Language models are unsupervised multitask learners.” OpenAI blog 1, no. 8 (2019): 9., https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, 2019.

[32] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan et al. “Language models are few-shot learners.” Advances in neural information processing systems 33 (2020): 1877-1901., https://browse.arxiv.org/pdf/2005.14165.pdf, 2020.

[33] OpenAI, “GPT-4 Technical Report,” Mar. 2023. Available: https://cdn.openai.com/papers/gpt-4.pdf , 2023.

[34] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding.”, 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada, 2019.

[35] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen et al. “OPT: Open Pre-trained Transformer Language Models.”, arXiv (Cornell University), doi: https://doi.org/10.48550/arxiv.2205.01068, May 2022.

[36] S. Iyer, X. V. Lin, R. Pasunuru, T. Mihaylov, D. Simig, P. Yu, et al., OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization, https://arxiv.org/abs/2212.12017, 2022.

[37] BigScience Workshop, “BLOOM: A 176B-Parameter Open-Access Multilingual Language Model”, https://browse.arxiv.org/pdf/2211.05100.pdf, 2022.

[38] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding et al., “GPT-NeoX-20B: An Open-Source Autoregressive Language Model.” In Proceedings of BigScience Episode #5 -- Workshop on Challenges & Perspectives in Creating Large Language Models, pp.95–136, virtual+Dublin. 2022.

[39] B. Wang, and A. Komatsuzaki, “GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model”, https://github.com/kingoflolz/mesh-transformer-jax, 2021.

[40] M. Khrushchev, R. Vasilev, A. Petrov, N. Zinov, YaLM 100B, https://github.com/yandex/YaLM-100B, 2022.

[41] Z. Du, Y. Qian, X. Liu, M. Ding, J. Qiu, Z. Yang, and J. Tang. “GLM: General Language Model Pretraining with Autoregressive Blank Infilling.” In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 320–335, Dublin, Ireland, 2022.

[42] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia et al., “Galactica: A Large Language Model for Science.” arXiv preprint arXiv:2211.09085, 2022.

[43] H. Touvron, T. Lavril, G. Izacard, X. Martinet, MA. Lachaux, T. Lacroix, B. Rozière et al. “Llama: Open and efficient foundation language models.”, arXiv preprint arXiv:2302.13971, 2023.

[44] OpenAI, Introducing ChatGPT, https://openai.com/blog/chatgpt, 2022.

[45] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts et al. "Palm: Scaling language modeling with pathways." arXiv preprint arXiv:2204.02311, https://browse.arxiv.org/pdf/2204.02311.pdf, 2022.

[46] R. Thoppilan, D. D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, HT. Cheng et al. "Lamda: Language models for dialog applications." arXiv preprint arXiv:2201.08239 (2022).

[47] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli et al. "The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only." arXiv preprint arXiv:2306.01116, 2023.

[48] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “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, pages 7871–7880, Online., 2019.

[49] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer. “Multilingual Denoising Pre-training for Neural Machine Translation.” Transactions of the Association for Computational Linguistics, 8:726–742., 2020

[50] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research 21, no. 1 (2020): 5485-5551., 2020.

[51] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant et al., “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, pp. 483–498, Online, 2021.

[52] W. Fedus, B. Zoph, N. Shazeer, “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity.” The Journal of Machine Learning Research 23, no. 1 (2022): 5232-5270., 2022.

[53] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, et al., “Multitask Prompted Training Enables Zero-Shot Task Generalization.” ICLR 2022 - Tenth International Conference on Learning Representations, Online, 2022.

[54] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Arunkumar et al., “Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks.”, In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 5085–5109, Abu Dhabi, United Arab Emirates, 2022.

[55] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W Fedus et al. "Scaling instruction-finetuned language models." arXiv preprint arXiv:2210.11416, 2022.

[56] Y. Tay, M. Dehghani, V. Q. Tran, X. Garcia, J. Wei, X. Wang et al. "Ul2: Unifying language learning paradigms." In the Eleventh International Conference on Learning Representations. 2022.

[57] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization." In International Conference on Machine Learning, pp. 11328-11339. PMLR, 2020.

[58] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer “Deep Contextualized Word Representations.” In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237, New Orleans, Louisiana. 2018.

[59] T. Mikolov, K. Chen, G. Corrado, J. Dean, “Efficient Estimation of Word Representations in Vector Space”, Proceedings of Workshop at ICLR. arXiv:1301.3781v1, 2013.

[60] J. Howard and S. Ruder, “Universal Language Model Fine-tuning for Text Classification”, In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 328–339, Melbourne, Australia. 2018.

[61] W. Qi, Y. Yan, Y. Gong, D. Liu, N. Duan, J. Chen et al., “ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training”, Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2401–2410, November 2020.

[62] I. Turc, M. W. Chang, K. Lee, and K. Toutanova. "Well-read students learn better: On the importance of pre-training compact models." arXiv preprint arXiv:1908.08962, 2019.

[63] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou et al., “A survey of large language models.” arXiv preprint arXiv:2303.18223. 2023.

[64] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos et al., “Palm 2 technical report.” arXiv preprint arXiv:2305.10403. 2023.

[65] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, “A survey for in-context learning,” CoRR, vol. abs/2301.00234, 2023.

[66] H. Zhou, A. Nova, H. Larochelle, A. Courville, B. Neyshabur, and H. Sedghi, “Teaching algorithmic reasoning via in-context learning.” arXiv preprint arXiv:2211.09066, 2022.

[67] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin et al., “Training language models to follow instructions with human feedback.”, Advances in Neural Information Processing Systems 35, pp.27730-27744, 2022.

[68] S. M. Kerner, “What is a large language model (LLM)? – TechTarget Definition,” WhatIs.com, https://www.techtarget.com/whatis/definition/large-language-model-LLM, Sep. 2023.

[69] J. He, W. Kryscinski, B. McCann, N. Rajani, and C. Xiong. “CTRLsum: Towards Generic Controllable Text Summarization.” In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp.5879–5915, Abu Dhabi, United Arab Emirates. 2022.

[70] W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, “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), pp.5245–5263, Dublin, Ireland, 2022.

[71] Y. Liu, P. Liu, D. Radev, and G. Neubig, “BRIO: Bringing Order to Abstractive Summarization.” In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.2890–2903, Dublin, Ireland, 2022.

[72] A. Bražinskas, M. Lapata, and I. Titov, “Few-Shot Learning for Opinion Summarization.” In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.4119–4135, Online, 2020.

[73] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer.” arXiv preprint arXiv:2004.05150, 2020.

[74] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang et al., “Unified language model pre-training for natural language understanding and generation.” NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp.13063–13075, Dec 2019.

[75] J. Dobreva, T. Pavlov, K. Mishev, M. Simjanoska, S. Tudzarski, D. Trajanov, and Lj. Kocarev, “MACEDONIZER - The Macedonian Transformer Language Model.” Communications in computer and information science, pp.51–62, Jan. 2022.

[76] A. Žagar, and M. Robnik-Šikonja, “One Model to Rule Them All: Ranking Slovene Summarizers.” In: Text, Speech, and Dialogue. TSD 2023. Lecture Notes in Computer Science, vol 14102. Springer, Cham, 2023.

[77] N. Landro, I. Gallo, R. La Grassa, and E. Federici, “Two New Datasets for Italian-Language Abstractive Text Summarization.” Information, vol. 13, no. 5, p. 228, Apr 2022.

[78] A. Nikolich, I. Osliakova, T. Kudinova, I. Kappusheva, and A. Puchkova, “Fine-Tuning GPT-3 for Russian Text Summarization.” In: Data Science and Intelligent Systems. CoMeSySo 2021. Lecture Notes in Networks and Systems, vol 231. Springer, Cham, 2021.

[79] N. Verma, “Positional Encoding in Transformers.” Medium, https://lih-verma.medium.com/positional-embeddings-in-transformer-eab35e5cb40d, Dec 2022.

[80] Y. Liu and M. Lapata, “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), pp. 3730–3740, Hong Kong, China, 2019.

[81] Z. Y. Dou, P. Liu, H. Hayashi, Z. Jiang, and G. Neubig, “GSum: A General Framework for Guided Neural Abstractive Summarization.” In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.4830–4842, Online, 2021.

[82] Y. Liu and P. Liu, “SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization,” In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp.1065–1072, Online, 2021.

[83] X. Gu, Y. Mao, J. Han, J. Liu, H. Yu, Y. Wu et al., “Generating Representative Headlines for News Stories,” Proceedings of The Web Conference 2020, Apr 2020.

[84] T. Kudo, and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.” In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 66–71, Brussels, Belgium, Nov 2018.

[85] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, F. Song et al., “Scaling Language Models: Methods, Analysis & Insights from Training Gopher.” arXiv preprint arXiv:2112.11446, Dec 2021.

[86] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas et al. "Training Compute-Optimal Large Language Models." arXiv preprint arXiv:2203.15556, 2022.

[87] N. Shazeer, “GLU Variants Improve Transformer,” arXiv:2002.05202 [cs, stat], Available: https://arxiv.org/abs/2002.05202, Feb 2020.

[88] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham et al. "Big bird: Transformers for longer sequences." Advances in neural information processing systems 33, pp.17283-17297, 2020.

[89] A. R. Fabbri, W. Kryściński, B. McCann, C. Xiong, R. Socher, D. Radev; SummEval: Re-evaluating Summarization Evaluation. Transactions of the Association for Computational Linguistics; 9, pp.391–409, 2021.

[90] C.Y. Lin, "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-81., 2004.

[91] J.P. Ng and V. Abrecht, “Better Summarization Evaluation with Word Embeddings for ROUGE,” 2015., In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1925–1930, Lisbon, Portugal, 2015.

[92] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “BERTScore: Evaluating Text Generation with BERT,” In ICLR 2020 Conference, 2020.

[93] J. Wang et al., “A Survey on Cross-Lingual Summarization,” Transactions of the Association for Computational Linguistics, vol. 10, pp.1304–1323, Jan 2022.

[94] S. Narayan, S. B. Cohen, and M. Lapata, “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, pp.1797–1807, Brussels, Belgium, 2018.

[95] M. Grusky, M. Naaman, and Y. Artzi, “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), pp.708–719, New Orleans, Louisiana, 2018.

[96] E. Sandhaus, The New York Times Annotated Corpus LDC2008T19. Web Download. Philadelphia: Linguistic Data Consortium, 2008.

[97] D. Graff, and C. Cieri. English Gigaword LDC2003T05. Web Download. Philadelphia: Linguistic Data Consortium, 2003.

[98] A. Cohan et al., “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), pp.615–621, New Orleans, Louisiana, 2018.

[99] M. Yasunaga et al., “ScisummNet: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Annual Conference on Innovative Applications of Artificial Intelligence, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019,” 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, pp. 7386–7393, 2019

[100] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective Classification in Network Data,” AI Magazine, vol. 29, no. 3, p. 93, Sep. 2008.

[101] E. Sharma, C. Li, and L. Wang, “BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization,” In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2204–2213, Florence, Italy., 2019

[102] Y. Deng et al., “Joint Learning of Answer Selection and Answer Summary Generation in Community Question Answering,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 7651–7658, Apr. 2020,

[103] B. Kim, H. Kim, and G. Kim, “Abstractive Summarization of Reddit Posts with Multi-level Memory Networks,” 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), pp.2519–2531, Minneapolis, Minnesota, 2019.

[104] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, “SAMSum Corpus: A Human-annotated Dialogue Dataset for Abstractive Summarization,”

Proceedings of the 2nd Workshop on New Frontiers in Summarization, 2019.

[105] Y. Chen, Y. Liu, L. Chen, and Y. Zhang, “DialogSum: A Real-Life Scenario Dialogue Summarization Dataset,” In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp.5062–5074, Online, 2021.

[106] R. Zhang and J. Tetreault, “This email could save your life: 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019,” ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, pp. 446–456, 2020.

[107] M. Koupaee, and W. Yang Wang. "WikiHow: A Large Scale Text Summarization Dataset." arXiv e-prints (2018): arXiv-1810. 2018.

[108] W. Kryściński, N. F. Rajani, D. Agarwal, C. Xiong, and D. Radev, “BOOKSUM: A Collection of Datasets for Long-form Narrative Summarization,” In Findings of the Association for Computational Linguistics: EMNLP 2022, pp.6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics, 2022.

[109] B. Hu, Q. Chen, and F. Zhu, “LCSTS: A Large Scale Chinese Short Text Summarization Dataset,” Empirical Methods in Natural Language Processing, Jun 2015.

[110] N. Landro, I. Gallo, R. La Grassa, and E. Federici, “Two New Datasets for Italian-Language Abstractive Text Summarization,” Information, vol. 13, no. 5, p. 228, Apr 2022.

[111] F. Koto, J. H. Lau, and T. Baldwin, “Liputan6: A Large-scale Indonesian Dataset for Text Summarization,” In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pp 598–608, Suzhou, China, 2020.

[112] K. Kurniawan, and S. Louvan. “Indosum: A New Benchmark Dataset for Indonesian Text Summarization.” 2018. International Conference on Asian Language Processing (IALP), pp.215-220, 2018

[113] D. Aumiller and M. Gertz, “Klexikon: A German Dataset for Joint Summarization and Simplification,” In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp.2693–2701, Marseille, France, 2022.

[114] E. Segarra Soriano, V. Ahuir, Lluís-F. Hurtado, and J. González, “DACSA: A large-scale Dataset for Automatic summarization of Catalan and Spanish newspaper Articles,” In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.5931–5943, Seattle, United States, 2022.

[115] L. Perez-Beltrachini and M. Lapata, “Models and Datasets for Cross-Lingual Summarisation,” Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Jan 2021.

[116] T. Scialom, P.-A. Dray, Sylvain Lamprier, B. Piwowarski, and J. Staiano, “MLSUM: The Multilingual Summarization Corpus,” In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8051–8067, Online, Apr 2020.

[117] F. Ladhak, E. Durmus, C. Cardie, and K. McKeown, “WikiLingua: A New Benchmark Dataset for Cross-Lingual Abstractive Summarization,” In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4034–4048, Online. Association for Computational Linguistics

[118] P. Tikhonov and V. Malykh, “WikiMulti: A Corpus for Cross-Lingual Summarization,” Communications in computer and information science, pp. 60–69, Jan 2022.

[119] N. Muennighoff et al., “Crosslingual Generalization through Multitask Finetuning,” Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Volume 1: Long Papers, pp.15991–16111, July 9-14, 2023

[120] D. Aumiller, A. Chouhan, and M. Gertz, “EUR-Lex-Sum: A Multi- and Cross-lingual Dataset for Long-form Summarization in the Legal Domain,” NASA ADS, https://ui.adsabs.harvard.edu/abs/2022arXiv221013448A, Oct 2022

[121] A. Fabbri, I. Li, T. She, S. Li, and D. Radev. 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, pp.1074–1084, Florence, Italy, 2019.

[122] V. Davidović, S. Martinčić-Ipšić, “Towards Sequence-to-Sequence Neural Model for Croatian Abstractive Summarization”, CECIIS 2023, Proceedings, str. 309-315., 2023.

APPENDIX A

TABLE IV. TRANSFORMER MODELS FOR ABSTRACTIVE SUMMARIZATION

Model	Size (num of params)	Specification	Type	Dataset	Task	Date	Created by
UniLM		pre-trained and fine-tuned. UNILM can be configured, using different self-attention masks to aggregate context for different types of language models, and thus can be used for both NLU and NLG tasks.	seq2seq	CNN/DailyMail and Gigaword abstractive summarization, SQuAD question generation, CoQA generative question answering, and DSTC7 dialog response generation.	NLU, NLG, abstractive summarization	Oct 2019	Microsoft
BART	Base: 140M Large: 406M		Seq2seq	160GB, -BookCorpus plus English Wikipedia, -CC-NEWS -OpenWebText, -Stories,	NLP tasks, NLG tasks, summarization, translation, comprehension	Oct 2019	Facebook
T5	Small: 60M Base: 220M Large: 770M XL: 3B, 11B	pretrained unsupervised on unlabeled data, fine-tuning supervised, multi-task mixture of supervised and unsupervised pre-training	seq2seq	C4 = Colossal Clean Crawled Corpus – new	text-to-text, NLP tasks, summarization	Oct 2019	Google
BERTSum	118 M	modify BERT as a document-level ->BERTSum, pre-train data on a BERTSum, fine-tune on encoder for extractive summary -> BERTSumExt, then fine-tune on decoder -> BERTSumExtAbs	seq2seq	CNN/DM NYT XSum	extractive/abstractive summarization	Nov 2019	Institute for Language, Edinburgh
Pegasus	Base: 223M Large: 568M	self-supervised pretrained for abs.summarization on large text, fine-tuning on 12 downstream datasets, and on zero-shot (10…) settings	seq2seq	C4 HugeNews – new	abstractive summarization	Dec 2019	Google
mBART	680 M	self-supervised pre-trained model, trained once for all languages. Multilingual. fine-tuned for any of the language pairs in supervised and unsupervised settings without task or language modification (mBART25, mBART06= pretrained on 25 or 6 languages)	seq2seq	CC25 – a subset of 25 languages extracted from the Common Crawl (CC)	translation, (summarization)	Jan 2020	Facebook
Longformer	Small: 41M large: 102M LED: 447M	pretrained, fine-tuned	decoder-only encoder-decoder (LED)	tairseq long document	LED is evaluated for summarization on long doc	Apr 2020	AI2
GPT-3	Small: 125 M Medium: 350 M Large: 760 M XL: 1.3 B GPT-3 175B: 175 B	unsupervised pretraining – model as GPT-2 in-context learning (ICL) during inference: few-shot (ICL) one-shot and zero-shot learning	decoder	Common Crawl, WebText2, Books1, Books2, Wikipedia - for pretraining, Lambada dataset for few-shot	NLP tasks, not summarization in paper	Jul 2020	OpenAI
ProphetNet		pretraining model	seq2seq	Base model: (16GB) BookCorpus (Zhu et al., 2015) and English Wikipedia. Large model: (160GB) -BookCorpus, English Wikipedia, CC-NEWS, OpenWebText, Stories	CNN/DailyMail, Giga-word, and SQuAD 1.1 benchmarks for abstractive summarization and	Oct 2020	Microsoft

					question generation tasks.		
CTRLSum	BART Large: 406M	Fine-tuning using ground-truth summary and automaticly extracted keyword	BART (seq2seq)	CNN/DM, arXiv, BigPatent	abstractive summarization	Dec 2020	Salesforce Research
BigBird		sparse attention mechanism on encoder side. Pre-training base model: BigBird+RoBERTa, pre-training large model: BigBird+Pegasus	seq2seq	ArXiv, PubMed, BigPatent, BBC XSum, CNN/DM	abstractive summarization of long documents	Jan 2021	Google
mT5	Small: 300 M Base: 580 M Large: 1.2 B XL: 3.7 B, XXL: 13 B	pre-trained on mC4 and fine-tuned on (1) English data (zero-shot), (2) (multilingual) machine translation from English to X, (3) training multitask on all target languages	seq2seq	Common Crawl (101 languages) mC4 (>100 languages)	NLP tasks	Mar 2021	Google
XL-Sum		fine-tuning mT5 model with large XL-Sum multilingual dataset	mT5 backbone (seq2seq)	XL-Sum covers 44 languages, high and low resourced - new	abstractive summarization	Jun 2021	Bangladesh University
GSum	BERT/BART	extract: highlighted sentences, keywords,salient triples, retrieved summary, then use them as a guidance in enc-dec training	BERTAbs, BERTExt BART, MatchSum (encoder+encoder-decoder)	Reddit, XSum, CNN/DM, WikiHow, NYT, PubMed	abstractive summarization	Jun 2021	Carnegie Mellon
SimCLS	BART large: 406 M Pegasus large: 568 M	generate-then-evaluate two-stage framework with contrastive learning, seq2seq model creates candidates, then evaluation model is trained to rank the candidates with contrastive learning.	BART and Pegasus (seq2seq)	CNN/DM XSum	abstractive summarization	Jun 2021	Carnegie Mellon
Gopher	44M 117M … Gopher: 280 B	Pretrained on unlabeled dana, then using fine-tuning, few-shot or zero-shot setting for different tasks (dialogue use fine-tuning/ few-shot prompt) Evaluate on MMLU and Big-bench	decoder-only (autoregressive)	MassiveText (MassiveWeb, Books, C4, News, GitHub, Wikipedia) - new	mathematics, logical reasoning, general knowledge, scientific understanding, ethics, reading comprehension, + NLP	Jan 2022	Google DeepMind
LaMDA	2 B … 173 B	pre-training on unlabeled text, fine-tuned on application-specific dialog. On scaling model - observing three key metrics: quality, safety, and groundedness	decoder-only	Infiniset- dialog data from public dialog data and other public web documents. It consists of 2.97B documents, 6.25% Non-English web documents. The total number of words in the dataset is 1.56T.	dialog	Feb 2022	Google
ST-MoE	ST-MoE-L: 4.1 B ST-MoE-32B: 269 B	Stable and Transferable Mixture-of-Experts (MoE) and Switch Transformers – design for energy efficient transformers. pretrain a sparse model and fine-tune it across NLP benchmark	seq2seq	Dataset from GLaM: reasoning (SuperGLUE, ARC Easy, ARC Challenge), summarization (XSum, CNN-DM), closed book question answering (WebQA, Natural Questions), and adversarially constructed tasks (Winogrande, ANLI R3)	reasoning, summarization, closed book question answering, and adversarially constructed tasks	Apr 2022	Google Brain
T0	3 B 11 B	Based on T5. Pre-trained in supervised and multi-task fashion, NLP tasks are converted into prompted form, to achieve better generalization, then use zero-shot task. Evaluate on Big-bench	seq2seq	Hugging Face datasets	NLP tasks: natural language inference, coreference, word sense disambiguation, sentence completion, + BIG-bench tasks	Mar 2022	Hugging Face, Brown

Chinchilla	70 B	Pretrained on unlabeled dana, then using fine-tuning, few-shot or zero-shot setting for different tasks (dialogue use fine-tuning/ few-shot prompt) Evaluate on MMLU and Big-bench	decoder-only (autoregressive)	MassiveText (MassiveWeb, Books, C4, News, GitHub, Wikipedia)	language modelling, reading comprehension, QA, common sense, MMLU, big-bench	Mar 2022	Google DeepMind
PRIMERA	447M	pre-trained for unlabeled multi-document summarization, fine-tuned on zero- few-shot and full-supervised settings	Longformer ED backbone (encoder-decoder)	Newshead	multi-document summarization	Mar 2022	AI2
BRIO	BART Large: 406M Pegasus Large: 568M	generate candidate summaries from generation model, and fine-tune BRIO model on candidate summaries using different datasets; few-shot fine-tuning	BART or Pegasus backbone (seq2seq)	CNN/DM XSum NYT	abstractive summarization	May 2022	Yale, Carnegie
OPT	125 M 175 B	unsupervised pre-training and evaluation using zero-shot, one-shot, few-shot settings on different tasks	decoder only LLM	BookCorpus, Stories, CCNews, thePile, PushShift,io Reddit	NLP tasks, not summarization	Jun 2022	Facebook Meta AI
GLaM	130 M 1.7 B … 1.2 T	feed-forward vs MoE (Mixture-of-Experts) layers. Pre-training. Zero-shot, one-shot, few-shot learning.	decoder-only	dataset of 1.6 trillion tokens Filtered Webpages, Wikipedia, Conversations, Forums, Books, News - GLaM dataset	NLP tasks, NLG benchmark: TriviaQA, NQS, WebQS, SQuADv2, LAMBADA, DROP, QuAC and CoQA.	Aug 2022	Google
PaLM	8 B 62 B 540 B	end-to-end model training multilingual pretrained, fine-tuned, few-shot language understanding and generation. Evaluate on MMLU and Big-bench	decoder-only	dataset based on LaMDA and GLaM, multilingual Wikipedia, filtered webpages (multilingual) code from GitHub (24 prog.lang) 780 billion tokens, 124 lang, 78% eng.	reasoning, translation,QA, mathematics NLG, summarization included	Oct 2022	Google
Galactica	125 M 1.3 B … 120 B	pre-training -> prompt pre-training (zero-shot task prompts) to boost performance and maximize generality -> instruction tuning -> fine-tuning	decoder-only	papers: arXiv, PMC, Semantic Scholar, PubMed, bioRxiv, … Wikipedia, Khan Academy, …, Common Crawl, GitHub code, …	scientific task, common sense reasoning, math	Nov 2022	Facebook Meta AI
Flan-PaLM	8 B 62 B 540 B	fine-tuning by scaling tasks, model size; fine-tune CoT, evaluate on Big-bench and MMLU	PaLM backbone (decoder-only) (or T5)	Muffin, T0-SF, Natural Instructions v2, Reasoning	multi-step reasoning, NLP tasks.	Dec 2022	Google
LLaMa	7 B 13 B 33 B 65 B	pre-training few-shot zero shot instruction fine-tuning evaluate on MMLU benchmark	seq2seq	CommonCrawl, C4, GitHub, Wikipedia, Books, ArXiv, StackExchange	common sense reasoning, QA, math, reading comprehension, code generation	Feb 2023	Facebook Meta AI
UL2	167M decoder 335M enc-dec scaling up to 20B for fine-tuning, few-shot, one-shot, zero-shot, ICL reasoning	supervised fine-tuning, in-context learning, chain-of-thought prompting and reasoning task, MoD,	UL2 decoder only UL2 encoder-decoder	C4	NLP tasks, language generation -> summarization	Feb 2023	Google

BLOOM	BLOOM-560M BLOOM-1.1B … BLOOM: 176B BLOOMZ: 176B	BLOOM is pre-trained on ROOTS. BLOOMZ = multilingual, multitask fine-tuned model. Evaluation: zero-shot and few-shot settings prompts	decoder-only	ROOTS corpus (46 natural languages, 13 programming languages) For fine-tuning BLOOMZ: P3 =Public Pool of Prompts is extended with more languages (xP3)	SuperGLUE, machine translation, abstractive summarization	Jun 2023	BigScience
Z-Code++	Large: 710	2 phase pre-training (NLU + supervised summarization), fine-tuned in zero-shot and few-shot settings. Deal with multilingual (5 lang), low resource summarization task and long sequences.	seq2seq	mC4 for multilingual (same as mT5)	abstractive summarization	Jul 2023	Microsoft

APPENDIX B

TABLE V. DATASETS FOR SUMMARIZATION TASK

Dataset	Num of docs	Avg len text (#tokens)	Avg len summary (#tokens)	Type
Gigaword	3.8 M	31	8	news
Newsroom	1.3 M	659	27	news
X-Sum	240 K	431	23	news
NY Times	655 K	530	38	news
CNN/ Daily Mail	300 K	781	56	news
Multi-News	56 k	2103	263	news, multidocument
XWikis	213,911	945	77	Wiki, cross-lingual
EUR-Lex-Sum	1,500 x 24	400-1 000 000	170-3000	legislative, cross-lingual
WikiMulti	157,014	1078	112	cross-lingual
WikiLingua	770k	391	39	cross-lingual
XL-Sum				news, multilingual
ML-SUM	1.5M+	812	34	news, multilingual
BookSum	436	112885	1167	books
BookSum (paragraphs)	142,753	160	41	books
WikiHow	230,843	101	42	Wiki
AESLC				conversation
DialogSum	13,460	131	14	conversation
SAMSum	16k	400	100	conversation
Reddit TIFU	122,933	Short: 342 Long: 433	Short: 9 Long: 23	conversation
WikihowQA	100 k			QA
BigPatent	1.3 M	3573	116	patents
Pubmed	133 k	3224	214	scientific
ScisummNet	1 k		151	scientific
ArXiv	215 K	4938	220	scientific