SPEC5G: A Dataset for 5G Cellular Network Protocol Analysis

Imtiaz Karim, Kazi Samin Mubasshir, Mirza Masfiqur Rahman, Elisa Bertino
Purdue University
{karim7, kmubassh, rahman75, bertino}@purdue.edu

Abstract

5G is the 5^{th} generation state-of-the-art cellular network protocol designed to connect virtually everyone and everything with increased speed and reduced latency. Therefore, its development, analysis, and security are critical. However, all approaches to the 5G protocol development and security analysis, e.g., property extraction, protocol summarization, and semantic analysis of the protocol specifications and implementations are completely manual. To reduce such manual efforts, in this paper, we curate SPEC5G–the *first-ever* public 5G dataset for NLP research. The dataset contains 3,547,587 sentences with 134M words, from 13094 cellular network specifications and 13 online websites. By leveraging large-scale pre-trained language models that have achieved state-of-the-art results on NLP tasks, we use this dataset for security-related text classification and summarization. Security-related text classification can be used to extract relevant security-related properties for protocol testing. On the other hand, summarization can help developers and practitioners understand the high-level idea of the protocol, which is itself a daunting task. To ensure the research community can benefit from this work, all the datasets and accompanying codebase are made publicly available[1].

1 Introduction

The deployment of the 5G cellular network protocol has generated a lot of enthusiasm in academia and industry, because of its promise of enabling innovative applications, such as autonomous vehicles (Ahmad et al., 2020), remote surgery (sur, 2022), industrial IoT (Satyanarayanan, 2017), augmented reality (Zhang et al., 2017), and multiplayer online gaming (scr, 2022). Therefore the security of 5G protocol is critical. Unfortunately, the 5G protocol development and analysis are all completely manual tasks requiring domain expertise. We observe that for 5G there is an unutilized resource of information available in the form of specifications (Spe, 2022) and numerous tutorials on the Internet. These resources have not yet been utilized.

Recently, a few approaches have been proposed that leverage natural language processing (NLP) and machine learning (ML) to detect risky operations in some of the specifications of 4G LTE (Chen et al., 2021) and to analyze change requests (Chen et al., 2022). These approaches are very limited, not generalizable, and not open-source. Automatic and systematic analysis of 5G networks is still a difficult task. One major problem is the lack of high-quality datasets to train ML models, which would enable the automation of different 5G-related downstream tasks e.g., security-related text classification, protocol summarization, semantic analysis, and automatic programming. In this paper, we address this need by introducing SPEC5G, a high-quality dataset of the 5G protocol specifications. 5G is not a single wireless technology, but an umbrella term used to categorize the fifth generation of wireless communication, including hundreds of different protocols at different layers of the protocol. Some of these protocols are VoWiFi, cellular IoT, IKE, and 5G-AKA. SPEC5G is a complete dataset that covers all these protocols and therefore, has the potential to impact different protocols affecting billions of devices. Such a high-quality dataset would be beneficial to numerous applications in different domains, such as security testing, policy enforcement, automatic code generation, and protocol summarization. It would encourage research and development in novel NLP tasks that are communication protocol-specific and critical for the security analysis of these protocols. Notable examples include formal model extraction from large-scale natural language documents and identifications of conflicting security guidelines.

[1]Datasets and codebase for SPEC5G are publicly available at https://github.com/Imtiazkarimik23/SPEC5G

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 20–38

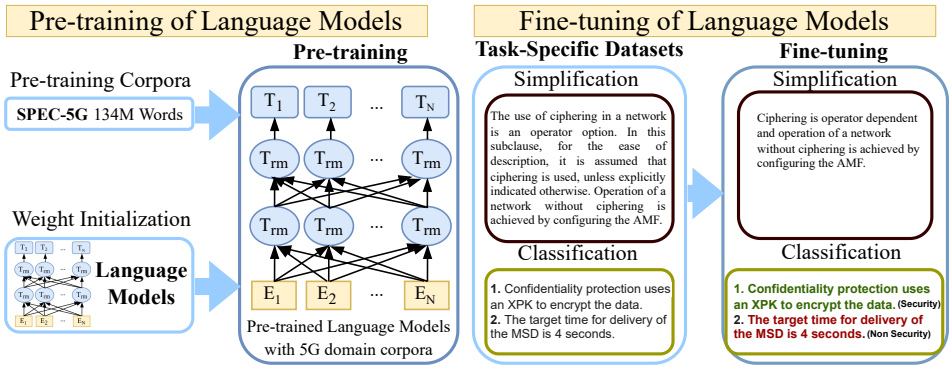

Figure 1: Overview of our pre-training and fine-tuning on downstream tasks using SPEC5G

To show the viability of our SPEC5G dataset, we use it for two downstream tasks (shown in Figure 1). First, we use it for *security-text classification*. In previous 5G security testing (Basin et al., 2018; Cremers and Dehnel-Wild, 2019; Hussain et al., 2019) the properties are manually extracted from the specifications. Using security-text classification, we can automatically identify texts, which specify important security properties to be used for formal verification and other testing approaches. Second, we use SPEC5G for the *paragraph summarization* task. The 5G specification is large and complex with specialized jargon, mostly due to backward compatibility requirements. Therefore, it is really daunting for a software developer to understand the high-level ideas of the protocol specification. With the summarization task, we show that it is possible to summarize and simplify the high-level ideas of the protocol. To achieve those tasks, we created two expert-annotated datasets: one for summarization and one for classification. The summarization dataset contains 713 long articles and their concise summaries. The classification dataset contains 2401 sentences and their class labels (*Non-Security*, *Security*, *Undefined*). Both datasets were annotated by multiple domain experts to ensure quality and fairness. Along with SPEC5G, these two expert annotated datasets have been open-sourced to enhance research.

On the whole, our contributions are threefold. First, we create the first-ever novel 5G dataset (SPEC5G) of 3,547,587 sentences by preprocessing the 5G specification and scraping data from different 5G tutorials on the Internet. Second, we create two expert-annotated datasets for baseline security-text classification and summarization tasks. We conduct an extensive evaluation of these datasets using several NLP models on the downstream tasks. The results show that the models pre-trained on SPEC5G outperform all baseline models. Third, all these research artifacts have been made available via a public repository. To the best of our knowledge, this is the *first-ever* public 5G dataset created for NLP research.

2 Related Work

The introduction of the attention-based transformer architecture by (Vaswani et al., 2017) beaconed the era of transformer-based Language Models (LM) in the field of NLP. A range of high-performing transformer-based language models have since been proposed, each with its own specific use cases. To train such LMs, high-quality large datasets are critical. In the following, we will discuss the research relevant to our work.

Cellular Networks Research Using NLP. CREEK (Chen et al., 2022) uses BERT models for detecting security-relevant change requests. For this, they pre-train BERT with a subset of 4G LTE specifications (1546 out of 13094). Moreover, in ATOMIC (Chen et al., 2021) they design a framework to semantically analyze LTE documents using NLP to obtain a set of hazard indicators for generating test cases based on a given threat model. These are the first steps in applying NLP techniques to analyze cellular network specifications. In a technical blog post from Erricson (err, 2022), the authors adopt LMs for the telecom domain and create a telecom question-answering dataset. Though promising, these approaches do not generalize and are ad-hoc and closed-source, thus accentuating the need for a complete and public dataset for 5G.

Summarization. Following BookCorpus and Wikidata, researchers have built summarization datasets such as Wikilarge (Zhang and Lapata, 2017), Wikismall (Zhu et al., 2010), and so

on (Coster and Kauchak, 2011; Kauchak, 2013). Such datasets are widely used in the field of sentence summarization. Early summarization models mostly relied on statistical machine translation (Wubben et al., 2012; Narayan and Gardent, 2014). Improvements of the machine translation model to obtain a new summarization model are done by (Nisioi et al., 2017) and investigations on how to simplify sentences to different difficulty levels are conducted after this (Scarton and Specia, 2018; Nishihara et al., 2019). Sentence alignment methods to improve sentence summarization are proposed by (Štajner et al., 2017) and (Jiang et al., 2020). There are several corpora related to summarization. A large-scale, human-annotated scientific papers corpus is provided by (Yasunaga et al., 2019). This corpus provides over 1,000 papers in the ACL anthology with their citation networks (e.g., citation sentences, citation counts) and their comprehensive, manual summaries. There is another dataset that has been created for the Computational Linguistics Scientific Document Summarization Shared Task which started in 2014 as a pilot (Jaidka et al., 2014) and which is now a well-developed challenge in its fourth year (Jaidka et al., 2018, 2017). A new dataset for summarisation of computer science publications by exploiting a large resource of the author-provided summaries is introduced by (Collins et al., 2017).

Sentence Classification. The Corpus of Linguistic Acceptability (CoLA) (Warstadt et al., 2018) consists of English acceptability judgments drawn from books and journal articles on linguistic theory. Each example is a sequence of words annotated with whether it is an English grammatical sentence. The Stanford Sentiment Treebank (Socher et al., 2013) consists of sentences from movie reviews and human annotations of their sentiment. SciCite (Cohan et al., 2019a) is a large dataset of citation intents for the task of automated analysis of scientific papers by identifying the intent of a citation (e.g., background information, use of methods, comparing results). Researchers have also leveraged other large datasets such as DEFT (Spala et al., 2019) and ACL-ARC (Bird et al., 2008) for the sentence classification tasks. CSABSTRUCT (Cohan et al., 2019b) is another new dataset of manually annotated sentences from computer science abstracts for Sequential Sentence Classification (SSC). Paper Field (Sinha et al., 2015) is built from the Microsoft Academic Graph and maps paper titles to one of 7 fields of study: geography, politics, economics, business, sociology, medicine, and psychology. DBpedia is aimed at extracting structured content from Wikipedia. This is a data extract (after preprocessing, with kernel included) with taxonomic, hierarchical categories, or classes, for 343k Wikipedia articles. A version of this dataset is also a popular baseline for text classification tasks.

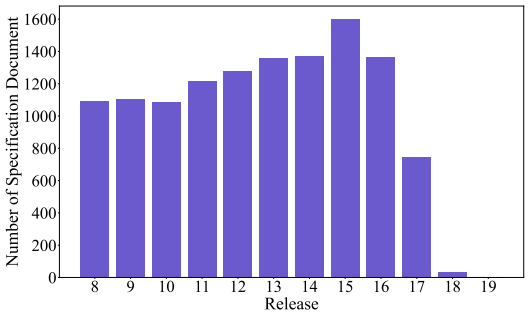

Figure 2: Number of Technical Specifications in subsequent releases from 3GPP. Release 15 has seen the largest number of documents. The current release (Rel-19) has only 1 specification document so far. Mean: 1021.08, median: 1160, min: 1, max: 1601, sd: 491.50, skewness: -1.19, kurtosis: 0.13

3 Dataset Curation

In this section, we discuss the collection and preparation of our dataset. A significant amount of data was collected from the 3GPP website (Spe, 2022).

3.1 3GPP

The 3rd Generation Partnership Project (3GPP) is an umbrella organization that hosts several organizations from different countries. 3GPP is globally considered the issuer of standards for cellular network protocols. These standards are publicized as releases, e.g., LTE standards were made public from Release 8 and 5G standards from Release 15. The current release is Release 19.

A large number of meeting minutes, Technical Reports (TR) can be found on the 3GPP FTP server (Spe, 2022). 3GPP releases a set of Technical Specifications (TS) as well, which subsequently add features and bug fixes. Figure 2 shows the count of specification documents per release.

3.2 Dataset Collection

As stated earlier a significant portion of the dataset has been collected from the 3GPP FTP server. Automated NLP tasks have been hindered in the 5G domain because of noisy data in the standard doc-

umentation. Often, the specification documents contain embedded codes, tables, and lists with definitions of varying terminologies, flow diagrams, finite state machines, and so on–which makes it hard to build models that reason and perform well on downstream applications.

Thus, to leverage downstream NLP tasks, we perform extensive preprocessing. Furthermore, we scrape data from 13 blogs, and forums of the internet. The web sources are listed in Table 4 and details about the web sources can be found in Appendix C.5. We extract approximately 17 GB of text data from specification releases and web portals using python web scrapper and Selenium (sel, 2022). Later we apply a set of standard and domain-specific preprocessing to obtain the final dataset.

3.2.1 Preprocessing

5G specifications and web data contain a variety of materials encompassing method and framework documentation, pseudocode, high-level implementations, numerous parameters, field constitution, and so on. At first, the raw data go through standard NLP preprocessing tasks, e.g., removing extra whitespaces, tabs, certain Unicode characters introduced from scrapping, HTML tags, etc. Later, we extend the preprocessing to handle special cases such as code snippets, tables, figures, references to other specification documents, etc. For the list of preprocessing tasks, we refer the reader to Appendix A.1. Finally, this dataset is used to pre-train baseline models for downstream applications.

3.2.2 Dataset Statistics

Our final processed dataset contains 3,547,587 sentences with a total of 134M words. Figure 3 shows the distribution of the number of sentences per document and Figure 4 shows the distribution of tokens per sentence.

3.3 Annotation

To demonstrate the effectiveness of SPEC5G, we additionally create and annotate two datasets specific to two NLP tasks - summarization and sentence classification.

3.3.1 Summarization

To prepare the summarization dataset, we randomly select 1500 locations to retrieve articles from the SPEC5G dataset. An article is defined as a sequential collection of sentences. Here we apply another round of manual processing to ensure semantic correctness among the sentences of each of

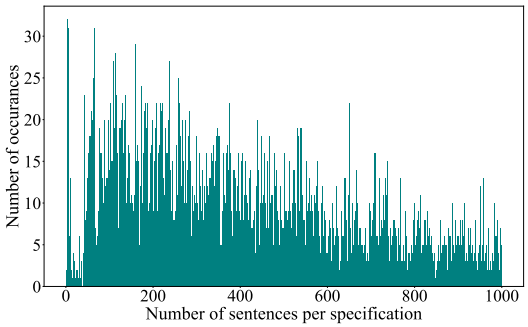

Figure 3: Document distribution based on sentences. Documents with more than 1000 sentences were omitted from the figure for better visualization. Mean: 400.90, median: 356, min: 0, max: 1000, sd: 257.83, skewness: 0.52, kurtosis: -0.73

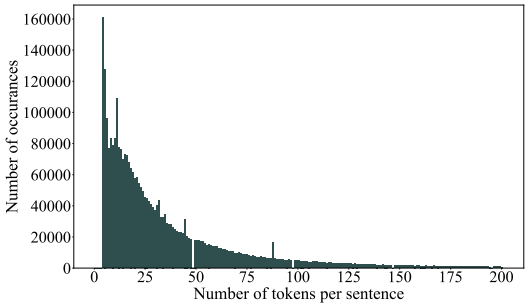

Figure 4: Sentence distribution based on tokens. Sentences with more than 200 tokens were omitted from the figure for better visualization. Mean: 34.89, median: 23, min: 4, max: 200, sd: 34.91, skewness: 1.95, kurtosis: 4.10

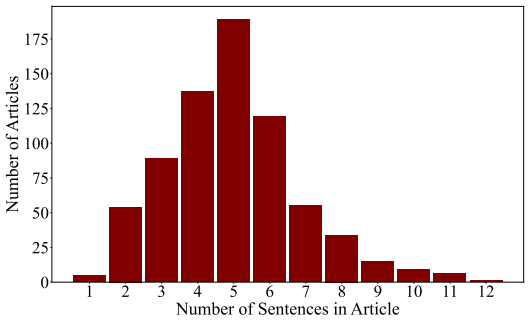

Figure 5: Sentence distribution per article. Mean: 4.97, median: 5, min: 1, max: 12, sd: 1.85, skewness: 0.61, kurtosis: 0.72

the articles. The final curated dataset contains 713 articles, each with 1-12 sentences. The distribution of sentences per article is shown in Figure 5. This dataset is subsequently labeled by 9 domain experts; each label itself is a smaller set of sentences that summarizes the article. The task of annotation (summarizing) varies in difficulty. The annotators have made insightful comments about the articles

they have faced challenges with. Another round of manual data cleaning has been done based on the comments which resulted in a very high-quality test set for protocol specification summarization. For the rest of this paper, we refer to this annotated dataset as 5GSum.

3.3.2 Security Classification

Similar to the summarization task, we randomly select and annotate 2401 sentences from our SPEC5G dataset to use for multi-class classification. We categorize the data into 3 classes- *Non-Security* (0), *Security* (1), and *Undefined* (2). To discard human bias, the dataset has been labeled by 9 domain experts. We do a 85-5-10 split for train, validation, and test data with 2040, 120, and 241 samples respectively. For the rest of the paper, we refer to this dataset as 5GSC.

Among the 3 classes, the least number of samples are from class 2 (*Undefined*: 484). Yet, the class with the highest number of samples (*Non-Security*: 1303) is about 3 times more than the class with the lowest number of samples. Therefore, the dataset is not highly imbalanced. Overall, this non-uniformity is expected, since most of the specification documents should not be related to *Security* issues and a high amount of *Undefined* statements in 5G specifications would rather mean inconsistencies in implementation.

4 Tasks

In this section, we define the downstream tasks: summarization and security sentence classification. Moreover, we discuss the relevance of these downstream tasks with respect to 5G.

4.1 Task 1: Summarization

Text summarization is the simplification of the original text to a more understandable text while keeping the main meaning of the original text unchanged (Štajner and Saggion, 2018; Maddela et al., 2020). It can provide convenience for non-native speakers (Petersen and Ostendorf, 2007; Glavaš and Štajner, 2015; Paetzold and Specia, 2016), non-expert readers (Elhadad and Sutaria, 2007; Siddharthan and Katsos, 2010). In the case of 5G standard documents, summarization can help developers and practitioners understand the high-level idea of the protocol, which can be really time-consuming without the summarization.

The document-level text summarization task can be defined as follows. Let C be an original complex article; suppose that C consists of n sentences, denoted as $C = S_1, S_2, \ldots, S_n$. Document-level summarization aims to simplify C into m sentences, which form the simplified article F, denoted as $F = T_1, T_2, \ldots, T_m$, where m is not necessarily equal to n. F retains the primary meaning of C and is more straightforward than C, making it easier for people to understand. The operations for sentence-level summarization include word reservation and deletion, synonym replacement (Xu et al., 2016). In our definition, document-level summarization should allow the loss of information but should not allow the loss of important information. The fact that sentence deletion is a prevalent phenomenon in document summarization is pointed out by (Zhong et al., 2019). We believe that information that has little relevance to the primary meaning should be removed to improve readability.

The objective is to simplify a paragraph without losing important information. Task 1 is more challenging when evaluating a model's ability to reason about unobserved effects.

4.2 Task 2: Sentence Classification

Text classification is a classic topic for natural language processing, in which one needs to assign predefined categories to free-text documents. The range of text classification research goes from designing the best features to choosing the best possible machine learning classifiers (Mekala et al., 2021; Liu et al., 2021; Zhang et al., 2022).

The multi-class sentence classification can be defined as follows. Given a sentence $s \in \mathcal{S}$, where $\mathcal{S}$ is some high dimensional sentence space and a finite set of categories or classes $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, the objective of multi-class sentence classification is to find a function $\mathcal{F}$ mapping sentences to categories, formally, $\mathcal{F} : \mathcal{S} \rightarrow \mathcal{C}$. Given a dataset $\tilde{\mathcal{D}}$ of m training samples $\{(s_i, c_i)\}_{i=1}^{m}$, we aim to learn the function $\bar{\mathcal{F}}$ that approximates $\mathcal{F}$.

For protocol analysis, an important step is property-guided testing (Hussain et al., 2019). Up to this point, the properties are manually extracted, and the testing is entirely manual. The security classification task aims to label the security-related sentences that in turn can be used as properties and enable semi-automated testing.

5 Experiments and Evaluation

In this section we provide a comprehensive assessment of the proposed methodology through rigorous experimentation and evaluation.

5.1 Experiment Setup

Baseline Models: For baseline models we use base versions of BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019), BART (Lewis et al., 2019), GPT2 (Radford et al., 2019), T5 (Raffel et al., 2019), ALBERT (Lan et al., 2019), CamemBERT (Martin et al., 2019), LongFormer (Beltagy et al., 2020), Pegasus (Zhang et al., 2019); large versions of GPT2 and mBART (Liu et al., 2020); medium version of GPT2; DistilGPT2 and DistilBERT (Sanh et al., 2019).

Pre-trained Models: We pre-train three models- BERT-base, ROBERTa-base, and XLNet-base, on the SPEC5G dataset; we refer to them as BERT5G, ROBERTa5G, and XLNet5G respectively. The reason for training these three models is discussed in Section 7 (Choice of Pre-trained Models). We then fine-tune the pre-trained models for the downstream tasks. The details of the pre-training and fine-tuning are discussed in detail in Section A.2.

Training Hardware: We use Google Colab Pro+ to pre-train and fine-tune the models. Around 3000 computing units (CU) of Premium GPU (A100) with high RAM configuration have been consumed to complete all our experiments. For details about CU and training time, we refer to Appendix A.3.

5.2 Performance Metric

To measure the performance of the sentence classification task we use standard performance metrics such as accuracy, precision, recall, and F1-score. We discuss the metrics for the summarization task here in detail.

5.2.1 Summarization Metrics

To measure the quality of summarization we use both automatic and human evaluation metrics. For automatic evaluation, we use the commonly used ROUGE score.

Human Evaluation Metric: Due to significant dissonance with human evaluation, automatic evaluation metrics are often considered unreliable for summarization quality evaluation. Hence, we resort to human evaluation metrics. The human annotator's rate on a scale from 1 (worst) to 5 (best) on three coarse attributes: (1) *Simplicity:* As the majority of the inferences require speculation, this metric measures how simple and concise the models and the annotators are. (2) *Correctness:* Whether the generated or annotated inferences are grammatically and from a protocol point of view correct. This is very important for the summarization of network protocol specifications. (3) *Contextuality:* Whether the generated or annotated inferences fit the context.

Model	RG-1	RG-2	RG-L
BERT-base	0.484	0.341	0.415
BERT5G	**0.543**	**0.382**	**0.472**
PEGASUS	0.239	0.120	0.199
RoBERTa-base	0.489	0.341	0.418
RoBERTa5G	0.540	0.379	0.469
BART-base	0.357	0.231	0.311
XLNET-base	0.483	0.340	0.416
XLNET5G	0.526	0.362	0.453
GPT2	0.488	0.340	0.418
GPT2-medium	0.481	0.333	0.408
GPT2-large	0.487	0.344	0.418
DistilGPT2	0.483	0.333	0.412
T5	0.444	0.285	0.363

Table 1: Performance of summarization over the 5GSum dataset based on ROUGE scores.

5.3 Performance

We now report the performance of the baseline language models to characterize our dataset.

Summarization: We report the models' mid-score fmeasure of ROUGE-1, ROUGE-2, and ROUGE-L in Table 1 and human evaluation scores in Table 3 to show the comparison of the baseline models with the pre-trained models. BERT5G outperforms all the models, though BERT-base was not the best-performing model. This shows the quality of our dataset for technical specification learning.

Sentence Classification: We report the performance of the models on the sentence classification task in Table 2. For sentence classification with relatively few classes, BERT, ROBERTa, and XLNet perform the best(Chang et al., 2020). Therefore, we pre-train 3 models- BERT-base, ROBERTa-base, and XLNet-base language models on the SPEC5G dataset. These 3 models along with other baselines are then fine-tuned on the 5GSC dataset to compare their classification performance. We observe that BERT5G, ROBERTa5G, and XLNet5G outperform their corresponding baselines by a significant margin. Additionally, BERT5G outperforms all other models in precision and F1 score. XLNet5G has the highest recall. Interestingly, the baseline GPT2 has the highest accuracy. Despite that, we do not

choose GPT2 for pre-training. The reason behind this is discussed in Section 6.

Model	Precision	Recall	F1	Acc
ALBERT-base	0.6485	0.6587	0.6459	0.7034
BART-base	0.6458	0.6503	0.6432	0.7103
BERT-base	0.6113	0.6229	0.6157	0.6897
BERT5G	**0.6972**	0.6762	**0.6856**	0.7655
CamemBERT	0.6107	0.6272	0.6174	0.7034
DistilBERT	0.5819	0.5769	0.5731	0.6621
GPT2	0.6133	0.5567	0.5767	**0.7973**
LongFormer	06280	0.6281	0.6274	0.7034
mBART-large	0.6598	0.6642	0.6606	0.7241
ROBERTa-base	0.5752	0.5562	0.5631	0.6690
ROBERTa5G	0.5944	0.5696	0.5785	0.6966
XLNet-base	0.6260	0.6339	0.6297	0.7034
XLNet5G	0.6480	**0.6829**	0.6619	0.7103

Table 2: Performance of baseline models on classification task over the 5GSC dataset.

6 Result Analysis

Performance Improvements Due to SPEC5G: The primary objective of our work is to introduce an anchor 5G dataset that might pave the way for future NLP research in 5G and NLP. The models pre-trained on 5G and fine-tuned on respective tasks achieve significant performance improvements, suggesting that such dataset should be considered the gold standard for pre-training models before deploying them for more sophisticated, 5G-oriented NLP applications.

Best Pre-trained Model: The scores of both tasks show that BERT5G is the best-performing model. It is not surprising that XLNet5G, the pre-trained version of more recent BERT variant XLNet, is a close competitor. While GPT2 is a good choice for our summarization task, we do not recommend it for security classification. Despite a high accuracy score, GPT2 failed to achieve a contending recall or F1-score. We observe that GPT2 could classify the *Non-Security* samples well (53 out of 70 test samples were correct) which dominates the dataset distribution, while poorly classifying samples from *Security* (11 out of 24 correct) and *Undefined* (4 out of 11 correct). This is the reason for its higher accuracy yet low recall and F1.

Results of Human Evaluation for Summarization: We randomly sample 40 inferences generated by each pre-trained model, their non-pre-trained versions, and corresponding gold inference. These inferences are then manually rated by three independent annotators based on the human-evaluation metrics. As shown in Table 3, we observe that the fine-tuned models perform similarly on SPEC5G but fail to reach gold annotation performance. Moreover, as expected, the pre-trained models significantly outperform their non-pre-trained counterparts. We provide some examples of the generated inferences in Figure 6. Inspection of the model-generated inferences reveals that the usage of keywords from the technical specifications is more frequent in inferences generated by models pre-trained on SPEC5G.

Model	Simplicity	Correctness	Contextuality
Gold	4.37	4.77	4.56
BERT-base	3.2	3.87	3.3
BERT5G	3.96	4.32	3.92
RoBERTa-base	3.7	4.03	3.76
RoBERTa5G	4.02	4.2	3.94
XLNET-base	3.57	3.84	3.53
XLNET5G	3.97	4.31	3.91

Table 3: Result of the human evaluation for SPEC5G.

7 Discussion

Broader Impact of SPEC5G. Our dataset can offer valuable insights and applications that extend beyond the immediate scope. It can be very useful also for other specialized communication protocols (like IoT, Bluetooth, Bluetooth Low Energy, Vehicular Protocols, and WiFi). One popular methodology to evaluate the design of communication protocols is to manually extract a formal model, for example in terms of finite state machines, of the protocol and evaluate the model against the desired security and privacy properties (Hussain et al., 2019). One major issue with this approach is that the manual model extraction from the protocol-text is error-prone and not scalable. Therefore, communication protocols are analyzed partially or within a specific scope. The analysis paved by SPEC5G can yield a deeper understanding of network behavior, interference patterns, and potential optimizations that can be applied to a variety of wireless communication scenarios. This can lead to an ecosystem around our initial dataset, which could include models trained on SPEC5G and additional relevant datasets that could be combined with ours.

The versatility can spark innovation in the design and development of future-generation protocols. There is a lot of work to design technologies and protocols for interconnecting cellular networks and non-terrestrial networks (via for example, low earth orbit–LEO satellites); therefore creating datasets for these new protocols would only require addition to a modest amount of new data to SPEC5G.

Article: The PDCCH channel is used to send control information to the UEs on a per-slot basis. It is used to schedule downlink transmissions, uplink transmissions, MCS of those transmissions, and HARQ information. The PDCCH can appear on any subcarrier; so the jammer must decode the parameter CORESET freq-domain. The parameter CORESET-time-duration, which can take on values 1, 2, or 3, indicates how many OFDM symbols the PDCCH occupies in each slot. The PDCCH always starts in the first symbol of each slot, is QPSK modulated and uses polar coding.

Reference: PDCCH sends per-slot control information to UEs. Schedules downlink, uplink, MCS, and HARQ transmissions. PDCCH can exist on any subcarrier; the jammer must decode CORESET freq-domain. The CORESET-time-duration parameter shows how many OFDM symbols the PDCCH occupies each slot. The PDCCH starts at the first symbol of each slot and uses polar coding.

BERT-base: The PDCCH channel is used to send control information to the UEs on a per-slot basis. **The PDCCH can appear on any subcarrier; so the jammer must decode the parameter CORESET freq-domain.**

BERT5G: The PDCCH channel is used to send control information to the UEs on a per-slot basis. **The parameter CORESET-time-duration, which can take on values 1, 2, or 3, indicates how many OFDM symbols the PDCCH occupies each slot.**

RoBERTa-base: The PDCCH channel is used to send control information to the UEs on a per-slot basis. **The PDCCH can appear on any subcarrier; so the jammer must decode the parameter CORESET freq-domain.**

RoBERTa5G: The PDCCH channel is used to send control information to the UEs on a per-slot basis. **The parameter CORESET-time-duration, which can take on values 1, 2, or 3, indicates how many OFDM symbols the PDCCH occupies each slot.**

XLNet-base: The PDCCH channel is used to send control information to the UEs on a per-slot basis. The PDCCH always starts in the first symbol of each slot, is QPSK modulated and uses polar coding.

XLNet5G: The PDCCH channel is used to send control information to the UEs on a per-slot basis. The PDCCH always starts in the first symbol of each slot, is QPSK modulated and uses polar coding.

Figure 6: Comparison of summarization task by pre-trained models and their base version. **Brown** colored lines denotes the base models inability to capture protocol specific sentences in summaries and **teal** colored lines donotes the sentences introduced by pre-trained models on SPEC5G that are more contextual.

Similarly, models trained on SPEC5G could also be tuned by using a modest amount of new data. By leveraging the insights gained from the interactions within the dataset, researchers and engineers can create protocols that can adapt to evolving communication landscapes. This adaptability will be essential as we move towards more interconnected and heterogeneous networks.

Furthermore, the utility of our dataset reaches beyond those exclusively working on 5G networks. As NLP research and natural language processing techniques continue to evolve, our dataset can serve as a foundation for various research avenues. For instance, the dataset can be employed to develop advanced predictive models, anomaly detection systems, and intelligent network management solutions. These applications are not limited to the realm of 5G but have the potential to influence and enhance NLP research across a broader spectrum.

Choice of Pre-trained Models. The motivation behind the choice of the machine learning models is to show the quality of SPEC5G. Hence we only use pre-existing models for the downstream tasks and do not measure the performance of simple baseline models like lead-3 extractive baseline (taking the first 3 sentences of the article as the summary) and the SummaRuNNer extractive model (Nallapati et al., 2016), nor improve the performance of the downstream tasks. We pick the models that perform well in both downstream tasks. Here the chosen models are all encoder-only to maintain consistency between the experiments. Nevertheless, encoder-decoder models or decoder-only models would also benefit from pre-training. It is well known that pre-training on domain-specific data can help to improve the performance of downstream tasks in the domain (Lee et al., 2019; Chalkidis et al., 2020). However, in our case after the first step of preprocessing, BERT only improves 2.73%, XLNet improvement is 1.96% and ROBERTa improves 0.061% in F1 score. Thus, although we commonly know that pre-training improves downstream tasks, evidently the preprocessing of the dataset signifies that process even more. The performance improvement of the base models after pre-training on our dataset indicates that the models could learn and sufficiently generalize their knowledge in technical specifications. We leave exploring the downstream tasks in detail and the criteria for the selection of different models on the technical specification domain as future work.

Downstream Task Dataset Size. While the downstream task datasets may seem small, recent high-quality manually annotated datasets had similar sizes–COUGH dataset(1236 labeled sentences)(Zhang et al., 2021) and YASO dataset (2215 labeled sentences)(Orbach et al., 2021). Thus, the current size is comparable to the contemporaries. To address the selection bias of the relatively short test set, the test points are randomly sampled on 3 different runs of each model, and the models are run on 3 different random seeds which show low standard deviation in performance metrics. Therefore, the randomness in the test set

removes the selection bias. Moreover, this dataset can easily be used as a seed alongside our trained models for semi-automatic annotation with minimal human effort. Our work enables this direction of using language models in technical specification documents. In the case of the summarization dataset, it is only used as a test set for the models that can already summarize articles. Their performance on summarizing network protocol specification is measured using this test set.

Project Maintenance. In the context of the 3GPP, major releases like 3G, 4G, and 5G are published every ten years. However, smaller, incremental functional changes are made each year within these larger frameworks. These updates are designed to be backward compatible and avoid conflicts with the previous releases, ensuring a smooth transition for existing infrastructure and devices. To address the concern with the new releases, we have devised a plan to maintain the quality and relevance of our dataset in the face of these protocol changes. After each major 3GPP release, we will analyze the changes and updates made to the protocol specifications. For each significant protocol update, we will review and re-evaluate the annotations in our dataset. This will involve identifying any modifications, additions, or clarifications in the protocol specifications. The Change Request (CR) procedure used by 3GPP to create revised versions of 3GPP specifications can be used to automatically identify the modifications. Our team will then update the dataset to accurately reflect these changes. Alongside these updates, we will provide summaries of the changes made in each protocol release. This will serve as a "TL;DR" version highlighting the key modifications that have taken place. This way, users can quickly understand what has changed in the context of each new release. Our commitment is to keep our dataset in sync with the evolving protocols and maintain its utility as a valuable resource for researchers, developers, and industry professionals.

8 Conclusion

We have created SPEC5G–a new dataset for 5G, 5GSum and 5GSC– expert annotated datasets for 5G protocol summarization and 5G security text classification respectively. To show the usefulness of SPEC5G in protocol specifications learning by the Language Models, we design security sentence classification and summarization tasks for state-of-the-art Language Models to solve.

Future Work. Given the specialized nature of 5G terminology in the dataset, it could be utilized for domain adaptation tasks in NLP (for instance, adapting language models to understand and generate content in the context of 5G communications). The dataset could be used to create datasets for named entity recognition tasks, focusing on extracting and categorizing specific entities such as protocols, technologies, companies, and standards relevant to 5G. With the wealth of information present in the dataset, question-answering datasets could also be constructed, where models are trained to answer questions related to 5G concepts, protocols, and technologies. SPEC5G can be used to develop semantic role labeling datasets, assisting in understanding the roles and relationships of various elements in sentences discussing 5G. Datasets for document classification tasks, where the goal is to categorize entire documents or articles based on their content related to 5G concepts can also be created. With content from various sources, the dataset could be used to create parallel corpora for translating technical 5G content between different languages. SPEC5G can be utilized to develop datasets for dependency parsing tasks, improving syntactic analysis and understanding of relationships between words. Generating datasets for topic modeling tasks can help in identifying and categorizing prevalent topics within the 5G domain.

9 Limitations

Here we discuss some limitations we faced.

9.1 Underspecifications in the standards

In this paper, we introduce SPEC5G, a dataset aimed at the automated analysis of the 5G protocol, and show the usefulness of SPEC5G in two downstream tasks. The performance of the two different downstream tasks on the dataset, in turn, depends on the 5G standards. In some cases, the standards are intentionally kept underspecified and contain ambiguities. The reason for such underspecifications and ambiguities is mainly to give vendors flexibility in the implementation design and performance enhancement. Nonetheless, the SPEC5G dataset can include some of the underspecified behaviors from the standards. These ambiguities existing in the text can be resolved using human expertise. This is precisely how we leverage human expertise for the two downstream tasks in

the paper. However, this can be accomplished by using NLP methods that exploit unlabeled data and human knowledge. This is the direction we plan to pursue in the future.

9.2 Automation

The aim of SPEC5G is to help automate the manual-intensive tasks of 5G protocol development, analysis, and testing using state-of-the-art NLP techniques. However, it is evident that it is still not possible to completely automate such tasks because of the manual annotation, which requires domain expertise. In spite of the limited annotated data, we show that it is still possible to achieve fairly good results in two downstream tasks. It may not be possible to completely automate the 5G related tasks, but we still hope it can help reduce the large manual efforts which is the current state-of-the-art.

10 Ethical Considerations

In regards to the datasets being released, all information is in the public domain and is not subject to any copyrights. To pre-train, we use different language models. It has been reported that the pretrained masked language models encode unfair social biases such as gender, racial bias, and religious biases (Bommasani et al., 2020). In our case, as we are dealing with a technical domain, we believe these biases do not have any impact on our results. Moreover, we randomly evaluated the model's outputs and found no evidence of these biases. In the case of annotations, the annotators for SPEC5G are all Ph.D. students doing active research in the area of networks. They are provided with specific guidelines (discussed in detail in Appendix C) and are strictly asked not to write any toxic content (hateful or offensive toward any gender, race, sex, or religion) and to consider gender-neutral settings.

Acknowledgement

We thank the anonymous reviewers for their insightful comments and the annotators: Adrian Li, Charalampos Katsis, Fabrizio Cicala, Mengdie Huang, Sonam Bhardwaj, Yiwei Zhang, and Zilin Shen from cyber2slab of Purdue University for their valuable time and effort in annotating the downstream task datasets. The work reported in this paper has been supported by NSF under grant 2112471 "AI Institute for Future Edge Networks and Distributed Intelligence (AI-EDGE)".

References

2022. *3GPP, Specifications*. https://www.3gpp.org/ftp/Specs.

2022. *5G-Powered Medical Robot Performs Remote Brain Surgery*. https://www.automate.org/blogs/5g-powered-medical-robot-performs-remote-brain-surgery.

2022. *Adopting neural language models for the telecom domain*. https://www.ericsson.com/en/blog/2022/1/neural-language-models-telecom-domain.

2022. *How to Reduce & Fix Gaming Lag*. https://www.screenbeam.com/wifihelp/wifibooster/how-to-reduce-lag-for-gaming-and-improve-your-internet-speed/.

2022. *Selenium*. https://www.selenium.dev/.

Fawad Ahmad, Hang Qiu, Ray Eells, Fan Bai, and Ramesh Govindan. 2020. CarMap: Fast 3d feature map updates for automobiles. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pages 1063–1081, Santa Clara, CA. USENIX Association.

David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. 2018. A formal analysis of 5g authentication. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 1383–1396, New York, NY, USA. Association for Computing Machinery.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting pretrained contextualized representations via reductions to static embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: the muppets straight out of law school. *CoRR*, abs/2010.02559.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *KDD 2020*.

Yi Chen, Di Tang, Yepeng Yao, Mingming Zha, XiaoFeng Wang, Xiaozhong Liu, Haixu Tang, and Dongfang Zhao. 2022. Seeing the forest for the trees: Understanding security hazards in the 3GPP ecosystem through intelligent analysis on change requests. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 17–34, Boston, MA. USENIX Association.

Yi Chen, Yepeng Yao, XiaoFeng Wang, Dandan Xu, Chang Yue, Xiaozhong Liu, Kai Chen, Haixu Tang, and Baoxu Liu. 2021. Bookworm game: Automatic discovery of lte vulnerabilities through documentation analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1197–1214.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019a. Structural scaffolds for citation intent classification in scientific publications. *CoRR*, abs/1904.01608.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019b. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 195–205, Vancouver, Canada. Association for Computational Linguistics.

Will Coster and David Kauchak. 2011. Learning to simplify sentences using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Cas Cremers and Martin Dehnel-Wild. 2019. Component-based formal analysis of 5g-aka: Channel assumptions and session confusion.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Biological, translational, and clinical language processing*, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China. Association for Computational Linguistics.

Syed Rafiul Hussain, Mitziu Echeverria, Imtiaz Karim, Omar Chowdhury, and Elisa Bertino. 2019. 5greasoner: A property-directed security and privacy analysis framework for 5g cellular network protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 669–684, New York, NY, USA. Association for Computing Machinery.

Kokil Jaidka, Muthu Chandrasekaran, Beatriz Fisas, Rahul Jha, Christopher Jones, Min-Yen Kan, Ankur Khanna, Diego Molla Aliod, Dragomir Radev, Francesco Ronzano, and Horacio Saggion. 2014. The computational linguistics summarization pilot task.

Kokil Jaidka, Muthu Chandrasekaran, Devanshu Jain, and Min-Yen Kan. 2017. The cl-scisumm shared task 2017: Results and key insights.

Kokil Jaidka, Muthu Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. 2018. Insights from cl-scisumm 2016: the faceted scientific document summarization shared task. *International Journal on Digital Libraries*, 19:1–9.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. *CoRR*, abs/2005.02324.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yonghao Liu, Renchu Guan, Fausto Giunchiglia, Yanchun Liang, and Xiaoyue Feng. 2021. Deep attention diffusion graph neural networks for text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8142–8152, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2020. Controllable text simplification with explicit paraphrasing. *CoRR*, abs/2010.11004.

Louis Martin, Benjamin Müller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2019. Camembert: a tasty french language model. *CoRR*, abs/1911.03894.

Dheeraj Mekala, Varun Gangal, and Jingbo Shang. 2021. Coarse2fine: Fine-grained text classification on coarsely-grained annotated data. In *EMNLP*.

Derek Miller. 2019. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *CoRR*, abs/1611.04230.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. volume 1.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Matan Orbach, Orith Toledo-Ronen, Artem Spector, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2021. YASO: A targeted sentiment analysis evaluation dataset for open-domain reviews. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9154–9173, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016. Unsupervised lexical simplification for non-native speakers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30.

Sarah E. Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *SLaTE*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Mahadev Satyanarayanan. 2017. The emergence of edge computing. *Computer*, 50(1):30–39.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Advaith Siddharthan and Napoleon Katsos. 2010. Reformulating discourse connectives for non-expert readers. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1002–1010, Los Angeles, California. Association for Computational Linguistics.

Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. pages 243–246.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Sasha Spala, Nicholas A. Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. DEFT: A corpus for definition extraction in free- and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131, Florence, Italy. Association for Computational Linguistics.

Sanja Štajner, Marc Franco-Salvador, Simone Paolo Ponzetto, Paolo Rosso, and Heiner Stuckenschmidt. 2017. Sentence alignment methods for improving

text simplification systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 97–102, Vancouver, Canada. Association for Computational Linguistics.

Sanja Štajner and Horacio Saggion. 2018. Data-driven text simplification. In *Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts*, pages 19–23, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R. Fabbri, Irene Li, Dan Friedman, and Dragomir R. Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. *CoRR*, abs/1909.01716.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Wuyang Zhang, Jiachen Chen, Yanyong Zhang, and Dipankar Raychaudhuri. 2017. Towards efficient edge cloud augmentation for virtual reality mmogs. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, SEC '17, New York, NY, USA. Association for Computing Machinery.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Xinliang Frederick Zhang, Heming Sun, Xiang Yue, Simon Lin, and Huan Sun. 2021. COUGH: A challenge dataset and models for COVID-19 FAQ retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3759–3769, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yiwen Zhang, Caixia Yuan, Xiaojie Wang, Ziwei Bai, and Yongbin Liu. 2022. Learn to adapt for generalized zero-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 517–527, Dublin, Ireland. Association for Computational Linguistics.

Yang Zhong, Chao Jiang, Wei Xu, and Junyi Jessy Li. 2019. Discourse level factors for sentence deletion in text simplification. *CoRR*, abs/1911.10384.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. volume 2, pages 1353–1361.

Appendix

A Preprocessing and Training

In this section, we outline the preprocessing steps undertaken to clean and transform the data, as well as the training methodology employed to optimize the model's performance. By employing rigorous preprocessing and training techniques, we aim to ensure reliable and accurate results in our subsequent analysis.

A.1 Preprocessing Details

The following preprocessing steps were performed-

- Sentences containing codes (e.g. consecutive '{{', '}}', '((', '))') are removed.
- Some of the remaining html tags present after web scrapping are removed.
- Citations and references are removed.
- Sentences mentioning subsequent figures, and tables are removed.
- Figure/table captions are skipped.
- Sentences containing consequent digits and dots refer to (sub)section headers are removed.
- Some malformed texts appearing from figures and tables after parsing text files from .doc or .pdfs files are filtered out.
- Sentences containing Unicode characters appearing as raw texts are removed.
- Multiple consecutive newlines, tabs, whitespace, and delimiters are processed into one.

- Starting numbers, dots, interpuncts, and hyphens appearing from (un)ordered lists are removed.
- Additional whitespaces after opening parentheses, curly braces, and brackets are removed. Similar to closing ones.
- 3GPP specifications contain numerous mentions of specification documents (i.e. TS 24.301). These do not add any useful features for learning. Those are renamed as "specification document".
- If a sentence contains a high amount of digits, they necessarily are from embedded codes. If more than 20% are seen, we filter out the sentence.
- Few special cases (for example: "e.g.,", "i.e.,") are handled to not be considered as the end of a sentence.
- An additional newline is added after adding all texts from each of the documents/web pages. This is to ensure that certain downstream applications (e.g., summarization) do not get affected by unrelated texts from multiple documents.

A.2 Training Details

To pre-train BERT Masked Language Model (MLM), we use the Adam optimizer with $\epsilon = 10^{-8}$ and train the model for 10 epochs. The learning rate is 5×10^{-5}, we set aside 10% of the data as validation to inspect the model performance at every 50k steps. BERTFastTokenizer has been used to tokenize the dataset. We use the same parameters to pre-train ROBERTa MLM for 5 epochs and ROBERTa BPE tokenizer to tokenize the dataset for this setting. For pre-training XLNet Permutation Language Model (PLM), we use the Adam optimizer in the same setting. Since XLNet requires approximately 5 times more than BERT or ROBERTa, we train the model for 1 epoch. We use the SentencePiece tokenizer in this case.

When fine-tuning the classification models, we set the learning rate to be 2×10^{-5}, weight decay to be 0.01, and batch size to be 16. The Huggingface standard pipeline with the Automodel class has been used for sequence classification. We train each model for 15 epochs.

We use the bert-extractive-summarizer (Miller, 2019) to generate summaries using BERT-base. The Huggingface standard pipeline libraries has been used to generate summaries using sequence-to-sequence models i.e., PEGASUS and T5 that comes with default summarization capability. To generate summary using RoBERTa-base, RoBERTa5G, XLNet5G, and BERT5G, we use the Huggingface Automodel Library. We use another Huggingface library TransformerSummarizer to generate summary using XLNet, GPT2, GPT2-base, GPT2-medium, GPT2-large and DistilGPT2.

A.3 Compute Unit and Training Duration

A compute unit (CU) is the unit of measurement for the resources consumed. To calculate CUs, one needs to multiply two factors: (1) Memory (GB) - size of the allocated server for task to run and (2) Duration (hours) - how long the server is used. This means, 1 CU = 1 GB memory x 1 hour. We have used around 80-90% of the GPU during training time. By definition of computing units, we have used roughly 100 hours of 30GB GPU time.

Pre-training BERT takes around 36 hours in our experimental setup. Pre-training RoBERTa and XLNET takes around 24 hours each. Fine-tuning each model takes around 5-6 hours.

B Performance Evaluation of Downstream Tasks

We report the evaluation of performance and metrics used for it in this section.

B.1 Performance Metrics

For automatic evaluation, we use the commonly used ROUGE score. We use the Python rouge_score library to calculate this. **ROUGE Score:** ROUGE-N measures the number of matching 'n-grams' between the model-generated text and a 'reference'. An n-gram is simply a grouping of tokens/words. A unigram (1-gram) would consist of a single word. A bigram (2-gram) consists of two consecutive words. In ROUGE-N, N denotes the n-gram that is being used. For ROUGE-1 the match rate of unigrams between the model output and reference are measured. ROUGE-2 and ROUGE-3 would use bigrams and trigrams respectively.

Recall: The recall counts the number of overlapping n-grams found in both the model output and reference, then divides this number by the total number of n-grams in the reference.

$$recall = \frac{count_n(gram_n)}{count(gram_n)}$$

Precision: We use the precision metric — which is calculated in almost the exact same way as recall,

but rather than dividing by the reference n-gram count, it is divided by the model n-gram count.

$$precision = \frac{num\ of\ ngrams\ in\ model\ \&\ ref}{num\ of\ ngrams\ in\ model}$$

Now that both the recall and precision values are available, they can be used to calculate the ROUGE F1 score with the following formula:

$$2 * \frac{precision * recall}{precision + recall}$$

ROUGE-L: ROUGE-L measures the longest common subsequence (LCS) between the model output and the reference. With this metric, the number of tokens in the longest sequence shared between both are counted. The idea here is that a longer shared sequence would indicate more similarity between the two sequences. The recall and precision calculations can be applied just like before — but this time the match is replaced with LCS.

$$recall = \frac{LCS(gram_n)}{count(gram_n)}$$

C Annotation Guidelines

Below are the specific guidelines that we have given to the annotators to ensure the standard of annotation.

C.1 Sentence Classification Guidelines

The annotators are given some general guidelines and are suggested to follow some steps to make the data annotation consistent. They are also provided with some rules and tips.

General Guidelines: For this task, an annotator is given a set of sentences. Based on the methods, fields, variables, and/or entities mentioned in the sentence, the annotator's objective is to identify if the sentence implies a potential security concern or sophisticated operation that might involve vulnerable consequences.

Steps:

1. Read the sentence carefully.
2. Identify items and the operation that involve a security issue.
3. Decide the label based on the following:
    a) *Non-Security*: The expressed operation cannot be exploited/ The sentence does not describe any security hazard/ does not describe any underspecified criteria/ does not involve complicated, flawed operations.
    b) *Security*: The text discusses situations or operations that might be risky/ The text involves certain properties or variables, exploiting which, one can seriously block the operations, or harm the entity, or breach privacy.
    c) *Undefined*: The discussed operation is not clear/ The sentence does not express all the parties or variables involved/ The sentence entails some previous operation unavailable to the annotator- without which the annotator can not decide about the potential risk.
4. If the sentences do not express any proper context or are semantically incorrect, or have no items with a sentiment expressed towards them, add a comment and proceed to the next data.

Rules and Tips:

- Select all items in the sentence that have a security hazard.
- If there are multiple such cases, you may choose any or all of them.
- Optionally, you may provide a comment about your rationalization or feedback about the data (e.g., errors, unclear descriptions.)

C.2 Summarization Guidelines

Similar to classification guidelines, the annotators are given general guidelines and suggested steps to annotate the summarization dataset. Again, they are also provided with some rules and tips. Below are the guidelines for annotation tasks for summarization.

General Guidelines: For this task, an annotator is given a set of articles. Based on the methods, fields, variables, and/or entities mentioned in the sentence, the annotator's objective is to summarize the article without losing important information, correctness and contextuality.

Steps:

1. Read the article carefully.
2. Identify the key points.
3. Summarize the article by doing the following:
    a) Deletion: Delete a sentence if it does not convey any important information.

b) Merge and shorten: Merge consecutive sentences if they convey continued information and make the merged sentences concise.

c) Rephrase and shorten: Rephrase a sentence to make it simpler and make it shorter if possible.

4. If the sentences do not express any proper context, or are semantically incorrect, add a comment and proceed to the next sentence.

Rules and Tips:

- Select all items in the article that have important information.
- Make the sentences simpler and concise keeping the important information.
- Under each article is a comments box. Optionally, you can provide article-specific feedback in this box. This may include a rationalization of your choice, a description of an error within the article, or the justification of another answer which was also plausible. In general, any relevant feedback would be useful and will help in improving this task.

C.3 Annotator Agreement

In total nine annotators have annotated the summarization dataset. Each of them is given 70 non-overlapping distinct articles. So there is no disagreement between annotators. Another round of manual cleaning has been done by two meta annotators who have gone through the whole dataset to ensure summarization quality and consistency, by addressing the comments and suggestions made by the annotators in the first round and making necessary changes(update/delete). For example in first round of annotation, annotators put comments like - "The paragraph is vague", "Independent sentences", "The paragraph does not have a logical flow. It cannot be further summarized", "It is not clear what the paragraph is talking about", etc. These comments are addressed by the meta annotators by manually correcting or removing the articles.

For the classification task, 3 annotators (we call them A1, A2, A3 here) separately annotate the dataset- A1 and A2 annotate 800 examples each and A3 annotates 801 examples. In the second step, they are assigned to reevaluate the annotations of each other (A1 reevaluating labels assigned by A3, A2 reevaluating labels assigned by A1, and A3 reevaluating A2). Such reevaluations bring forth disagreements on several labels which are finally resolved by their combined discussion. For example: "The AMF shall not indicate to the SMF to release the emergency PDU session.": A2 labels this as Security, while A3 assigns Undefined. This disagreement is later resolved by discussing their reasoning for the respective labels.

C.4 Examples

We are listing some example annotated data for both tasks.

C.4.1 Sentence Classification:

Here we show a few examples of sentence classification–each containing a sentence and the correct label associated with it.

Sentence 1: If the positioning method parameter indicates both E-Cell ID and GNSS positioning, the eNB may use E-Cell ID measurement collection only if the UE does not provide GNSS-based location information.

Label 1: *Security*

Sentence 2: SIGN_VAR shall be included in the channel quality report.

Label 2: *Non-Security*

Sentence 3: After performing the attach, the MS should activate PDP context(s) to replace any previously active PDP context(s).

Label 3: *Security*

Sentence 4: It switches the user from the UTRAN user plane to the GAN user plane

Label 4: *Undefined*

Sentence 5: If the BSIC cannot be decoded at the next available opportunities re attempts shall be made to decode this BSIC.

Label 5: *Non-Security*

Sentence 6: This might lead to an empty or even absent structure, if no parameter was modified.

Label 6: *Undefined*

C.4.2 Summarization:

Here are a few examples, each containing an article and its summary.

Article 1: As indicated, 5G NR Meas Gap Length is not fixed and 3GPP specifications made it configurable. Having a fixed Meas Gap could cause unnecessary degradation of throughput in the serving cell. The SMTC window and window duration can be set to match SSB transmissions and accordingly, the MGL. For example, if we consider the SMTC window duration as 2 ms and the Meas Gap Length as 6 ms, here 4 ms segment would not be available for transmission, and reception of data in the serving cell will result in low DL/UL throughput.

Summary 1: 5G NR Meas Gap Length is adjustable per 3GPP specs. A fixed Meas Gap can degrade serving cell throughput. The SMTC window and duration can match SSB transmissions and the MGL. If the SMTC window duration is 2 ms and Meas Gap Length is 6 ms, a 4 ms segment is not accessible for transmission, resulting in limited DL/UL throughput.

Article 2: IMSI-catching attacks have threatened all generations (2G/3G/4G) of mobile telecommunication for decades. As a result of facilitating backward compatibility for legacy reasons, this privacy problem appears to have persisted. However, the 3GPP has now decided to address this issue, albeit at the cost of backward compatibility. In case of identification failure via a 5G-GUTI, unlike earlier generations, 5G security specifications do not allow plain-text transmissions of the SUPI over the radio interface. Instead, an Elliptic Curve Integrated Encryption Scheme (ECIES)-based privacy-preserving identifier containing the concealed SUPI is transmitted. This concealed SUPI is known as SUCI (Subscription Concealed Identifier).

Summary 2: Unlike earlier generations, in the case of identification failure via a 5G-GUTI, 5G security specifications do not allow plain-text transmissions of SUPI over the radio interface. Instead, an Elliptic Curve Integrated Encryption Scheme (ECIES)-based privacy-preserving identifier containing the concealed SUPI (also known as SUCI) is transmitted.

Article 3: A SUPI is usually a string of 15 decimal digits. The first three digits represent the Mobile Country Code (MCC) while the next two or three form the Mobile Network Code (MNC), identifying the network operator. The remaining (nine or ten) digits are known as Mobile Subscriber Identification Numbers (MSIN) and represent the individual user of that particular operator. SUPI is equivalent to IMSI, which uniquely identifies the ME, and is also a string of 15 digits.

Summary 3: SUPI is a string of 15 decimal digits consisting of the Mobile Country Code, Mobile Network Code, and Mobile Subscriber Identification Number. SUPI is equivalent to IMSI which uniquely identifies the ME.

Article 4: Next-generation 5G cellular systems will operate in frequencies ranging from around 500 MHz up to 100 GHz. Till now, with LTE and Wi-Fi technologies, we were operating below 6GHz and the channel models were designed and evaluated for operation at frequencies only as high as 6 GHz. The new 5G systems are to operate in bands above 6 GHz and existing channel models will not be valid, hence there is a need for accurate radio propagation models for these higher frequencies, which requires new channel models. The requirements of the new channel model that can support 5G operation across frequency bands up to 100 GHz are based on the existing 3GPP channel models along with extensions to cover additional 5G modeling requirements.

Summary 4: 5G will operate in frequencies ranging from around 500 MHz up to 100 GHz. Up to now 4G and WiFi were operating below 6GHz and the channel models were designed and evaluated for operation at frequencies only as high as 6GHz.

Article 5: Carrier Aggregation (CA) increases the bandwidth by combining several carriers. Each aggregated carrier is referred to as a Component Carrier (CC). 5G NR CA supports up to 16 contiguous and non-contiguous CCs with different numerologies in the FR1 band and in the FR2 band. A Carrier aggregation configuration includes the type of carrier aggregation (intra-band, contiguous or not, or inter-band), the number of bands, and the bandwidth class. CA Bandwidth Class is a series of alphabets that defines the minimum and maximum

bandwidth along with the number of component carriers.

Summary 5: Carrier Aggregation (CA) increases the bandwidth by combining several carriers. 5G NR CA supports up to 16 contiguous and non-contiguous CCs with different numerologies.

C.5 Data Sources

Below we list the websites that were scrapped to create SPEC5G.

Portal Name	Web Address	Description	# of Sentences	# of Words
Artiza Networks	artizanetworks.com	ArtizaNetworks contains tutorials about 3G, 4G, and 5G Radio Access Network (RAN) and Core Network (CN).	383	5182
Event Helix	eventhelix.com	Event Helix is a private corporation based on Maryland. They develop tools for networking and distributed systems and host numerous blogs about 5G Radio, TCP/IP, and so on.	595	6691
3G LTE Info	3glteinfo.com	3G LTE Info offers tutorials and articles for network professionals. These articles encompass GSM, GPRS, 3G, LTE, 5G, Bluetooth, and so on.	790	9651
4G 5G World	4g5gworld.com	Powered by NgnGuru Solutions Pvt. Ltd., 4G 5G World delivers news, reports, and tutorials about 4G and 5G advanced technologies.	80	1069
Info NR LTE	info-nrlte.com	Run by telecom experts, Info NR LTE delivers technology overviews about NR LTE and NR 5G.	508	7431
Resurchify	resurchify.com	Resurchify contains research gatherings from conferences, journals, symposiums, meetings from multiple sectors.	138	1815
Share Tech Note	sharetechnote.com	ShareTechNote aims to be a reference guideline on numerous fields, such as, programming languages, engineering, mathematics, advanced technologies. 5G is one of them.	8325	91575
Telecompedia	telecompedia.net	Telecompedia is a tutorial resource written by 4G, 5G, and radio experts from Rakuten Mobile on different 5G related technologies such as D-RAN, Open-RAN, power control etc.	2173	24997
RF Wireless	rfwireless-world.com	Following IEEE and 3GPP standards, RF Wireless hosts articles, tutorials, source code, terminologies about wireless technologies.	610	8032
Tech Play On	techplayon.com	Tech Play On contains technology news and guidelines on 5GNR and LTE.	5220	89035
Telecom Hall	telecomhall.net	A forum to discuss the advances on telecom domain and to guide developers or practitioners.	7982	127712
How LTE Stuff Works	howltestuffworks.blogspot.com	Hosts numerous blogs about 5GNR and LTE.	4213	65520
Pro-Developer Tutorial	prodevelopertutorial.com	Delivers tutorials about C/C++, Git, System design, 4G LTE, 5GNR, shell-scripting, etc.	3178	31708

Table 4: List of blogs & forums crawled as part of dataset collection