POLITECNICO DI TORINO

Master's Degree in DATA SCIENCE AND ENGINEERING

Master's Degree Thesis

Interpreting the output of Transformer-based architectures for text summarization

Supervisors	Candidate
Prof. GIUSEPPE RIZZO	JUAN J. MÁRQUEZ V.

APRIL 2023

Summary

Current state of the art models in autonomous summarization tasks utilize Transformer-based architectures trained on large corpora. The novelty of these architecture is in the use of the attention mechanism that links the generated output to input tokens and weighs more the parts that are more relevant for fitting the task. The interpretation of these attention scores provides a first step in interpreting the inner-workings of these architectures. However, recent studies demonstrated that attention scores are not exhaustive in providing an interpretation of the output of these systems. This thesis investigates the role of explainable mechanisms of transformer-based architectures designed for text summarization.

Table of Contents

List of Tables

List of Figures

Acronyms

AI

Artificial Intelligence

Ml

Machine earning

DL

Deep Learning

NLP

Natural Language Processing

Chapter 1

Introduction

Deep Learning models (e. g. transformers) still lack interpretability of their reasoning. It becomes a difficult challenge, giving the highly recursive nature of their architectures. For high risks tasks the necessity of understanding these models becomes more apparent.

The focus of this thesis is on the proposed algorithm in "Attention-based Clinical Note Summarization" [1], which leverages with extractive text summarization in the clinical context

1.1 Automatic Text Summarization Systems

As pointed out by [2]: "the main objective of an Automatic Text Summarization (ATS) system is to produce a summary that includes the main ideas in the input document in less space and to keep repetition to a minimum". In the case of a medical context, it can help professionals store patients' medical records only keeping relevant information about their hospital stay. In this way, time spent analyzing the patient's medical status can be reduced, improving the efficiency of the work.

1.1.1 Extractive Summarization

The pipeline of extractive summarization (ES) starts with a source document (or multiple) that will then be pre-processed to be passed to a scoring technique (e.g. transformer) that gives a particular score for each sentence in the document (or documents). After all the sentences have been documented, the summary is created by concatenating these extracted sentences.

The main advantage of this technique is that it produces highly accurate summaries "due to direct retrieval of sentences". In this way, [3] says, the reader does not have to worry about erroneous interpretations, which might occur in abstractive summarization. Another benefit of extractive summarization is that it is faster than its counterpart. On the other hand, a downside of this method is that the resulting summary could contain inevitable information redundancy, as mentioned in [4]. The lack of semantical coherence is also mentioned, like in the case of references or anaphora of excluded entities. For this study, we will focus on this type of summarization.

1.1.2 Abstractive Summarization

A similar pipeline can be modeled for abstractive summarization (AS). Analog to the ES pipeline, it starts with a source document (or multiple) to then perform the pre-processing step, preparing the input for the model (technique). But now, this model is set to generate a summary from its original input. Later, after some post-processing, the summary is done.

Generating a quality abstractive summary is not a straightforward task, it can have problems with out-of-vocabulary words, as well as problems regarding the repetition of certain characters [4]. Another limitation of this approach, highlighted in [2], is that its capabilities are constrained by the richness of its text representations, and the way they model the input text. As they allude: "Systems cannot summarize what their representations cannot capture". On the positive side, the output of this method is more similar to the manual summary [5], having more flexible expressions and more condensed sentences that avoid redundancy [6].

In the case of Transformers, sequence-to-sequence (seq2seq) text generation has been immensely impactful in recent years [7]. This is an example of how human-like AS can generate outputs.

1.1.3 Hybrid Methods

This type of approach combines the aforementioned strategies (ES and AS) to produce the summary. Now the pipeline incorporates the ES technique, after the pre-processing, followed by the AS, and finally the post-processing. According to [6], integrating these procedures results in complementary and improved performance. The authors in [8] propose a copy mechanism to extract some of the words in the original input to accompany the abstractive generation, outperforming significantly baseline methods on the CNN/Daily Mail [9] and Gigaword [10] dataset. As a

disadvantage, it generates a less quality AS since the generation depends now on the extraction instead of the input [2].

1.1.4 Metrics for evaluation

The evaluation of ATS systems can be divided into two different types: intrinsic evaluation and extrinsic evaluation [11]. The latter measures how useful can the output summary be for other tasks, as seen in [12]. Meanwhile, the former measures quantitatively, the quality of the generation. This assessment can be done using several techniques. However, it is important to first distinguish between reference summaries and generated (peer) summaries. Reference summaries come from human-generated short texts, a sort of manual label. Conversely, generated summaries refer to the ones automatically produced by the ATS's [13].

Most of the early studies in the evaluation of ATS's adapted Information Retrieval (IR) techniques, for instance, recall, precision, and F-measure. Evaluating these metrics between the peer summary and the reference summary. Using these techniques alone posted some problems. Nowadays, the most popular metric for automated summary evaluation is ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [13]. As the name suggests, this metric measures the overlap of n-grams between the model output and the reference divided by the total number of n-grams inside the reference, as described in equation 1.1.

$$ROUGE - N = \frac{Count_{match}(gram_n)}{Count_{match}(gram_n)} \tag{1.1}$$

This metric can be further divided into several types: ROUGE-N (N from 1 to 4) refers to the number of overlapping n-grams for the ROUGE, ROUGE-L which measures the Longest Common Sequence, ROUGE-S that measures consecutive n-grams in the reference, but non-sequential in the peer summary, among others.

For the particular case of this study, since there are no reference summaries, the ROUGE score cannot be measure. Instead, following the same evaluation metrics from [1], the objective metrics will be: the Kullback-Leibler divergence (KLD) and the Jensen-Shannon divergense (JSD). These metrics can measure the divergence between the probability distributions of words between the peer and the reference summary.

$$\mathbf{KLD(P||Q)} = \sum_{x=\zeta} P(X) log \frac{P(X)}{Q(X)} \tag{1.2}$$

$$\mathbf{JSD(P||Q)} = \frac{1}{2} KLD(P||M) + \frac{1}{2} KLD(Q||M), \quad M = \frac{1}{2}(P+Q) \tag{1.3}$$

1.1.5 Benchmarks

Some benchmark testing to measure current performance on different datasets has been done for both extractive and abstractive summarization techniques. Some of the most common datasets include: GigaWord [10], PubMed [14], Reddit [15], and CCN/Daily Mail [9].

Extractive Summarization

For the extractive summarization task, the benchmark model is currently HAHSum [16] for the CNN/Daily Mail dataset in terms of ROUGE score. This model uses Graph Attention Networks (GAT) [17] to make their prediction. The input text is disected to extract syntactical knowledge and create a graph, which is also further divided into three subgraphs, each is modeled as GATs in order to get better embeddings for each one of the sentences in the text, to then pass these embeddings to several classifiers (fully connected linear layers) that make the prediction of whether a sentence should be included in the summary. The architecture can be seen in figure 1.1.

Abstractive Summarization

Current first place model for this task on the CNN/ Daily Mail dataset belong to MoCa [18]. This model consist of two seq2seq tranformer models along with a '**Mo**mentum **Ca**libration' mechanism to better align the model scores with an evaluation model that ranks the candidate sample generated by the momentum generator. Common benchmark transformer models for this type of generation include: PEGASUS, BART and T5.

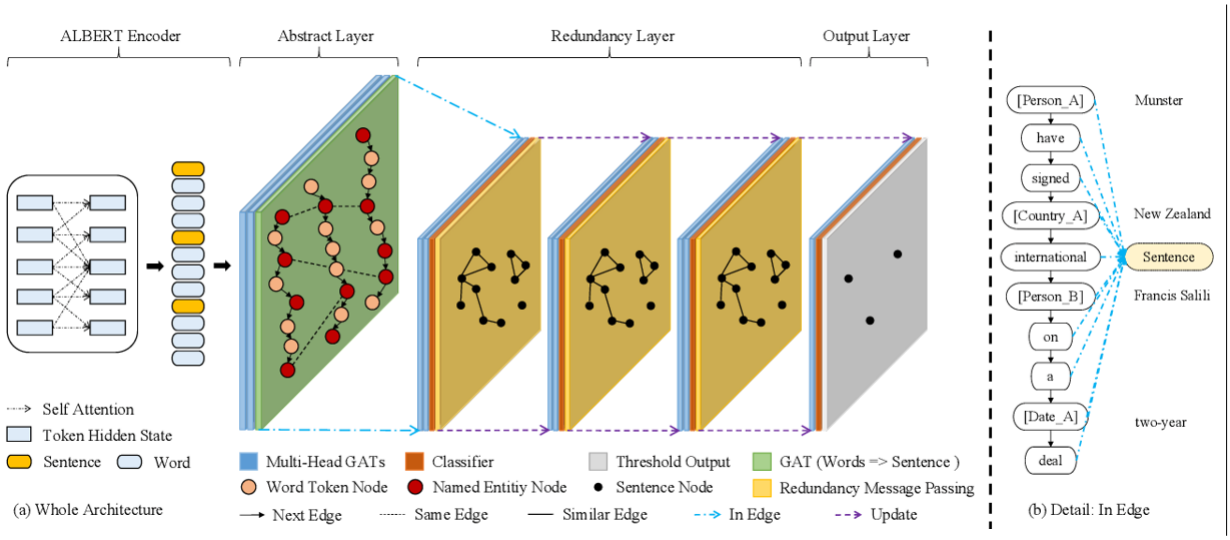

Figure 1.1: Architecture of HAHSum[16].

1.2 Text Summarization with Transformers

1.2.1 BERT

BERT stands from Bidirectional Encoder Representations from Transformers c[19]. It is designed to an embedding representation from text using left and right context across its layers. This model uses the self-attention mechanism of the Transformer architecture [20]. It is composed of 12 encoder blocks, each with 12 attention heads. It was trained for the tasks of Masked Language Model (MLM) task and Next Sentence Prediction (NSP) simultaneously. It does this by adding a CLS and SEP token at the beginning and end of the input sentence, respectively. An overview of its architechture can be seen in fig 1.2 .

1.2.2 BART

The BART model uses both an encoder and a decoder part to generate its outputs [21]. One important advantage of this model is its noising flexibility, "arbitrary transformations can be applied to the original text, including changing its length" [21]. This ability is learned thanks to the encoder part of its architecture that corrupts the input document by replacing spans of the text with mask tokens (as can be seen in figure 1.3. The decoder part of the the model computes the likelihood of the original document with its autoregressive decoder. A few variations of the model (or uses along with other models) have escalated to the benchmark of the

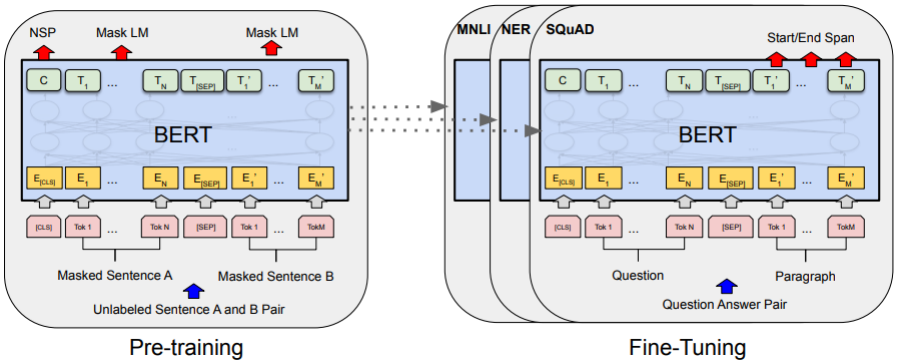

Figure 1.2: Overview of the BERT model [19].

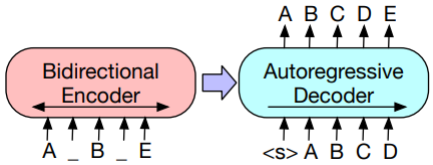

Figure 1.3: Overview of the BART model [21].

CNN/Daily Mail ranking, e.g. [22] [23].

1.2.3 PEGASUS

The PEGASUS model, introduced in [24], holds a similar concept to the BART model, using an encoder-decoder architecture. The main difference is that this model now performs the masking for "multiple whole sentences, rather than smaller continuous text spans" [24]. PEGASUS has been standing in the benchmark competition with several variations or combinations of this model with others, including the works in [25] [26].

1.2.4 T5

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer is the title of the paper describing the work of this model, in which the idea is to introduce the "text-to-text" framework [27]. The text-to-text problem consist in taking a text input and producing a new text output based on the former. In recent years there is growing change in the training paradigm, from "pre-train, fine-tune" to "pre-train, prompt, and predict" [28]. This means that for sentiment

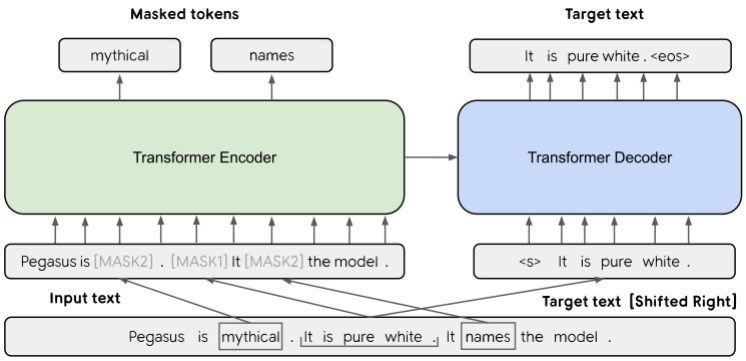

Figure 1.4: Overview of the PEGASUS mode [24]l.

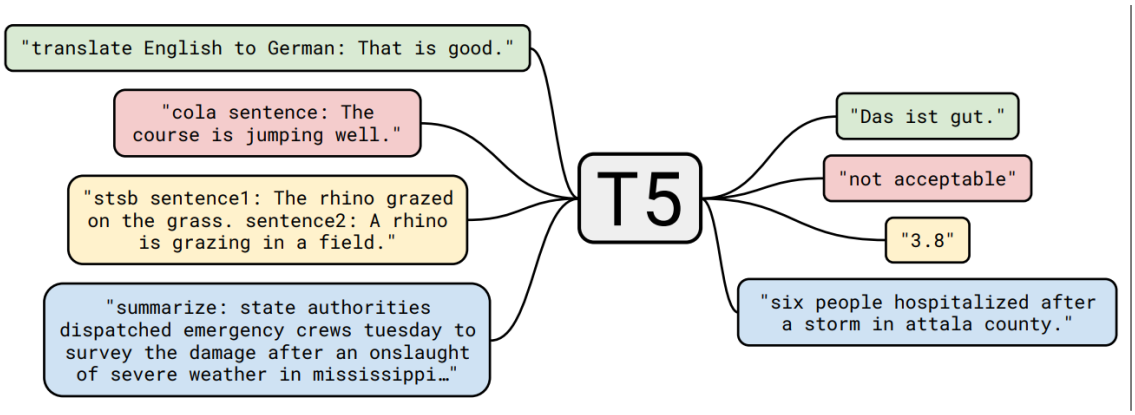

Figure 1.5: Overview of T5's text-to-text generation [27].

classification task, when having the input text "This is a bad movie" and output "negative", it is possible to modify the input to: "This is a bad movie, I feel _" adding the prompt " I feel _" and using a generative model (e.g. T5), which are usually trained with large training data, to generate the sentiment. Similarly, in the case of machine translation, the prompt can become "English: This is a bad movie, French: _", or for the particular case of summarization the prompt might be: "[X]; TLDR: _", where X would be the long input text. With this technique it is possible to use powerful Large Language Models for any task, if having the correct prompt.

1.3 Interpretability

The main goal of interpretability is that it is possible for people using machine learning models to understand why the model is outputting its results. In the past decade, with the increase of deep learning models, which consists of several layers and neurons, each with millions of trainable parameters, it gets harder and harder to track the reasoning process of these models. But still, it is harder to give a single definition of interpretability, as indicated in [29], the definition of "interpretability" is either universally agreed upon, but with no official defenition, or it is still ill-defined, arguing that is not a monolithic concept and that several ideas should also be disentangle before a definition can surge.

When defining interpretability an important question arises: why do we need interpretability? or in the same sense, why is it important? [29] spells out various desiderata that the interpretability research should demand, which are: Trust, Causality, Transferability, Informativeness and Fair and ethical decision making. As stated in [30]: "The systems must be transparent to earn experts' trust and be adopted in their workflow", but if we define trust, [29] stated, as being confident that the model is performing well, then there is no need for interpretability in the first place, since we already have metrics that confirm how well models are performing. Definitions for trust can also be quite subjective. The goal of ML models is not only to provide predictions for previously known data, but to give an idea of causal relationships inside the data, in this way interpretable models can help us better understand this possible causal relationships. Transferability refers to the ability of transfering learned skills to unfamiliar situations. Models should also provide useful information about the real world, either trough its outputs or via some procedure that gives additional information the human decision-maker. All of these must be satisfied while preserving fair and ethical real-world decisions.

[31] defends that the primary reason for the need of interpretability is curiosity and that the primary function of explanations is to facilitate learning. We can improve the learning phase of ML models by having a good understanding of their reasoning process. [31] also makes the comparison of these explanations to the social sciences field, where explanation might help enlight a shared meaning that can be crowdsourced from similar models.

1.3.1 Explainability vs. Interpretability

It is important to differentiate between the concepts of interpretability and explainability. The latter is more centered on a post-hoc model analysis, where there is

a predefined model and the goal is to explain how it is making its decisions by looking at how is treating the input. Whereas the latter's main question is: How can we create inherently interpretable models? [32]. In this case, there would be no need for a post-hoc explanation, the model's 'thought' process is already transparent since the beginning. [33] defends that: "There is a vast and growing body of research on posthoc explainability of deep neural networks, but not as much work in designing interpretable neural networks". In this work, the discussion between keep on creating posthoc explanations for complex uninterpretable models instead of using inherently interpretable models arises.

1.3.2 Accuracy vs. Interpretability

There is a common misconception that as ML models get more complex (e.g. DL models with large numbers of layers), the accuracy of the models gets better while the effectiveness of their explanations worse, since it gets harder to track the model's reasoning, thus creating a fictional trade-off between the accuracy of a model and its interpretability. This is not always the case, especially for structured data with meaningful features, in deed, accurate interpretable models might exist in many domains [33]. The challenge is to devise and use interpretable models that can also perform well in target metrics such as accuracy.

1.3.3 Transformer's Attention

[20] successfully showed that a model using only the attention mechanism, that was previously accompanied by RNN's and RNN-like architectures [34], reports state-of-the-art results surpassing the results of these previous implementations. Thusly, highlighting the importance of the attention in neural network architectures.

Types of Attention

Attention is not a monosemy concept, on the contrary, [35] classifies the type of attention into categories depending on the number of sequences, abstractions, positions or representations. According to the number of input sequences (i.e. input documents) the attention could be of the distinctive type (meaning the key and query values come from distinct sequences), co-attention (having multiple input sequences) or self-attention (most popular, key and query values come from the same input sequence). The number of abstractions classifies the attention into single-level (only computed once) or multi-level (a hierarchical approach, computationally costly). The number of positions selected for the computation of the

attention labels the type of attention into soft (considering all positions, tokens), hard (considering only a sample of the tokens) or local (using a window around a specific position). And lastly, the number of representations of the input can catalog the attention into multi-representational (having several representations for the same input) or multi-dimensional (computing relevance over every dimension of the input). Not all categories are applicable to all tasks.

Applications

In the literature several works have been reported using the attention mechanism and exploring the different aforementioned types of attention. For the machine translation task, [36] made use of a distinctive, single-level, soft attention mechanism for English-to-French translation. Also for machine translation, [37] presented a similar distinctive, multi-level, but local attention approach, this time for English-to-German translation. [38] used a co-attention, multi-level, soft attention model for sentiment classification of aspect and opinion terms extraction from user-generated texts. As final example, [39] created a self, single-level, multi-representational, soft attention BiLSTM to introduce dynamic meta-embeddings. In this manner, different uses of the attention mechanism have been displayed in the literature exhibiting the flexibility and utility of this component.

Transformer's Self Attention

The transformer's architecture makes use of a Self Attention mechanism for its encoder part (in models like BERT, RoBERTa [40], which are encoder only) and a Cross Attention mechanism for the decoder part (in models like GPT [41], which is encoder only). The attention mechanism was first propossed in [36] based on the biological idea that, visually, humans process their vission by ignoring certain parts of the images they see, irrelevant information. In parallel, certain words (tokens) of the input document might not be as relevant for the task at hand. This mechanism is centered around a query-key-value mapping to an attention distribution [35]. This attention distribution discloses the more relevant keys (token embeddings) with respect to a specific query (token embedding). In the self-attention case, the key and query vector embeddings are derived from the same input document.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \tag{1.4}$$

Equation 1.4 defines the computation of the self-attention mechanism used in [20], introducing a scaled dot product having a scaling factor $\frac{1}{\sqrt{d_K}}$, where d is

the dimension of the query, key and value vectors. According to the authors this alignment function ([35]) is "much faster and more space-efficient in practice". This type of approach can be sometimes memory consuming, which is why there is a limit to the amount of input characters (512 for the case of BERT-base and 1024 for BERT-large). Other authors have tried to overcome this limitation by introducing several transformer variants, that applied the same concept [42]. The authors also make use of a Multi-Head Attention mechanism, which they found beneficial and allows the model to jointly attend to information from modified subspace of the text representation within different positions [20]. This Multi-Head feature concatenates several (12 in the case of BERT) attention vectors (heads) and introduces an additional learnable matrix.

Attention in interpretability

The attention mechanism in transformers can give insights about token importance by analyzing the flow of the information throughout the network [43]. It becomes possible to deduct which tokens become more important for the model (applied to a specific task), as well as identifying pattern in the attention scores (e.g. in BERT)[44]. Some studies even proclaim that this scores can even reflect linguistic syntactical structure across several languages [45] . Combining all this findings with the input text to produce the output text is the main goal of interpretability using the attention. Nonetheless, there is still the question: is exposing and analizing the attention scores sufficient to provide an explanation of the model's reasoning?

Is attention explainability?

Transformer's attention mechanism has been debated in the literature as an explanation for itself [46]. The attention scores can describe which parts of the model are being focused on the most, but it does not tell you how the model is using those scores to make its predictions. For instance, the model in figure 1.6 shows the attention map of the model, displaying which parts of the images are being more used for the classification of the model. Unexpectedly, the model is selecting the right parts of the images (as a human would do), but it predicts that the image shows evidence of an animal being a Transverse Flute. From this, we can realize that saliency alone cannot be sufficient as an explanation. And indeed, this is what is discussed in [47]. Interestingly enough, [48] argues that attention can be an explanation if used rightly. They allude to the results purported in [46] claiming that their results call for further analysis and that whether attention alone is or is not explanation depends on the types of the desired explanation (e.g. faithful, plausible). Following this topic, [49] proposes word-level techniques to effectively

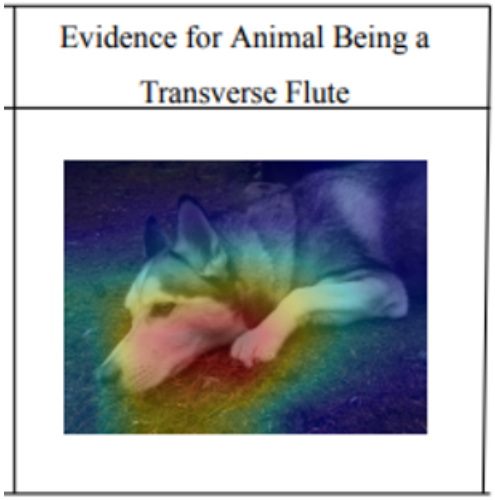

Figure 1.6: Missclassified sample using attention maps [33].

use attention as a source of explanation.

Attention vs. Saliency

Seemingly, using attention scores alone does not provide a reliable explanation. On that account,[50] proposed the use of saliency methods in lieu of these values. In this work, it is argued that saliency methods should be preferred over attention for explanation purposes. Saliency can indicate which words (input tokens, which can be also used for sentences) need to be changed to affect the model's score the most [51]. These methods have been proven to provide better results, as in the case of [52] where the authors were capable of generating word interpretations with better quality than attention-based mechanisms. Counterintuitively, saliency is not the ultimate solution for an explanation, it has its limitations and it is important to identify where and how to use these methods[53] .

Saliency vs. Sensitivity

There is a differentiation between the concepts of saliency and sensitivity, which is helpful to better understand which type of interpretability task are we performing.[54] defines sensitivity as a description of how the output changes when the input features are perturbed. On the other hand, saliency explanation methods describe a marginal effect of removed features on the output, when having the same input. Compared to saliency methods, sensitivity approaches are much faster,

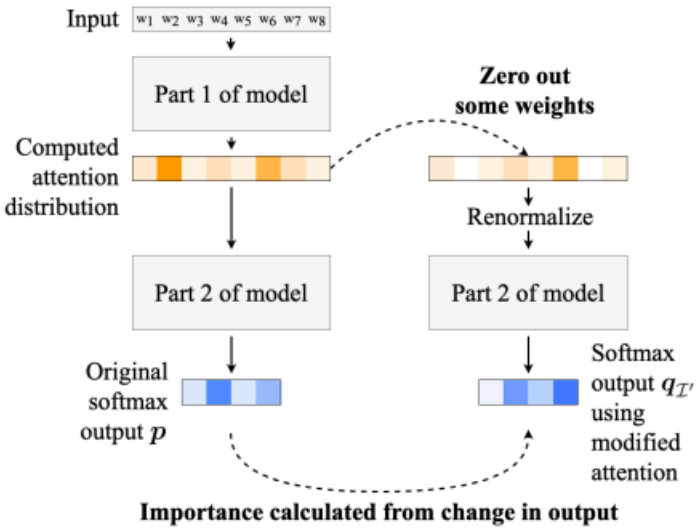

Figure 1.7: Example of zeroed-out attention weights approach for feature importance [58].

since they only require a few passes through the network, whereas its counterpart need to analyze the contribution of each individual feature.

Explanation methods

Examples of explanation methods can be found in the literature. Most common sensitivity approaches include LRP (Layer-wise Relevance Propagation [55]) used in [56] for visualizing hidden states of an RNN for Machine Translation, as well as in [57] in the case of stance classification from tweets using BERT; Occlusion (erasure) -based methods as the one presented in [58] where their method zeroed-out the attention weights to compute the importance of the representations (an expample of this approach is presented in figure 1.7); In a simmilar manner, Perturbation-based methods provide a slight change to the input features in order to analyze its importance as well as its resilience, which is a desirable property [59], like in the case of [60] and [61]. Some helpful tools that have used gradient-based saliency methods include: Ecco [62], SHAP [63] and LIME [64]. These tools will be further discussed.

Tokens vs. Words

The previously mentioned work of [57], makes the differentiation between input tokens and words for its analysis. In his work, the aforementioned BERT model is used. This model's pre-processing makes use of the WordPiece tokenizer [65]. These approach breaks words into *wordpieces* through a data-driven approach that provides better performances. When analyzing the input features of BERT, it is important to take this into account since the explanation might be on the tokens, rather than on the input words. [57] provides a clearer explanation of the contribution of the words by arguing that the "tokens themselves don have a meaning, and their relevance to the domain can only be analyzed if the context of the word they were contained is identified".

1.3.4 Metrics for interpretability

But how can we measure a model explanation? In the literature, several metrics have been defined for interpretability [66]. Some of these metrics include Readability, Plausibility, Human-interpretability, Persuasiveness [67] [68], Faithful interpretability (Hallucinations - Intrinsic, Extrinsic) [64] [69] [70] [71], Accountability [72], Trustworthiness [73], Descriptiveness [74] [75], Descriptive accuracy [76], Transparency [77], Fidelity [78], Robustness [79], Simulability, and Decomposability [29], among others. It becomes important, as well as a challenge, to define which type of metric(s) it is wanted to optimize.

Who are we explaining the model to?

When defining which metric is desired to optimize, an important factor is to know the type of user we are describing the explanation to. Most common user the explanation can be targeted to are: a layperson, a domain expert, a developer and even another software (machine). Each of these users may require different kinds of explanation, for some the explanation might be complex while for others the explanation are quite straight-forward. As remarked in [80]: "An explanation that is plausible to a domain expert may not be plausible to a layperson". In the particular case of the medical context, a domain person might be more inclined for more plausible explanations rather than faithful ones, since their main objective is not to obtain faithful information on the model's inner workings, but rather on the "human-like" reasoning of it. As a rather philosophycal juxtaposition, [29] makes the comparison between the black-box nature of our human brains and the post-hoc interpretability verbal explanations, explaining that this type of interpretations can explain the predictions made without giving an insight into the mechanisms

use by the model to derive such predictions, revealing an interesting contradiction between popular definitions of interpretability.

Faithfulness vs. Plausibility

In the case of faithfulness and plausibility (commonly used in the literature), there is an important differentiation. Faithfulness refers to how accurately the explanation reflects the true reasoning process of the model. Instead, plausibility refers to how convincing the interpretation is to humans. For example, as stated in [81] the top-k attention weights for each token can provide a plausible, but not always faithful. Conversely, the information flow process of the attention mechanism is a faithful explanation but it differs from human reasoning for the same kind of task. A good explanation model should be able to perform well in one of these metrics.

1.3.5 Approaches for post-hoc interpretations

Post-hoc explanations can come in different forms. [82] defines three main types of explanations: Natural language explanations, visual explanations, and explanations by example.

Natural Language Explanations

Natural language explanations focus on presenting the explanation in a textual form. An example of this can be the top_k words from a topic in a topic modeling task.

Explanations by example

Explanations by example can show the similarity between the model's reasoning with another model that was trained on a similar task. As [82] puts it: "This sort of explanation by example has precedent in how humans sometimes justify actions by analogy. For example, doctors often refer to case studies to support a planned treatment protocol."

Visualization explanations

And last but not least, visualizations. This type of explanation can be the most appealing type for users since it provides a different way to portray the model's

output. In this category, t-SNE [83], attention-saliency maps, and other visualization tools (e.g. LIME, SHAP, etc.) come into play. One of these tools, SHAP [84], tries to define an additive function where it is possible to assign a contribution score to each of the input's features. In this way, you can discriminate each of the model's features by its contribution to the output.

Visualization Tools

Commonly developed visualization tools aim at creating a representation of the models' explanation. Some of these tools are model-agnostic, meaning that they can fit to any model, without any kind of restriction. Others are particularly design for a specific model, optimizing the explanations to fit this model. In the field of textual data, some aplicable tools are: LIME [64], Anchors (a variation of LIME) [85], Neat-Vision, HotFlip (adversarial explanations) [86], BERTviz [87], Ecco [62], T3-Vis [88], DeepNLPVis [89] and SHAP [84].

Chapter 2

Background

2.1 Research Questions

2.1.1 How can transformer-based Automatic Text Summarization (ATS) systems benefit from explainability?

As discussed in 1, explainability can bring various benefits to any model. It becomes possible to produce a human-in-the-loop framework, creating a collaborative relationship to produce more efficient predictions, including domain experts and providing transparency and trust [90]. Also, including an explainability framework into these systems in situations where there are high-risk tasks (like in the case of the clinical context or law context) transparency becomes a more important addition, as in the case presented in [80], [91].

Infusing domain knowledge into the summarization task in the medical context has been proven to yield good results in the work of [91]. This study discloses the utility of including domain knowledge features into the summarization task, as well as including an explantion framework based on this domain knowledge infusion. In this way, it is possible to include domain experts into the analysis of the model. This is similar to the work in [92], where domain experts where able to be included in the evaluation of the multi-head attention modules in the transformers architecture. The human-in-the-loop pipeline allowed human experts to identify patterns in the attention mechanism for the task of extractive summarization. For this study, the authors also used the T3-vis[88] visualization tool to help illustrate the attention workflow of the model. Both of this approaches further support the idea of integrating explainability techniques for the ATS task and bringing benefits, such as including domain experts into the improvement of the system.

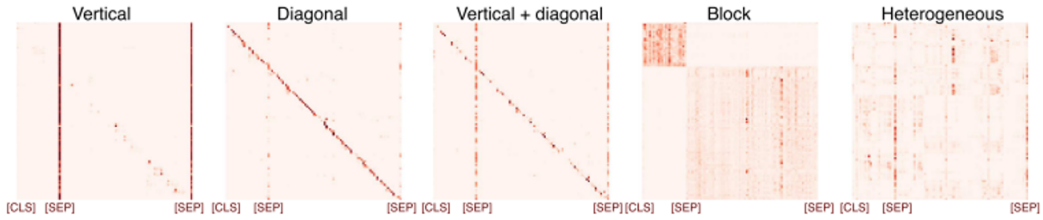

Figure 2.1: Attention patterns present in BERT architectures [44].

The MIMIC-III dataset [93] provides extensive clinical notes of patients describing vital information that can be later used by doctors and medics for alternative uses (e.g. clinical trials). Due to the extension of these clinical notes, the necessity to extract relevant information becomes more apparent. The proposed summarization approach in [1] does not provide explanations of the inferences made to produce the summaries. By all means, it is blindly making use of the black-box nature of a BERT-based model. As stated in chapter 1, this *modus operandi* can come with several consequences and would definitely benefit from interpretability.

2.1.2 What is the impact of the attention mechanism and the models' hidden states on the explainability of transformer-based architectures?

On the genesis state of the learnable weights of Transformer-based architectures, these values are initialized at random, once the model starts adjusting these parameters input sentences move differently across the network. The impact of these changes compared to the randomly initialized values is noticeable [94]. From this, we can understand that there exists a big impact of the finetuned attention weights, making explanations based on attention scores more susceptible to these changes.

Different types of attention patterns within BERT's architecture arise and are noticeable, [44]. As it is explained in this work, an analysis of the attention mechanism structure of BERT suggests that there is redundancy in the information encoded in these modules, indicating an overparametrization of these models (even in the smaller base version). This predictability might suggest an additional caveat in the explainability of attention scores. An example of these patterns can be seen in figure 2.1. It becomes also important to analyze the attention scores at different parts of the network, some of these parts can provide different insights into the inner workings of the model [43].

In the particular case of clinical note summarization, the attention scores are

Figure 2.2: Pipeline of explainability as a communication problem [81].

being used to extract sentences that will become part of the summary. In this way, the impact of the attention mechanism is quite large. Understanding this attention mechanism and how it behaves helps improve the performance of the original model. Identifying the tokens that are more important for the model's sentence selection is not an easy task and it can be accompanied by domain experts (in the case of the medical context) along with software developers to more accurately recognize these features and adjust the model accordingly.

2.1.3 What is the proper (explainable) way to present the output of transformer-based models in relation with the inputs?

In explaining the output of transformer-based summarizer models, the output-input relation should be also taken into account. [81] models explainability as a communication problem where a layperson should be able to reconstruct the output of a model from the provided explanation, and this reconstruction should be done accurately. Figure 2.2 depicts the pipeline of the desired explanation methodology.

Regarding the ES task, this pipeline could also be applied. It is conceivable to view ES as a classification problem, where the model should be able to detect whether or not the analyzed sentence should belong in the generated summary. In a similar fashion, [91] includes domain knowledge interpretations. A sneak peek of this work can be seen in figure 2.3, where fragments of the output are classified into the clinical PICO framework (Population, Intervention, Control, and Outcomes). By having this classification, domain experts can more easily identify portions of the summary output, as well as incorporate domain knowledge into the model's reasoning.

Another common approach that can be used for explainability is to train an alternative inherently interpretable model (a.k.a. transparent) on the predictions of another uninterpretable mode, but with good predictive performances, as a

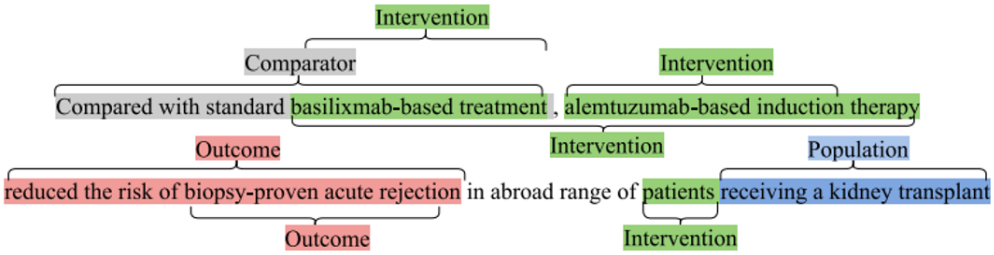

Figure 2.3: Example of the PICO framework application on a summary [91]

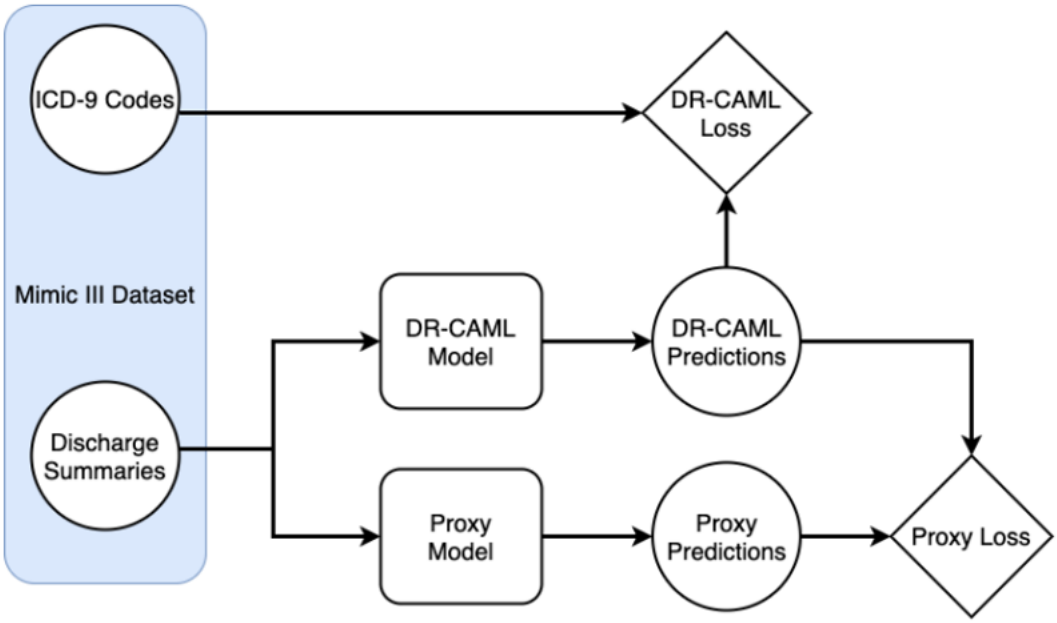

Figure 2.4: Training setup for proxy model explanations [80].

sort of proxy model for the task, optimizing the explanation for faithfulness. An example of this methodology is found in [80]. In this work, the idea was to train a series of linear regression models (one for each class) from clinical texts using a bag-of-words pre-processing on the predictions of the DR-CAML CNN model presented in [95]. This procedure reduces significantly the number of training parameters and provides an interpretable model similar to the visualization tool LIME [64], but making it able to explain the entire dataset instead of just one single input text. A workflow of this model's training setup can be seen in figure 2.4.

A plethora of techniques can be used to provide faithful and plausible explanations for NLP tasks. In the case of clinical notes summarization, the presented output of explanation models should also balance the results on these metrics making them readable for domain experts who are generally not involved in the development process of these systems, but rely on them for their work.

Chapter 3

Method

3.1 Problem description

Sumly

The *sumly* algorithm presented in [1] creates an extractive summary of the clinical notes, based on the attention scores of the BERT model finetuned for the classification task of the ICD-9 code labels. The clinical notes for the training are taken from the MIMIC-III dataset, that includes the annotated ICD-9 codes of each clinical note, but does not include an annotated summary. The challenging task of their study is to provide an unsupervised trained model that is able to extract relevant information from the clinical notes. The main idea was to use the attention scores from the BERT model. The reasoning behind this approach is that these scores are meant to highlight relevant input tokens for the task at hand.

The algorithm breaks the input clinical note into sentences. Each sentence is then passed through BERT where the last layer of the first attention head is captured. The aggregated attention score of the [CLS] token is used to measure the significance of the sentence in the clinical note, knowing that this special token can capture the attention score of the whole sentence. Once every sentence's attention score is calculated and saved, sentences with a higher attention score than the average attention score are selected to be included in the summary. As presented in figure 3.1, the algorithm passes each sentence to the model to then extract the attention scores. The highlighted senteces represent the sentences that will be used for the final summary.

Once the summaries are generated, the evaluation is done by applying the same metrics as described in 1. Both JSD and KLD, measure the divergence in the

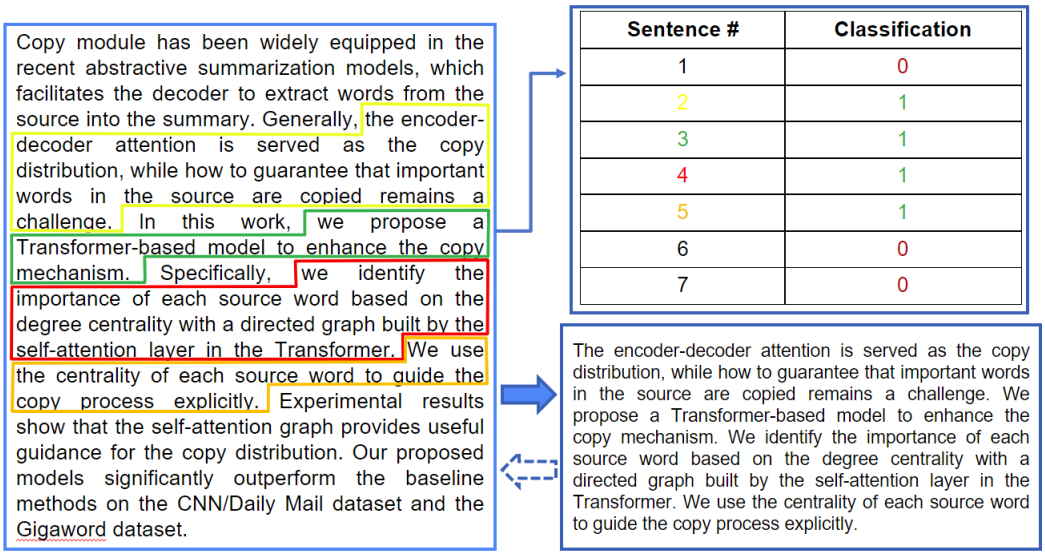

Figure 3.1: Example of *sumly* sentence extraction process.

probability distribution of words between the original clinical note and the manufactured summary. Lower divergence values are preferred, indicating a more similar word distribution, inferring that the crafted summary still retains the information presented in the clinical note.

This algorithm does not provide any kind of explanation for its reasoning process, since, as discussed in 1, it is not possible to use raw attention weights as an explanation. Furthermore, the clinical context belongs to a high-risk task designation, where there is a need for interpretability of complex models used in this environment. From this, it is possible to identify some limitations of the original *sumly* algorithm that can be improved.

Limitations

One of the most noticeable limitations, and one that has been discussed, is that it can benefit from interpretability techniques. As a start, the use of visualization tools can provide interactive explanations, useful for bringing domain experts into the development loop. A tool like SHAP(discussed in previous sections) can provide insights into the reasoning process of *sumly* for ATS to possibly tweak the algorithm to provide more efficient results.

Interpretability is not the only limitation of the algorithm. The input text

is fragmented into sentences, where each sentence is further passed individually through BERT. This individual passing ignores sentence relations inside the input text. This can be harmful to the efficiency of the algorithm and some analysis can be performed on this token. This work is centered around the clinical context and does not escape from this topic. Some studies can be done to evaluate the applicability of the algorithm in alternative contexts. Moreover, only the BERT model is used for this task. Experimenting with alternatives version of BERT (or even Transformers) brings a bigger scope of the real applicability of the algorithm. This can be done since all the Transformer architectures share an attention module to perform their task.

Proposals on improvement

For the use of visualization tools, the proposed explanation interface is SHAP. SHAP computes an additive function that can measure the contribution of every input token for every sentence passed through the model. One of the benefits of SHAP is that is model agnostic, meaning that is fittable to any ML model, including any variation of the BERT model or Transformer-based model, in the case of wanting to experiment with various distinct models. It also provides a handy and accessible API with proper documentation, easy to adapt to NLP models. This type of explanation is a wrapper method, that greedily tries to evaluate combinations of the input tokens to calculate the contribution of each token to the final output (in this particular case, to select whether a sentence should be selected to be in the final summary). By using this tool domain experts can easily identify the tokens that are contributing more or less to the task and help data scientists make proper changes to the code. It is possible to plot the contribution of local individual samples as well as the global contribution of the tokens for all the input dataset. For the present circumstances, the clinical note is still divided into sentences and each sentence becomes an input sample for the model. Thus, global explanations become global in terms of a specific clinical note dissected into sentences.

In addition to the interpretability analysis, a parallel study is presented finetuning a RoBERTa [40] model for the multi-label classification of the ICD-9 codes, following the work in [95]. The model is set to classify the top 50 most common ICD-9 codes from the clinical notes. Two RoBERTa-based models were used for the finetuning: the first one is a pre-trained model from the HuggingFace platform called *"minhpqn/bio_roberta-base_pubmed"*, and the second one is a distilled version of RoBERTa that follows the same principle presented in [96]. The distilled version was able to provide a much faster training process (which is already onerous due to the magnitude of the dataset and the nature of the multi-label prediction

task). The results of the classification task were favorable compared to the F1-score reported in [95].

It is important to highlight that the main objective is not to train a good classifier for the ICD-9 code prediction, but rather that from this process the model can adjust the proper weights for the summarization task. Contrarily, the *bio_roberta-base_pubmed model* was used to compare the visualization analysis of the SHAP tool with the original *sumly* model. As well as using this model to test the summarization task with different datasets. The following datasets were taken from the HuggingFace hub for testing: "tweet_eval", "amazon_polarity", "yelp_review_full", "rotten_tomatoes", and "poem_sentiment". These datasets were selected because they are specifically for the classification task. In this way, we can follow the same procedure proposed in [1] finetune for the classification task, extract relevant sentences based on the attention scores, and then construct the extractive summary. The difference is in the type of model and dataset used.

3.2 Dataset

3.2.1 Description

The MIMIC-III dataset [93] is a collection of 26 tables with anonymized data containing information about admission to the Beth Israel Deaconess Medical Center in the city of Boston, USA. It is a publicly available (for researchers) dataset widely used for a large number of tasks, not only on NLP. The textual descriptions inside the dataset (e. g. clinical notes) are extensive and descriptive of different processes of patients, for instance, their admission to the medical center, follow-up treatments, and procedures. In addition to textual descriptions, it also stores the International Classification of Disease 9th Edition (ICD-9) codes. These codes describe disease classifications useful to identify specific entities inside the textual descriptions. A paramount of studies have been made to create models that can accurately identify these codes inside a textual input (some of these have been presented in section 1).

The dataset includes a considerable amount of imperfections and information that are not useful for model analysis, making it evident the need for pre-processing. A cleaning on the textual fields of the dataset was performed, as well as a depuration of the number of ICD-9 codes to be analyzed for the finetuning following the procedures in [95]. Figure 3.2 shows the distribution of the frequency of all ICD-9 codes inside the dataset.

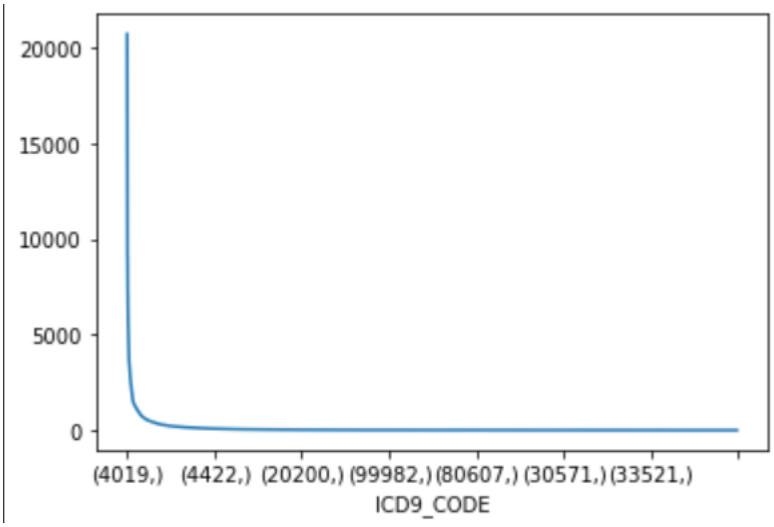

Figure 3.2: ICD-9 codes frequency distribution for all codes.

The dataset consisted of a total of 1.801.852 distinct possible ICD-9 codes. The pre-processing consists of transforming the text inside the clinical notes to lowercase and removing some special characters. As expected, most of the ICD-9 codes were used rarely while few were used more frequently. This is why, for the multi-label classification task, the top 50 most frequent ICD-9 codes were chosen. The distribution of the frequency of the top 50 codes is shown in picture 3.3. From the figure the more balanced frequency distribution is clear. One single clinical note can consist of several ICD-9 codes. The distribution of the number of ICD-9 codes for every individual clinical note can be seen in figure 3.4.

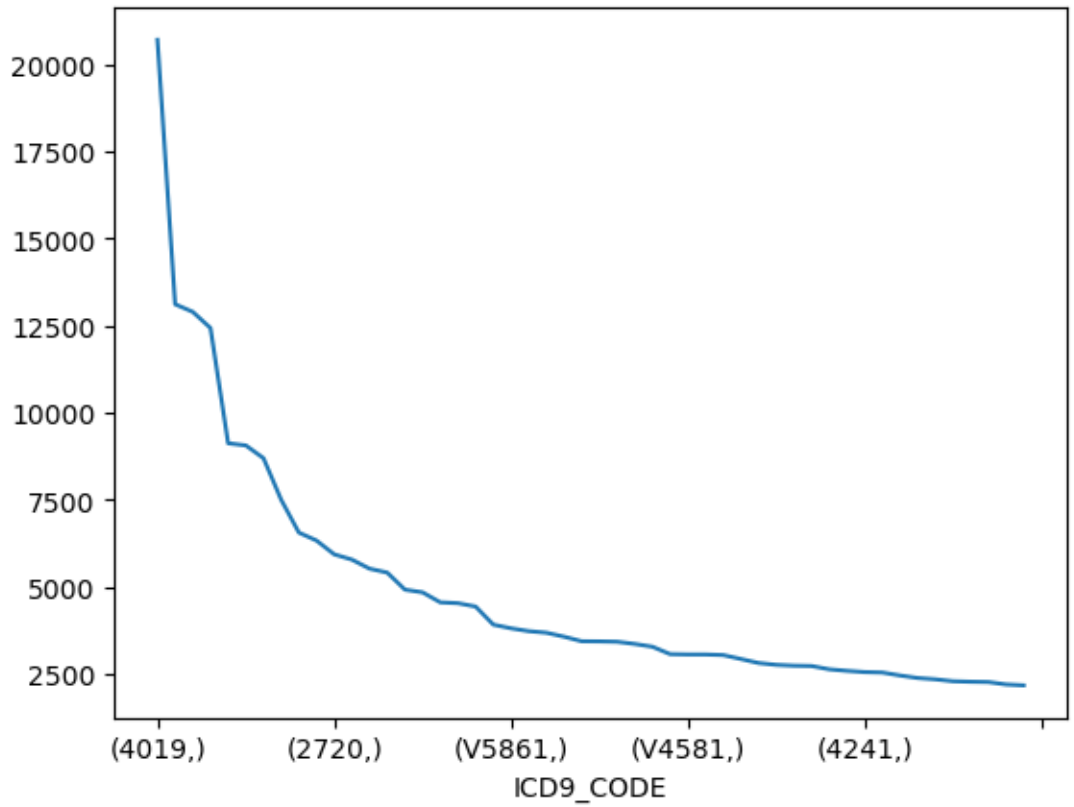

Figure 3.3: ICD-9 codes frequency distribution for the top 50 most common codes.

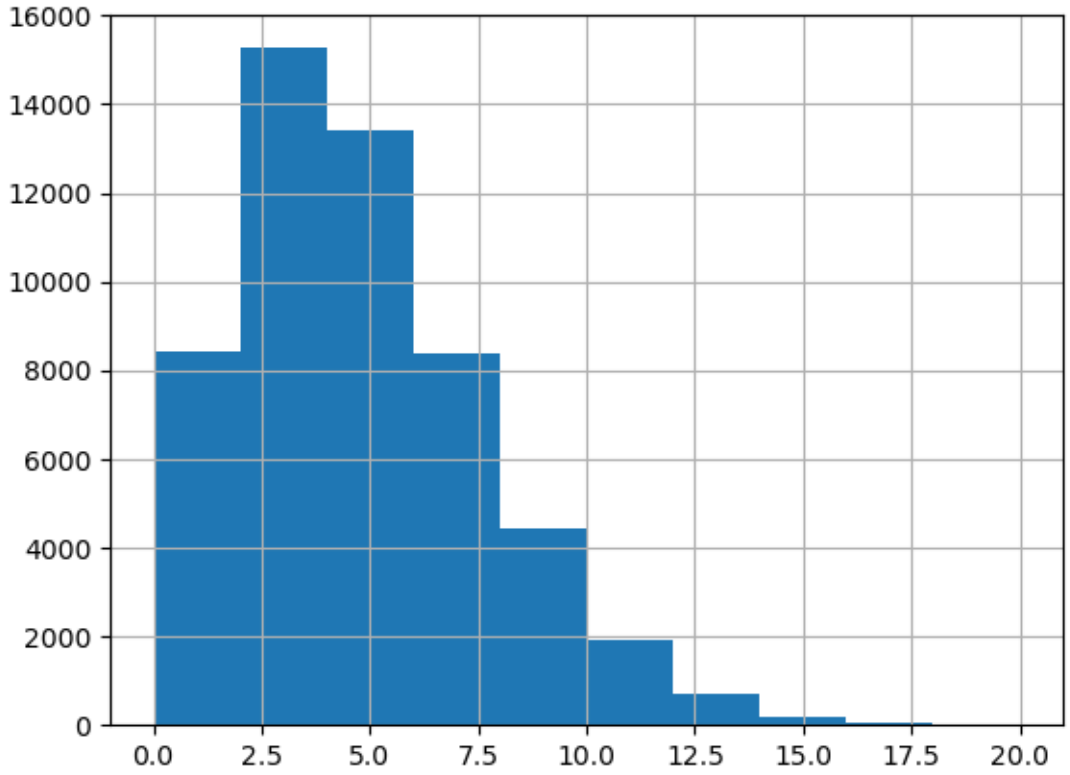

Figure 3.4: Distribution of the count of medical codes inside each clinical note.

Chapter 4

Experiments

4.1 Setup

As discussed in section 3, the SHAP explanation tool is used to analyse the *sumly* algorithm to identfy which input tokens are driving the output of the algorithm to the 1 (select the sentence for the summary) or 0 (not select the sentence for the summary) value for each input sentence. The way this algorithm operates is by first masking (replacing an input token with the [MASK] token) all of the token in the input sentence, and then trying different combinations of the original input tokens to calculate an additive function that describes the contribution of every individual token. After the masking, it computes a base value, which is then compare to the models output value. Arrows in the output represent to which part of the output line each token is driving the output value to. As an example, having three tokens were the base value is 3, the output value 2 and the additive contribution function is defined as follows: $base\_value(3) + token1(2) + token2(5) + token3(-8) = 2$. In here tokens 1 and 2 are driving the prediction towards the positive side of the output, while token 3 is pushing more weightly to the contrary side.

Two RoBERTa were used for comparative analysis of the task. The first model (distilled RoBERTa) was finetuned using the HuggingFace library API, which offers an accessible framework for NLP tasks (as well as other modalities). This model was finetuned using the pre-processed MIMIC-III dataset, for the prediction of the top 50 medical codes. Another pretrained bio_RoBERTa model was used for the evaluation of the summarization technique presented in [1] using other datasets, utilizing also the HuggingFace libraries. This model was also used in the same setting as the original BERT model for the interpretability task.

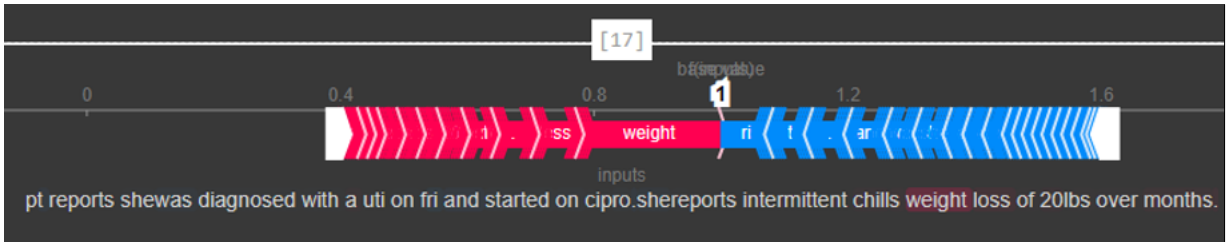

Figure 4.1: SHAP output for finetuned BERT model.

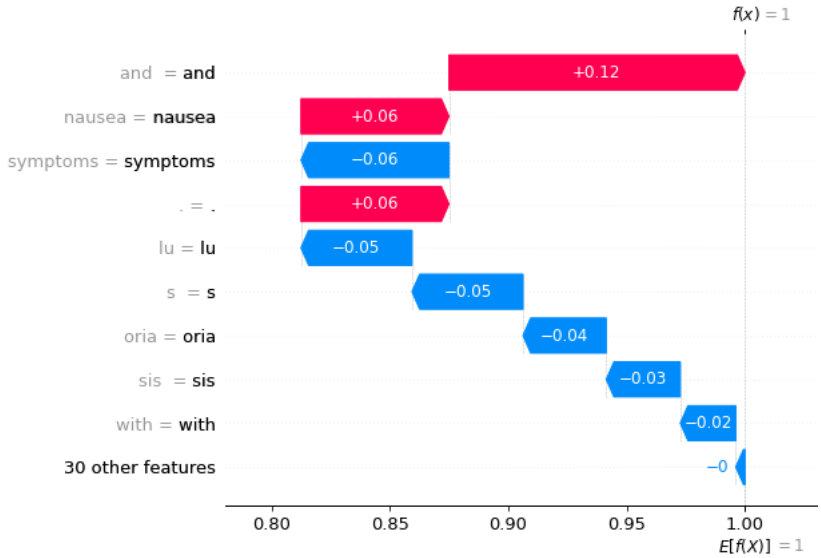

Figure 4.2: SHAP waterfall plot.

4.2 Results

Figure 4.1 reports the result of using the SHAP tool along with the *sumly* algorithm. In the illustration, the base value of the explainer is shown (corresponding to the 1 value), the output of the model is the $F(x) = 1$ value, and at the bottom of the output the passed example sentence is set where the hue of the tokens' background color represents the contribution to the token to the positive value (1, represented by red color) or to the negative side (towards the 0, blue color). In this way, the color represents the direction each token is driving the final result to. The number at the top center side of the image represents the sentence position inside the clinical note.

Figure 4.2 is an alternative plot for the results of the explainer. In this plot, tokens' contributions inside the sentence are shown in a waterfall shape. The additive contribution of the most relevant tokens is shown in the picture. These

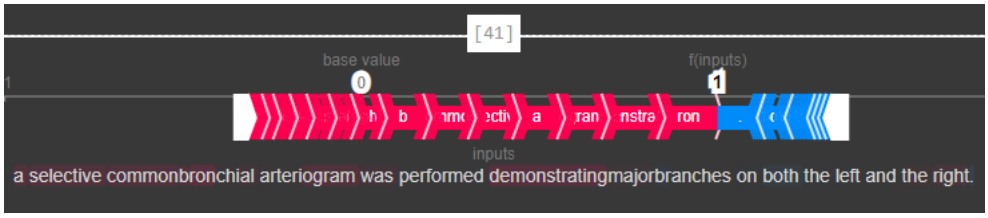

Figure 4.3: SHAP output #1 for finetuned RoBERTa model.

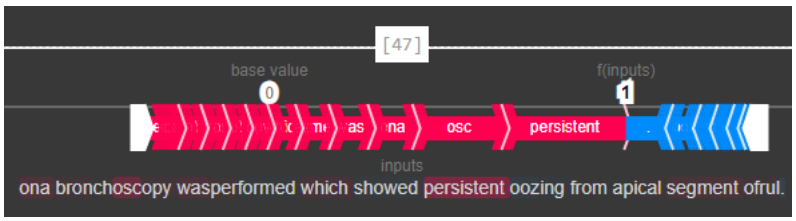

Figure 4.4: SHAP output #2 for finetuned RoBERTa model.

images represent how the output value of the model moves according to the contribution of each token. Again, the red color represents tokens driving the output toward the positive side, and the blue color, in contrast, represents tokens driving the output toward the negative side of the output line.

Figures 4.3 and 4.4 depict the output of the SHAP model, but now using the finetuned RoBERTa model in contrast to the original *sumly* model. The images follow the same logic as the aforementioned results in figure 4.1. As a remark not highlighted in the picture, it is possible to interact with the output of the explainer by clicking on each individual token and showing the contribution to the additive function. This corresponds to a better user experience that can be helpful for domain experts that might not be familiar with the finetuning and developing process.

The finetuning process of the pre-trained RoBERTa model can be seen in table 4.1. At each step of the training process the accuracy on the classification task, the mean and standard deviations of the divergences of the word distributions between the summary and the clinical notes are reported. This process was made for 3 epochs having a size of 4 for each batch and a weight decay of 0.01.

For the distilled RoBERTa model the finetuning results are reported in table 4.2. The finetuning was made for 10 epochs, showing increasingly results for the F1 measure and following the procedures on [95].

Step	Accuracy	Mean KLD	Mean JSD	SD KLD	SD JSD
50	0.636	1.859	0.459	0.334	0.050
100	0.631	0.895	0.266	0.224	0.057
150	0.636	0.897	0.266	0.222	0.056
200	0.782	0.912	0.271	0.210	0.052
250	0.754	0.950	0.280	0.209	0.051
300	0.826	1.072	0.308	0.251	0.058
350	0.843	0.979	0.287	0.212	0.051
400	0.865	0.979	0.287	0.221	0.053
450	0.882	0.961	0.283	0.213	0.051
500	0.882	0.967	0.284	0.202	0.049
550	0.893	0.941	0.278	0.195	0.048
600	0.893	0.929	0.275	0.201	0.049
650	0.905	0.919	0.273	0.201	0.049

Table 4.1: Training results for the finetuning process in different contexts.

Step	Training Loss	Validation Loss	Macro F1
1000	16.473600	16.479305	0.348268
2000	16.193800	16.183510	0.398516
3000	15.956800	16.081776	0.419651
4000	15.896000	16.019419	0.429194
5000	15.836200	16.005844	0.436733

Table 4.2: Training results for the finetuning process of the multi-label classification of ICD-9 codes using a distilled RoBERTa model.

Chapter 5

Analysis

The interpretability experiment for the *sumly* algorithm yields some interetinganalyzing results. By analizing a particular clinical note, sentence by sentence, it is possible to detect some limitations and improvements that can help improve the model. Figure 4.1 shows the output of SHAP, here we can observe that the base value for the model for every sentence is 1 (selecting the sentence for the model). From this, e can already notice a problem, because this means that even just masking every token in the sentence (i. e. transforming the sentence into "[MASK][MASK][MASK][MASK][MASK][MASK]...") the model stills decides to select this sentence for the final summary. When more tokens are later added to the sentence, is when the model takes the decision, based on the attention scores, to discard the sentence. Apart from this fact, the tokens that are contributing to the model's final output are not generally words that belong to medical terms, but instead, some tokens corresponding to breakdowns of a word (as already discussed in section 1). These wordpieces seem to be driving the model into the positive label of the classification. This is problematic since these words hinder the model to detect more context-relevant words to create the summary.

Figure 4.2 shows a waterfall plot generated by SHAP as an alternative representation of the token contribution to the output. For another particular example, it is possible to notice which tokens are contributing the most and to which direction of the final result. Tokens "and", "nausea", and "." are pushing the output of the model towards the $f(x) = 1$ (select the sentence for the summary), while other words such as "symptoms", "lu" or "oria" are pushing the output to the contrary side of the output line. It is noticeable to identify which tokens are causing problematic results for the model. The tokens "and" and "." should not be important tokens in order to discretize whether a medical sentence is important enough to belong in the summary of the clinical note. The same applies to the wordpiece tokens "lu", "s" or "oria" that don't represent by themselves any significant meaning (as

discussed in section 1).
The interpretability experiment for the *sumly* algorithm yields some interesting results. By analyzing a particular clinical note, sentence by sentence, it is possible to detect some limitations and improvements that can help improve the model. Figure 4.1 shows the output of SHAP, here we can observe that the base value for the model for every sentence is 1 (selecting the sentence for the model). From this, e can already notice a problem, because this means that even just masking every token in the sentence (i. e. transforming the sentence into "[MASK][MASK][MASK][MASK][MASK][MASK]...") the model stills decides to select this sentence for the final summary. When more tokens are later added to the sentence, is when the model takes the decision, based on the attention scores, to discard the sentence. Apart from this fact, the tokens that are contributing to the model's final output are not generally words that belong to medical terms, but instead, some tokens corresponding to breakdowns of a word (as already discussed in section 1). These wordpieces seem to be driving the model into the positive label of the classification. This is problematic since these words hinder the model to detect more context-relevant words to create the summary.

Figure 4.2 shows a waterfall plot generated by SHAP as an alternative representation of the token contribution to the output. For another particular example, it is possible to notice which tokens are contributing the most and to which direction of the final result. Tokens "and", "nausea", and "." are pushing the output of the model towards the $f(x) = 1$ (select the sentence for the summary), while other words such as "symptoms", "lu" or "oria" are pushing the output to the contrary side of the output line. It is noticeable to identify which tokens are causing problematic results for the model. The tokens "and" and "." should not be important tokens in order to discretize whether a medical sentence is important enough to belong in the summary of the clinical note. The same applies to the wordpiece tokens "lu", "s" or "oria" that don't represent by themselves any significant meaning (as discussed in section 1).

The results from the finetuned RoBERTa model show some improvement to the original *sumly* algorithm. An example of SHAP's output using the finetuned RoBERTa model is shown in figure 4.3 Starting from the base value of the explainer that has changed to 0 (not selecting the sentence for the final summary). This provides a more plausible explanation since this means that having a complete masked sentence, this sentence should not be selected for the summary, then when relevant tokens for the summary are added to the sentence, the decision (if important for the summary) can be shifted to the positive value. It is also possible to observe that more meaningful clinical words are contributing more to the selection of the sentences as in the case of "bronchial arteriogram". Some wordpiece tokens are still

hindering the model's results, in this way it is possible to perform a further analysis on the preprocessing part and repeat the process and in this way ameliorate the performance of the model. A second example can be seen in figure 4.4, where is also important to highlight that words that don't belong to the medical context can still be important for the final summary. As in the case of "persistent", this word might indicate a recurrent disease that a patient is suffering that needs to be accentuated in the final summary.

An analysis both on the side of the domain expert and the developer side must be performed. Combining these two examinations, using the explanations provided, to interpret the inner workings of the model provides a wider overview of where changes can be made to the model.

In addition to the interpretation part, the results of the finetuning are reported in table 4.1. As the model is finetuned for the classification task on other context datasets for the task of classification with long textual inputs, the accuracy of the model's classification increases while the mean divergence metrics become less and more consistent. At each step, a batch of textual inputs are passed to the model, and the attention scores are being evaluated to create a summary of this texts, following the *sumly* procedure. The divergence between the probability distributions of the summary and the text is aggregated and reported. This shows that the approach proposed in the original work is also applicable to other datasets that follow a similar framework of classification. The result of the distilled RoBERTa model can also elucidate an improvement in the classification of clinical codes, that can benefit from interpretability techniques. In contrast to [95], this model can also benefit from explainability, but using the SHAP explainer as well as with the interpretation of domain experts and software developers to interpret the contributions of the input tokens of the model.

Chapter 6

Conclusion

This work reflects a study on the "Attention-based clinical note summarization" [1], an interpretability analysis on the matter as well as proposal for the improvement of such studies. The attention mechanism inside transformer-based architectures was investigated in parallel to the automatic text summarization task. An interpretability analysis was performed on the *sumly* algorithm, using the SHAP explainer as well as a comparison using newer BERT-like architectures like the RoBERTA model. This comparison showed that the original algorithm can benefit from more recent and optimized transformer-based models. The interpretation analysis suggests that the procedure can definitely benefit from domain experts' intervention in the developing process as a human-in-the-loop approach. In addition to these works, it was also possible to show that the procedure is flexible, adaptable, and applicable in different contexts besides the medical one for the task of summarization. Further investigation can be done on the classification task using the attention scores to help improve these works, as well as a deeper analysis on how to better create plausible and faithful explanations for the summarization task in a medical environment.

6.1 Future Work

Some further work is proposed for the improvement of the studies presented here. The implementation of GAT attention networks for the summarization task can be beneficial not only for the generation of the final summary but also for interpretability's sake. A similar approach as the one shown in [16] can be adapted to this task. The challenge here is to overcome the problem of not having a reference label for the original clinical notes. In this way, it becomes possible to train a GAT that can be also infused with domain knowledge, for instance, including the description of the ICD-9 codes present in the medical note. This could also repress the limitation of the size of the input for long medical notes, making it possible

to pass the entire text and let the GAT recreate a representation of the text in a graph-like form, understanding the relationship between sentences. Following the same lines the use of Longformers [42] or more efficient (or bigger) transformers that can handle the size limit constraint could be useful. An approach as the one shown in [80] could also be applied once the challenge of the reference labels is solved. This work could provide a different kind of faithful explanation as well as a more efficient approach, based on the original model. One proposal to overcome the evident lack of reference label problem is to devise a human-annotated dataset consisting of summaries performed by domain experts. This is not only beneficial for further creating supervised explainable models, but also it can help in providing plausible explanations for models. Finally, the use of a more recent dataset can provide a different overview of a more updated medical field. The last version of the MIMIC dataset [97], accompanied by ICD-10 code classification might provide different insights into newer or updated medical terminologies. The challenge here is to adapt the pre-processing step for this change in the dataset.

Bibliography

[1] Kanwal et al. «Attention-based clinical note summarization». In: (2022) (cit. on pp. 1, 3, 18, 22, 25, 29, 36).

[2] El-Kassas Vaswani et al. «Automatic text summarization: A comprehensive survey». In: (2020) (cit. on pp. 1–3).

[3] Amol Tandel, Brijesh Modi, Priyasha Gupta, Shreya Wagle, and Sujata Khedkar. «Multi-document text summarization - a survey». In: *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. 2016, pp. 331–334. DOI: 10.1109/SAPIENCE.2016.7684115 (cit. on p. 2).

[4] Liwei Hou, Po Hu, and Chao Bei. «Abstractive Document Summarization via Neural Model with Joint Attention». In: *Natural Language Processing and Chinese Computing*. Ed. by Xuanjing Huang, Jing Jiang, Dongyan Zhao, Yansong Feng, and Yu Hong. Cham: Springer International Publishing, 2018, pp. 329–338. ISBN: 978-3-319-73618-1 (cit. on p. 2).

[5] Rui Sun, Zhenchao Wang, Yafeng Ren, and Donghong Ji. «Query-Biased Multi-document Abstractive Summarization via Submodular Maximization Using Event Guidance». In: *Web-Age Information Management*. Ed. by Bin Cui, Nan Zhang, Jianliang Xu, Xiang Lian, and Dexi Liu. Cham: Springer International Publishing, 2016, pp. 310–322. ISBN: 978-3-319-39937-9 (cit. on p. 2).

[6] Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. «Integrating Extractive and Abstractive Models for Long Text Summarization». In: *2017 IEEE International Congress on Big Data (BigData Congress)*. 2017, pp. 305–312. DOI: 10.1109/BigDataCongress.2017.46 (cit. on p. 2).

[7] Malak Abdullah, Alia Madain, and Yaser Jararweh. «ChatGPT: Fundamentals, Applications and Social Impacts». In: *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. 2022, pp. 1–8. DOI: 10.1109/SNAMS58071.2022.10062688 (cit. on p. 2).

[8] Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. «Self-Attention Guided Copy Mechanism for Abstractive Summarization». In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1355–1362. DOI: 10.18653/v1/2020.acl-main.125. URL: https://aclanthology.org/2020.acl-main.125 (cit. on p. 2).

[9] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. «Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond». In: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. DOI: 10.18653/v1/K16-1028. URL: https://aclanthology.org/K16-1028 (cit. on pp. 2, 4).

[10] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. «English gigaword». In: *Linguistic Data Consortium, Philadelphia* 4.1 (2003), p. 34 (cit. on pp. 2, 4).

[11] Vishal Gupta and Gurpreet Lehal. «A Survey of Text Summarization Extractive Techniques». In: *Journal of Emerging Technologies in Web Intelligence* 2 (Aug. 2010). DOI: 10.4304/jetwi.2.3.258-268 (cit. on p. 3).

[12] Kathleen McKeown, Rebecca Passonneau, David Elson, Ani Nenkova, and Julia Hirschberg. «Do Summaries Help? A Task-Based Evaluation of Multi-Document Summarization». In: *Proceedings of SIGIR* (Jan. 2005) (cit. on p. 3).

[13] Elena Lloret, Laura Plaza, and Ahmet Aker. «The challenging task of summary evaluation: an overview». In: *Language Resources and Evaluation* 52.1 (Sept. 2017), pp. 101–148. DOI: 10.1007/s10579-017-9399-2. URL: https://doi.org/10.1007/s10579-017-9399-2 (cit. on p. 3).

[14] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. «Collective classification in network data». In: *AI magazine* 29.3 (2008), pp. 93–93 (cit. on p. 4).

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. «Inductive representation learning on large graphs». In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 4).

[16] Jia et al. «Neural Extractive Summarization with Hierarchical Attentive Heterogeneous Graph Network». In: (2020) (cit. on pp. 4, 5, 36).

[17] Veličković et al. «Graph Attention Networks». In: (2017) (cit. on p. 4).

[18] Xingxing Zhang, Yiran Liu, Xun Wang, Pengcheng He, Yang Yu, Si-Qing Chen, Wayne Xiong, and Furu Wei. «Momentum Calibration for Text Generation». In: (Dec. 2022). DOI: 10.48550/arXiv.2212.04257 (cit. on p. 4).

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «Bert: Pre-training of deep bidirectional transformers for language understanding». In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on pp. 5, 6).

[20] Ashish Vaswani et. al. «Attention is all you need». In: (2017) (cit. on pp. 5, 9–11).

[21] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. «Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension». In: *arXiv preprint arXiv:1910.13461* (2019) (cit. on pp. 5, 6).

[22] Yixin Liu and Pengfei 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)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1065–1072. DOI: 10.18653/v1/2021.acl-short.135. URL: https://aclanthology.org/2021.acl-short.135 (cit. on p. 6).

[23] Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. «R-drop: Regularized dropout for neural networks». In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 10890–10905 (cit. on p. 6).

[24] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. «Pegasus: Pre-training with extracted gap-sentences for abstractive summarization». In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11328–11339 (cit. on pp. 6, 7).

[25] Mathieu Ravaut, Shafiq Joty, and Nancy Chen. «SummaReranker: A Multi-Task Mixture-of-Experts Re-ranking Framework for Abstractive Summarization». In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4504–4524. DOI: 10.18653/v1/2022.acl-long.309. URL: https://aclanthology.org/2022.acl-long.309 (cit. on p. 6).

[26] Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter Liu. «Calibrating Sequence likelihood Improves Conditional Language Generation». In: (Sept. 2022) (cit. on p. 6).

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer». In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html (cit. on pp. 6, 7).

[28] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. «Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing». In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815 (cit. on p. 6).

[29] Zachary C. Lipton. «The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery.» In: *Queue* 16.3 (June 2018), pp. 31–57. ISSN: 1542-7730. DOI: 10.1145/3236386.3241340. URL: https://doi.org/10.1145/3236386.3241340 (cit. on pp. 8, 14).

[30] Been Kim. «Interactive and interpretable machine learning models for human machine collaboration». PhD thesis. Massachusetts Institute of Technology, 2015 (cit. on p. 8).

[31] Tim Miller. «Explanation in artificial intelligence: Insights from the social sciences». In: *Artificial Intelligence* 267 (2019), pp. 1–38. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2018.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0004370218305988 (cit. on p. 8).

[32] El Zini et al. «On the Evaluation of the Plausibility and Faithfulness of Sentiment Analysis Explanations». In: (2022) (cit. on p. 9).

[33] Cynthia Rudin. «Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead». In: *Nature machine intelligence* 1.5 (2019), pp. 206–215 (cit. on pp. 9, 12).

[34] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. «Structured attention networks». In: *arXiv preprint arXiv:1702.00887* (2017) (cit. on p. 9).

[35] Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. «An attentive survey of attention models». In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.5 (2021), pp. 1–32 (cit. on pp. 9–11).

[36] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. «Neural machine translation by jointly learning to align and translate». In: *arXiv preprint arXiv:1409.0473* (2014) (cit. on p. 10).

[37] Thang Luong, Hieu Pham, and Christopher D. Manning. «Effective Approaches to Attention-based Neural Machine Translation». In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1412–1421. DOI: 10.18653/v1/D15-1166. URL: https://aclanthology.org/D15-1166 (cit. on p. 10).

[38] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. «Coupled multi-layer attentions for co-extraction of aspect and opinion terms». In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017 (cit. on p. 10).

[39] Douwe Kiela, Changhan Wang, and Kyunghyun Cho. «Dynamic meta-embeddings for improved sentence representations». In: *arXiv preprint arXiv:1804.07983* (2018) (cit. on p. 10).

[40] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL] (cit. on pp. 10, 24).

[41] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. «Improving language understanding by generative pre-training». In: (2018) (cit. on p. 10).

[42] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. *Efficient Transformers: A Survey*. 2022. arXiv: 2009.06732 [cs.LG] (cit. on pp. 11, 37).

[43] Samira Abnar and Willem Zuidema. «Quantifying Attention Flow in Transformers». In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4190–4197. DOI: 10.18653/v1/2020.acl-main.385. URL: https://aclanthology.org/2020.acl-main.385 (cit. on pp. 11, 18).

[44] Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. «Revealing the Dark Secrets of BERT». In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4365–4374. DOI: 10.18653/v1/D19-1445. URL: https://aclanthology.org/D19-1445 (cit. on pp. 11, 18).

[45] Vinit Ravishankar, Artur Kulmizev, Mostafa Abdou, Anders Søgaard, and Joakim Nivre. «Attention Can Reflect Syntactic Structure (If You Let It)». In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 3031–3045. DOI: 10.18653/v1/2021.eacl-main.264. URL: https://aclanthology.org/2021.eacl-main.264 (cit. on p. 11).

[46] Jain et al. «Attention is not Explanation». In: (2019) (cit. on p. 11).

[47] Rudin et al. «Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead». In: (2019) (cit. on p. 11).

[48] Wiegreffe et al. «Attention is not not Explanation». In: (2019) (cit. on p. 11).

[49] Martin Tutek and Jan Snajder. «Staying True to Your Word: (How) Can Attention Become Explanation?» In: *Proceedings of the 5th Workshop on Representation Learning for NLP*. Online: Association for Computational Linguistics, July 2020, pp. 131–142. DOI: 10.18653/v1/2020.repl4nlp-1.17. URL: https://aclanthology.org/2020.repl4nlp-1.17 (cit. on p. 11).

[50] Jasmijn Bastings and Katja Filippova. «The elephant in the interpretability room: Why use attention as explanation when we have saliency methods?» In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, Nov. 2020, pp. 149–155. DOI: 10.18653/v1/2020.blackboxnlp-1.14. URL: https://aclanthology.org/2020.blackboxnlp-1.14 (cit. on p. 12).

[51] Misha Denil, Alban Demiraj, and Nando de Freitas. «Extraction of Salient Sentences from Labelled Documents». In: *CoRR* abs/1412.6815 (2014). arXiv: 1412.6815. URL: http://arxiv.org/abs/1412.6815 (cit. on p. 12).

[52] Shuoyang Ding, Hainan Xu, and Philipp Koehn. «Saliency-driven Word Alignment Interpretation for Neural Machine Translation». In: *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 1–12. DOI: 10.18653/v1/W19-5201. URL: https://aclanthology.org/W19-5201 (cit. on p. 12).

[53] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. «The (un) reliability of saliency methods». In: *Explainable AI: Interpreting, explaining and visualizing deep learning* (2019), pp. 267–280 (cit. on p. 12).

[54] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. «Gradient-Based Attribution Methods». In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Ed. by Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. Cham: Springer International Publishing, 2019, pp. 169–191. ISBN: 978-3-030-28954-6. DOI: 10.1007/978-3-030-28954-6_9. URL: https://doi.org/10.1007/978-3-030-28954-6_9 (cit. on p. 12).

[55] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. «On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation». In: *PloS one* 10.7 (2015), e0130140 (cit. on p. 13).

[56] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. «Visualizing and Understanding Neural Machine Translation». In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1150–1159. DOI: 10.18653/v1/P17-1106. URL: https://aclanthology.org/P17-1106 (cit. on p. 13).

[57] Carlos Córdova Sáenz and Karin Becker. «Assessing the use of attention weights to interpret BERT-based stance classification». In: Dec. 2021, pp. 194–201. DOI: 10.1145/3486622.3493966 (cit. on pp. 13, 14).

[58] Sofia Serrano and Noah A. Smith. «Is Attention Interpretable?» In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 2931–2951. DOI: 10.18653/v1/P19-1282. URL: https://aclanthology.org/P19-1282 (cit. on p. 13).

[59] Shengzhong Liu, Franck Le, Supriyo Chakraborty, and Tarek Abdelzaher. «On Exploring Attention-based Explanation for Transformer Models in Text Classification». In: Dec. 2021, pp. 1193–1203. DOI: 10.1109/BigData52589.2021.9671639 (cit. on p. 13).

[60] Shishir Rao, Yikuan Li, Rema Ramakrishnan, Abdelaali Hassaine, Dexter Canoy, John Cleland, Thomas Lukasiewicz, Gholamreza Salimi-Khorshidi, and Kazem Rahimi. «An explainable Transformer-based deep learning model for the prediction of incident heart failure». In: *IEEE Journal of Biomedical and Health Informatics* 26.7 (2022), pp. 3362–3372 (cit. on p. 13).

[61] Shunsuke Kitada and Hitoshi Iyatomi. «Attention meets perturbations: Robust and interpretable attention with adversarial training». In: *IEEE Access* 9 (2021), pp. 92974–92985 (cit. on p. 13).

[62] J Alammar. «Ecco: An open source library for the explainability of transformer language models». In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*. 2021, pp. 249–257 (cit. on pp. 13, 16).

[63] Scott M Lundberg and Su-In Lee. «A unified approach to interpreting model predictions». In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 13).

[64] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. «" Why should i trust you?" Explaining the predictions of any classifier». In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144 (cit. on pp. 13, 14, 16, 20).

[65] Yonghui Wu et al. «Google's neural machine translation system: Bridging the gap between human and machine translation». In: *arXiv preprint arXiv:1609.08144* (2016) (cit. on p. 14).

[66] Jacovi et al. «Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?» In: (2020) (cit. on p. 14).

[67] Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. *An Evaluation of the Human-Interpretability of Explanation*. 2019. arXiv: 1902.00006 [cs.LG] (cit. on p. 14).

[68] Bernease Herman. *The Promise and Peril of Human Evaluation for Model Interpretability*. 2019. arXiv: 1711.07414 [cs.AI] (cit. on p. 14).

[69] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. *Explaining Explanations: An Overview of Interpretability of Machine Learning*. 2019. arXiv: 1806.00069 [cs.AI] (cit. on p. 14).

[70] Jialin Wu and Raymond J. Mooney. *Faithful Multimodal Explanation for Visual Question Answering*. 2019. arXiv: 1809.02805 [cs.CL] (cit. on p. 14).

[71] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. «Faithful and Customizable Explanations of Black Box Models». In: *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '19. Honolulu, HI, USA: Association for Computing Machinery, 2019, pp. 131–138. ISBN: 9781450363242. DOI: 10.1145/3306618.3314229. URL: https://doi.org/10.1145/3306618.3314229 (cit. on p. 14).

[72] Supriyo Chakraborty et al. «Interpretability of deep learning models: A survey of results». In: *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 2017, pp. 1–6. DOI: 10.1109/UIC-ATC.2017.8397411 (cit. on p. 14).

[73] Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. *Can I Trust the Explainer? Verifying Post-hoc Explanatory Methods*. 2019. arXiv: 1910.02065 [cs.CL] (cit. on p. 14).

[74] Ivan Sanchez, Tim Rocktaschel, Sebastian Riedel, and Sameer Singh. «Towards extracting faithful and descriptive representations of latent variable models». In: *AAAI Spring Syposium on Knowledge Representation and Reasoning (KRR): Integrating Symbolic and Neural Approaches* 1 (2015), pp. 4–1 (cit. on p. 14).

[75] Przemyslaw Biecek. «DALEX: Explainers for Complex Predictive Models in R». In: *Journal of Machine Learning Research* 19.84 (2018), pp. 1–5. URL: http://jmlr.org/papers/v19/18-416.html (cit. on p. 14).

[76] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. «Interpretable machine learning: definitions, methods, and applications». In: *arXiv preprint arXiv:1901.04592* (2019) (cit. on p. 14).

[77] Joris Baan, Maartje ter Hoeve, Marlies van der Wees, Anne Schuth, and Maarten de Rijke. *Do Transformer Attention Heads Provide Transparency in Abstractive Summarization?* 2019. arXiv: 1907.00570 [cs.CL] (cit. on p. 14).

[78] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. «A Survey of Methods for Explaining Black Box Models». In: *ACM Comput. Surv.* 51.5 (Aug. 2018). ISSN: 0360-0300. DOI: 10.1145/3236009. URL: https://doi.org/10.1145/3236009 (cit. on p. 14).

[79] David Alvarez-Melis and Tommi S. Jaakkola. *On the Robustness of Interpretability Methods*. 2018. arXiv: 1806.08049 [cs.LG] (cit. on p. 14).

[80] Wood-Doughty et al. «Faithful and Plausible Explanations of Medical Code Predictions». In: (2021) (cit. on pp. 14, 17, 20, 37).

[81] Treviso et al. «The Explanation Game: Towards Prediction Explainability through Sparse Communication». In: (2020) (cit. on pp. 15, 19).

[82] Lipton et al. «The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery». In: (2018) (cit. on p. 15).

[83] Laurens van der Maaten and Geoffrey Hinton. «Viualizing data using t-SNE». In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605 (cit. on p. 16).

[84] Lundberg et al. «A Unified Approach to Interpreting Model Predictions». In: (2017) (cit. on p. 16).

[85] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. «Anchors: High-precision model-agnostic explanations». In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018 (cit. on p. 16).

[86] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. «HotFlip: White-Box Adversarial Examples for Text Classification». In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 31–36. DOI: 10.18653/v1/P18-2006. URL: https://aclanthology.org/P18-2006 (cit. on p. 16).

[87] Jesse Vig. «A Multiscale Visualization of Attention in the Transformer Model». In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 37–42. DOI: 10.18653/v1/P19-3007. URL: https://www.aclweb.org/anthology/P19-3007 (cit. on p. 16).

[88] Raymond Li, Wen Xiao, Lanjun Wang, Hyeju Jang, and Giuseppe Carenini. «T3-Vis: visual analytic for Training and fine-Tuning Transformers in NLP». In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 220–230. DOI: 10.18653/v1/2021.emnlp-demo.26. URL: https://aclanthology.org/2021.emnlp-demo.26 (cit. on pp. 16, 17).

[89] Zhen Li, Xiting Wang, Weikai Yang, Jing Wu, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Hui Zhang, and Shixia Liu. «A Unified Understanding of Deep NLP Models for Text Classification». In: June 2022. DOI: 10.48550/arXiv.2206.09355 (cit. on p. 16).

[90] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. «Human-in-the-loop machine learning: a state of the art». In: *Artificial Intelligence Review* 56 (Aug. 2022). DOI: 10.1007/s10462-022-10246-w (cit. on p. 17).

[91] Xie et al. «Pre-trained language models with domain knowledge for biomedical extractive summarization». In: (2022) (cit. on pp. 17, 19, 20).

[92] Raymond Li, Wen Xiao, Linzi Xing, Lanjun Wang, Gabriel Murray, and Giuseppe Carenini. *Human Guided Exploitation of Interpretable Attention Patterns in Summarization and Topic Segmentation*. 2022. arXiv: 2112.05364 [cs.CL] (cit. on p. 17).

[93] Johnson et al. «MIMIC-III Clinical Database (version 1.4). PhysioNet.» In: (2016) (cit. on pp. 18, 25).

[94] Yiyun Zhao and Steven Bethard. «How does BERT's attention change when you fine-tune? An analysis methodology and a case study in negation scope». In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics,

July 2020, pp. 4729–4747. DOI: 10.18653/v1/2020.acl-main.429. URL: https://aclanthology.org/2020.acl-main.429 (cit. on p. 18).

[95] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. «Explainable Prediction of Medical Codes from Clinical Text». In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1101–1111. DOI: 10.18653/v1/N18-1100. URL: https://aclanthology.org/N18-1100 (cit. on pp. 20, 24, 25, 31, 35).

[96] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. «DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter». In: *ArXiv* abs/1910.01108 (2019) (cit. on p. 24).

[97] Johnson et al. «MIMIC-IV (version 2.2). PhysioNet.» In: (2023) (cit. on p. 37).