applied sciences

MDPI

Article

An Explainable Method for Automatic Extraction of Natural Language Access Control Policy Key Components

Luca Petrillo [1,4,*], Fabio Martinelli [2], Antonella Santone [3] and Francesco Mercaldo [3,4,*]

[1] IMT School for Advanced Studies Lucca, 55100 Lucca, Italy
[2] Institute for High Performance Computing and Networking, National Research Council of Italy (CNR), 87036 Rende, Italy; fabio.martinelli@icar.cnr.it
[3] Department of Medicine and Health Sciences "Vincenzo Tiberio", University of Molise, 86100 Campobasso, Italy; antonella.santone@unimol.it
[4] Institute for Informatics and Telematics, National Research Council of Italy (CNR), 56124 Pisa, Italy
* Correspondence: luca.petrillo@iit.cnr.it (L.P.); francesco.mercaldo@unimol.it (F.M.)

Abstract

Access control schemes and models are essential tools for system administrators to protect the integrity of the information. However, they are frequently articulated in natural language, which is a powerful form that guarantees flexibility and expressiveness; however, their inherent ambiguity and unstructured nature pose significant challenges for automated enforcement and rigorous analysis. In this study, we evaluated several transformer-based models for the automated extraction of key components of Natural Language Access Control Policy (NLACP). To this end, we relied on a labeled dataset comprising software requirements specifications from different sectors, such as healthcare and conference management systems. We then conducted a fine-tuning phase, where the BERT model demonstrated optimal performance in extracting entities within a 3-entity paradigm, achieving an F-Measure value of 0.89. ModernBERT proved to be the most promising model in the more complex 5-entity extraction task, with a maximum F-Measure score of 0.84. Furthermore, we introduce an explainability step using layer-wise integrated gradients to gain insight into the decision-making process of these deep models, ensuring that the extracted policy components are both accurate and interpretable.

Keywords: security; access control; transformer; explainability

Academic Editor: George Drosatos

Received: 9 October 2025
Revised: 30 October 2025
Accepted: 4 November 2025
Published: 7 November 2025

Citation: Petrillo, L.; Martinelli, F.; Santone, A.; Mercaldo, F. An Explainable Method for Automatic Extraction of Natural Language Access Control Policy Key Components. *Appl. Sci.* **2025**, *15*, 11854. https://doi.org/10.3390/app152211854

1. Introduction

Access control is a fundamental security mechanism adopted in almost all organizations and is used to protect access to all sensitive resources by regulating who can perform a specific action on a resource or a set of resources [1]. These systems are essential since they help manage the permissions for legitimate users and/or processes to retrieve and manipulate the data [2]. Throughout the decades, numerous formal models have been developed and met different organizational requirements, with different compromises among expressiveness, manageability, and complexity for rules enforcement in the field of access control. Among these models, a widely used one is the Role-based Access Control (RBAC), in which permissions and actions are granted based on the assigned role for an individual within the system. The process is eased through such a form of authorization management, and specific roles can be assigned for the granted accessibility for effective administration and tightened security controls. Unlike the RBAC, Attribute-based Access Control (ABAC) is another authorization framework that evaluates specific characteristics

or attributes for protecting objects like data, information technology resources, and network devices from unwarranted actions and accessibility.

Given the importance of such security mechanisms, system architects must ensure that access control schemes and models run efficiently and reliably to prevent unauthorized use and maintain the integrity of the information. In real-world scenarios, particularly during the initial requirements elicitation phase of system design, access control rules are defined in the form of policies, i.e., articulated in natural language. Although this format is compelling because it offers excellent flexibility, it also introduces ambiguity [3]. While intuitive for human understanding, this policy's definition presents a significant challenge for its automated application due to its possible imprecision and lack of structured semantics [4]. This scenario represents a semantic gap between how policies are initially described and how they are ultimately enforced. In fact, to be used effectively, these policies must be implemented in IT systems using a machine-enforceable format that removes ambiguity and enables deterministic decision-making. In this step, these rules must be converted into a structured configuration, such as XACML, Rego, or cloud IAM templates that policy decision points can evaluate at runtime.

A crucial step in this translation process is the extraction from these Natural Language Access Control Policies (NLACPs) of the key components, i.e., subjects (who), actions (what), and resources (which) [5]. This triplet represents the central elements to express these rules and is also the initial phase to develop a suitable structured representation that can be integrated into automated systems. In the Natural Language Processing (NLP) context, transformer models have become the state-of-the-art for various tasks [6]. They are able to learn rich contextual representations of tokens and can capture long-range dependencies, making them suitable for parsing NLACPs. These characteristics make transformers well-suited to analyze and identify subtle lexical distinctions (e.g., "view" vs. "export") to determine the enforcement semantics. In this context, these models can be fine-tuned on a specialized dataset, which consists of adapting their pre-trained general knowledge to a specific task using a smaller task-specific dataset. Fine-tuned transformer encoders like BERT variants or lightweight distilled models can be trained to extract token roles in policy statements with greater precision than shallow statistical extraction or rule-based extraction.

Given these premises, and to automate this extraction stage, we employed transformer-based models for a named-entity-recognition (NER) task by parsing NLACPs. To this end, we relied on a labeled dataset composed of access control policies extracted from different software requirements documents to identify subjects, actions, and resources. Finally, to analyze the behavior of these models, we evaluate their performance and employ an explainability phase to shed light on their decision-making process, ensuring that the extracted policy components are both accurate and interpretable, aligning with similar efforts in other domains like privacy policy analysis.

The paper proceeds as follows: in the next Section, we present the methodology used in this work. Section 3 presents the results of the experimental analysis along with the datasets and models used. Section 4 reports the current state-of-the-art literature about extracting attributes from NLACPs. Finally, the conclusion and future research plan are presented in the last Section.

2. The Method

This Section provides an overview of the proposed method of identification and extraction of the key components from NLACPs, which is composed of two main steps.. In the first step (Figure 1), we composed and preprocessed the dataset and then fine-tuned and tested four different transformer-based models. As a final step (Figure 2), we evaluated

the effectiveness of the models. Additionally, we employed an explainability technique that shows how different parts of the input contribute to the model's predictions, using the Layer Integrated Gradients (LIG).

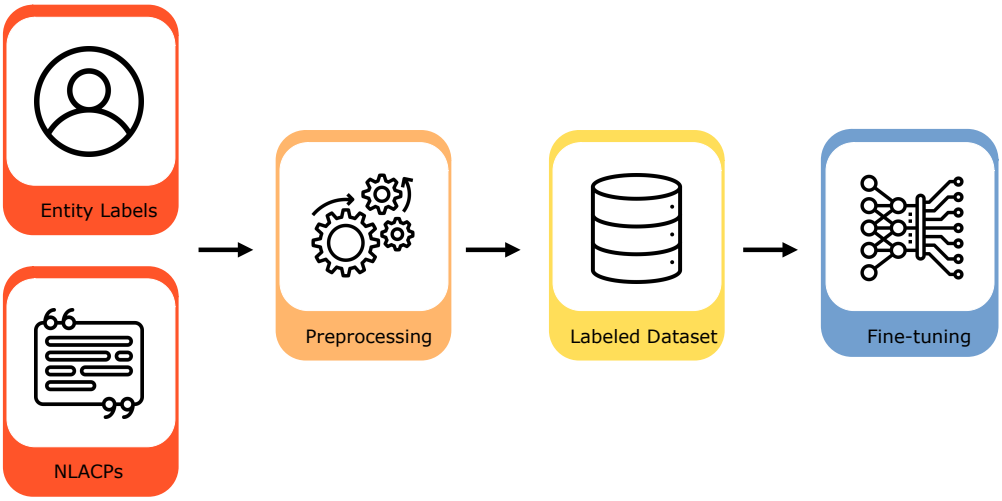


Figure 1. Models training step.

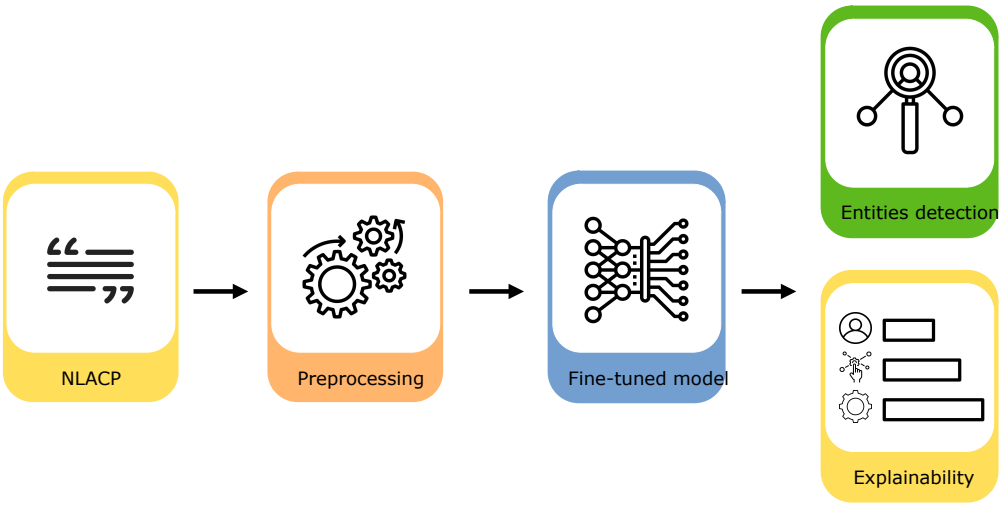


Figure 2. Models evaluation and explainability.

2.1. Hyperparameter Tuning

Before the effective fine-tuning phase, we performed a hyperparameter tuning phase, which helped us select the best configurations to achieve higher performance. Our hyperparameter search examined learning rate, batch size, optimizer, and two regularization techniques, such as dropout rate and label smoothing, which help to prevent overfitting, reduce model confidence, and improve generalization. Table 1 shows the hyperparameter space selected to conduct the experiments to optimize the models. To perform this step, we employed an open-source framework called Optuna [7], which efficiently explores the defined search space through Bayesian optimization to identify optimal configurations. Initially, it employs a random sampling approach, which entails randomly selecting hyperparameter values from the defined range. Following this stage, the framework leverages the outcomes and configurations from these preliminary trials to construct the previously mentioned surrogate model, thereby enhancing the guidance of the search.

Table 1. Hyperparameters and the search space values used in this work. The values in brackets indicate the maximum and minimum values considered.

Learning Rate	Weight Decay	Dropout Rate	No. of Epochs	Batch Size	Optimizer
$[1 \times 10^{-5}, 5 \times 10^{-5}]$	$[1 \times 10^{-8}, 0.3]$	[0.1, 0.5]	[5, 20]	8, 16	sgd, adamw_torch, adamw_torch_fused

2.2. Explainability

After identifying the optimal configurations and the fine-tuning phase, we employed an explainability step, which helps to understand the rationale behind the classifications, particularly in the access control domain, where, in these sensitive domains, trust and transparency are fundamental [8]. This step is essential since it guarantees that the extracted key components from NLACPs are interpretable as well as accurate. In this context, to understand the models' decision-making process, we relied on the Layer Integrated Gradients (LIG) method, first introduced in the context of image processing [9] and now adapted for the NLP field. It is a technique that provides a way to understand the importance of each feature in the model's predictions, thereby elucidating which specific tokens in the NLACP influenced the identification of the key components analyzed. In our scope, we used this technique to understand how individual words within the policy contribute to the model's classification of subjects, actions, resources, conditions, and purposes. It also helped us understand, in a misclassification case, why a specific entity was not wrongly classified and what factors were involved.

3. Experimental Analysis

In this Section, we present a detailed experimental analysis of our approach, evaluating the performance of fine-tuned transformer models on the task of extracting access control policy components from natural language descriptions. To perform the experiments, we used a machine equipped with an Intel Core i7-11th Generation CPU and an NVIDIA GeForce RTX 3070 GPU with 8GB of RAM. Regarding implementation, we utilized our framework proposed in [10], which enabled us to perform the hyperparameter tuning, fine-tuning, and explainability phases.

3.1. Dataset

In this work, we relied on a corpus of NLACPs introduced by Xiao et al. [11] and Slankas et al. [12], comprising sentences from high-level requirement specifications and different contexts. The sentences extracted from those requirements documents come from different domains, i.e., conference management software, course registration system, and software from the healthcare domain. In the first phase, we filtered and extracted only the sentences related to the context of access control. In the second step, we converted all the text to lowercase, removed all special characters, removed any leading non-letter characters, replaced tabs with spaces, and collapsed multiple spaces into one. Table 2 shows the statistics for the sources of the dataset and the total elements.

We utilized two complementary annotated corpora to fine-tune transformer models for the named entity recognition task, adhering to the IOB (Inside, Outside, Beginning) tagging schema. This schema is employed in natural language processing to categorize words within a sentence based on their function in linguistic structures, including noun or verb phrases. Each word is classified to denote whether it is at the start of a block (B), within a block (I), or external to any block (O). Finally, both datasets were split into training, test, and validation sets using the 80-10-10 criteria. This first dataset (D1) is aligned with prior research and is labeled with canonical triplet designations such as subject, action, and resource, thus establishing a baseline for evaluating the efficacy of existing methodologies.

Figure 3 shows the number of annotated entities for each split of this first dataset, where for the training set, we obtained 23,118 entities labeled as "Outside", meaning that the entity is not part of any named entity span. At the same time, we obtained 800 subject entities and 871 action entities.

Regarding the second dataset (D2), we created it to capture a richer policy semantics [13] and it is annotated with five entity-types: subject, action, resource, condition, and purpose. We built this second version starting from the annotations of Jayasundara et al. [14], where they parsed and generated machine-readable access control policies from natural language sentences, utilizing Large Language Models. To align their dataset structure to the IOB tagging scheme, we employed an automatic script that first downloads the necessary resources for part-of-speech (POS) tagging and lemmatization, then it maps POS tags from the Treebank [15] format to WordNet [16] format, which is used for lemmatization. We then tagged the list of sentence tokens with their respective POS and reduced them to their base forms. Finally, we assigned the IOB tags to tokens based on their alignment with specified roles and values from input frames. As shown in the example of Listing 1, this systematic approach enables the transformation of structured annotations into an IOB tagging scheme. Figure 4 shows the statistics of these five tags for each split of the second dataset, where, in this case, we obtained 23,118 outside entities for the training set and 127 entities labeled as "Purpose" and 102 as "Conditions". Tables 3 and 4 show statistics related to D1 and D2, respectively, specifically the number of sentences analyzed for each split, the number of entities analyzed, and the relative proportion of classes.

Table 2. Dataset sources and their relative number of access control policy sentences.

Source	No. of Sentences
iTrust Acre	437
iTrust T2P	344
Collected ACP	116
CyberChair	104
IBM	104
Total elements	1105

Table 3. Statistics for each split of D1, including the number of sentences, the number of entities, and the relative class proportion.

Metric	Training Set	Test Set	Validation Set
No. of sentences	883	111	111
No. of entities	13,972	1759	1759
Entity	**%Author: I confirm, many thanks.hlCount (Proportion %)**		
O	11,559 (82.73)	1453 (82.60)	1453 (82.60)
SUBJECT	721 (5.16)	91 (5.17)	91 (5.17)
ACTION	869 (6.22)	109 (6.20)	109 (6.20)
RESOURCE	823 (5.89)	106 (6.03)	106 (6.03)

Table 4. Statistics for each split of D2, including the number of sentences, the number of entities, and the relative class proportion.

Metric	Training Set	Test Set	Validation Set
No. of sentences	883	111	111
No. of entities	12,438	1642	1642
Entity	**Count (Proportion %)**		
O	9126 (73.37)	1216 (74.06)	1216 (74.06)
SUBJECT	866 (6.96)	102 (6.21)	102 (6.21)
ACTION	1045 (8.40)	136 (8.28)	136 (8.28)
RESOURCE	1172 (9.42)	161 (9.81)	161 (9.81)
PURPOSE	127 (1.02)	14 (0.85)	14 (0.85)
CONDITION	102 (0.82)	13 (0.79)	13 (0.79)

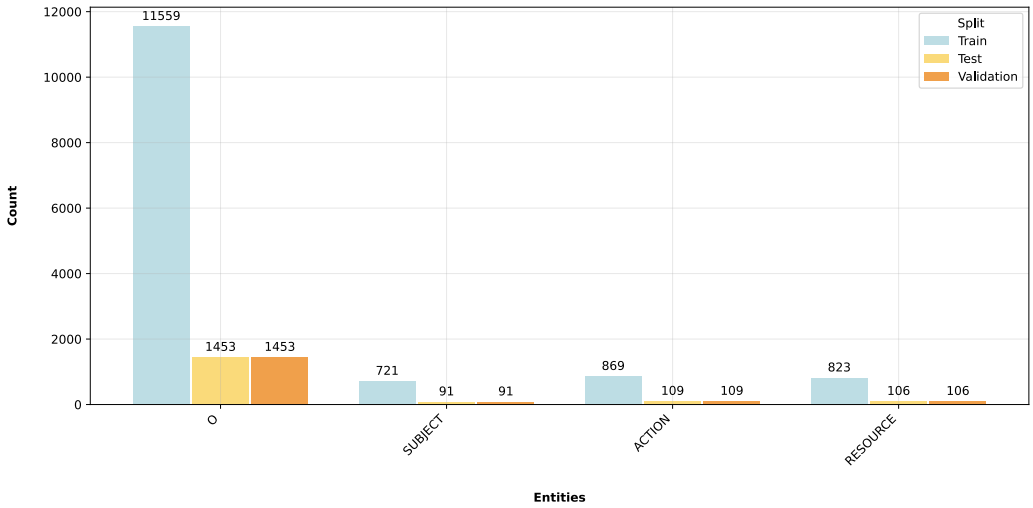


Figure 3. Number of entities in the first dataset (D1) for each split.

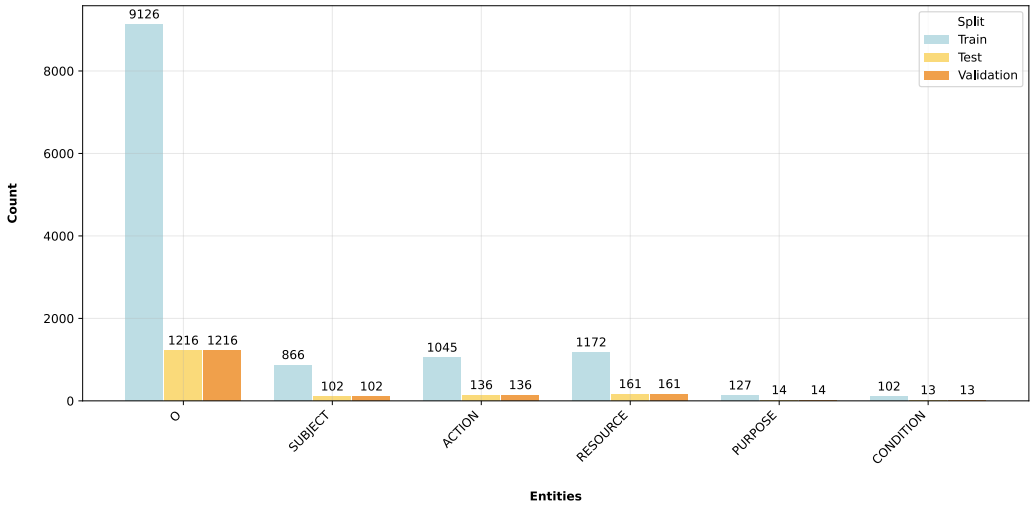


Figure 4. Number of entities in the second dataset (D2) for each split.

Listing 1. Example of dataset alignment between the annotations from [14] and the IOB tagging scheme.

```
{
"NLACP": "the pc chair can assign a paper to a pc member for reviewing except if he
  is one of its authors"
  "initial annotation": {"subject": "pc chair", "action": "assign", "resource": "paper",
  "purpose": "reviewing", "condition": "he is one of its authors"},
  "final annotation": {
    "tokens": ["the", "pc", "chair", "can", "assign", "a", "paper", "to", "a", "pc",
    "member", "for", "reviewing", "except", "if", "he", "is", "one", "of",
    "its", "authors"],
    "tags": ["O", "B-SUBJECT", "I-SUBJECT", "O", "B-ACTION", "O", "B-RESOURCE", "O", "O",
    "O", "O", "O", "B-PURPOSE", "O", "O", "B-CONDITION", "I-CONDITION", "I-CONDITION",
    "I-CONDITION", "I-CONDITION", "I-CONDITION"]
  }
}
```

3.2. Results

This Section reports the results obtained for both datasets after the hyperparameter tuning phase. Section 3.2.2 illustrates the results obtained after the fine-tuning phase on the test set using the optimal configurations acquired during the previous stage. Finally, we also report the explainability results of both models trained on the different datasets.

3.2.1. Hyperparameter Tuning Results

As mentioned in Section 3.1, we performed a hyperparameter tuning phase for both datasets by employing a Bayesian optimization technique. This approach allowed an efficient exploration of the hyperparameter space to identify the optimal configurations and enhance the model's performance. In this phase, we performed 50 trials by maximizing the F-Measure of four different transformer-based models (BERT, ELECTRA, ModernBERT, and RoBERTa). Table 5 summarizes the results of the employed model for each dataset, showing the maximum F-Measure reached with the corresponding trial. As can be seen, for the first dataset (D1), BERT achieved the highest value of 0.909. In contrast, for the second dataset (D2), ModernBERT achieved the highest value (0.867) together with RoBERTa, but with fewer trials. To evaluate the effectiveness of this stage, we analyzed the optimization history of these two best-performing models for each dataset (BERT and ModernBERT) over the 50 trials. Figure 5 shows the entire process of BERT for the dataset D1, which began with an initial configuration that yielded a low F-Measure (below 0.1), because the Bayesian optimization initially explores the search space randomly. At trial 5, a significant improvement was observed, where the value increased to 0.9. From this point onward, the optimization process entered a stable plateau, indicating that the algorithm had converged to a near-optimal set of hyperparameters for the respective models and dataset, until trial 41, where the F-Measure reached its peak of 0.909.

In order to understand the reason for the rapid performance improvement, we analyzed the effect of each hyperparameter and identified the optimizer as the most dominant factor. As illustrated in Figure 6 , the "sgd" trials are confined at the bottom with low results, while the trials with the other two optimizers are clustered at the top. The poor performance of "sgd" was not only because it was used in the first trials (where the hyperparameter search is done randomly), but also around trial 30–35, where the results were still poor. Conversely, with "adam_torch" and "adam_torch_fused", the model consistently performed above 0.85 values.

Table 5. The maximum F-Measure reached along the trial number for the hyperparameter search on both datasets.

Model	Max. F-Measure	Trial Number
	D1	
bert-base-uncased	**0.909**	**41**
electra-base-generator	0.895	49
ModernBERT-base	0.899	34
roberta-base	0.892	49
	D2	
bert-base-uncased	0.850	41
electra-base-generator	0.851	11
ModernBERT-base	**0.867**	**19**
roberta-base	0.867	27

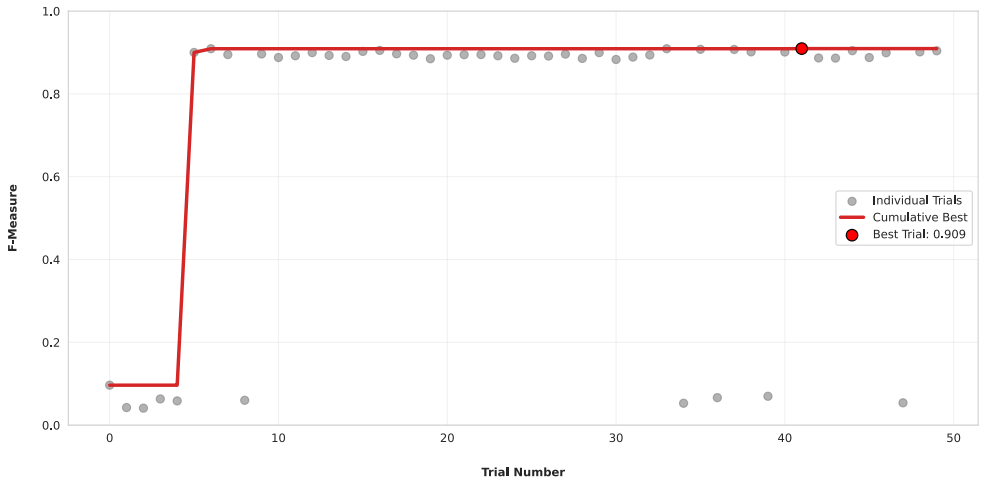


Figure 5. Optimization history of the BERT model on the dataset D1.

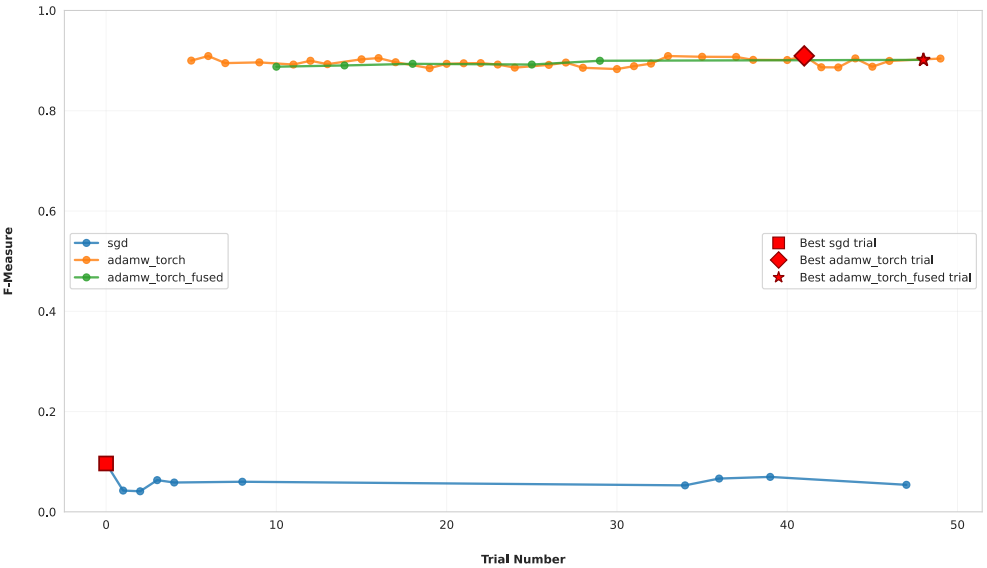


Figure 6. Optimization results per trial of each optimizer of the BERT model on the dataset D1.

As for BERT, we also performed the same analysis for Modern-BERT for the dataset D2, which shows that the search was highly effective. Figure 7 illustrates the optimization history of this second model, where the F-Measure increased within the first 5–10 trials,

achieving values near 0.85. After that, the search converged to a plateau, probably because the region of hyperparameter space was thoroughly explored. The subsequent trials, in fact, did not yield significant improvements. As before, in this case (Figure 8), the optimizer played a significant role in the performance, with adaptive optimizers consistently outperforming stochastic gradient descent. As in the previous case, the "sgd" trials are clustered at the bottom, consistently failing to achieve a meaningful F-Measure value. Meanwhile, the "AdamW" optimizers are clustered at the top, achieving almost high F-Measure values.

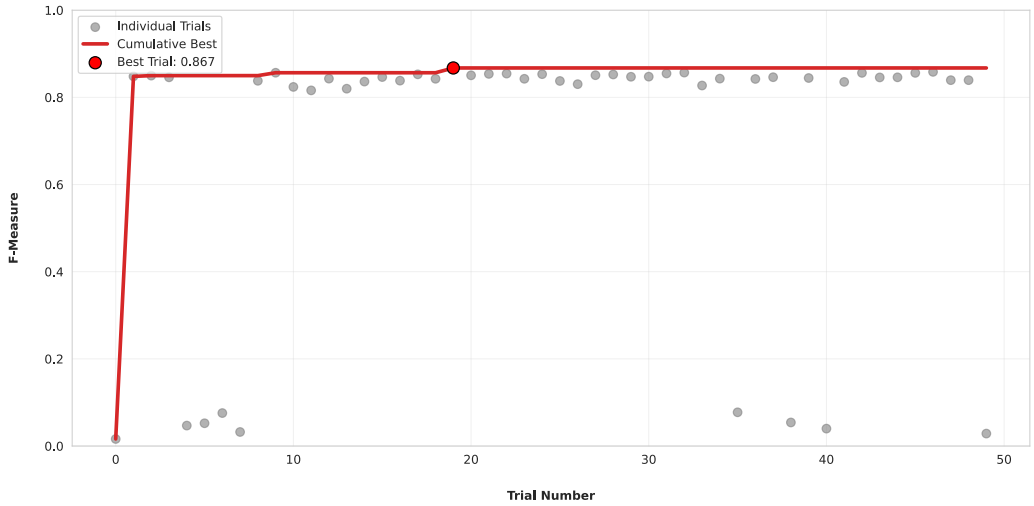


Figure 7. Optimization history of the ModernBERT model on the dataset D2.

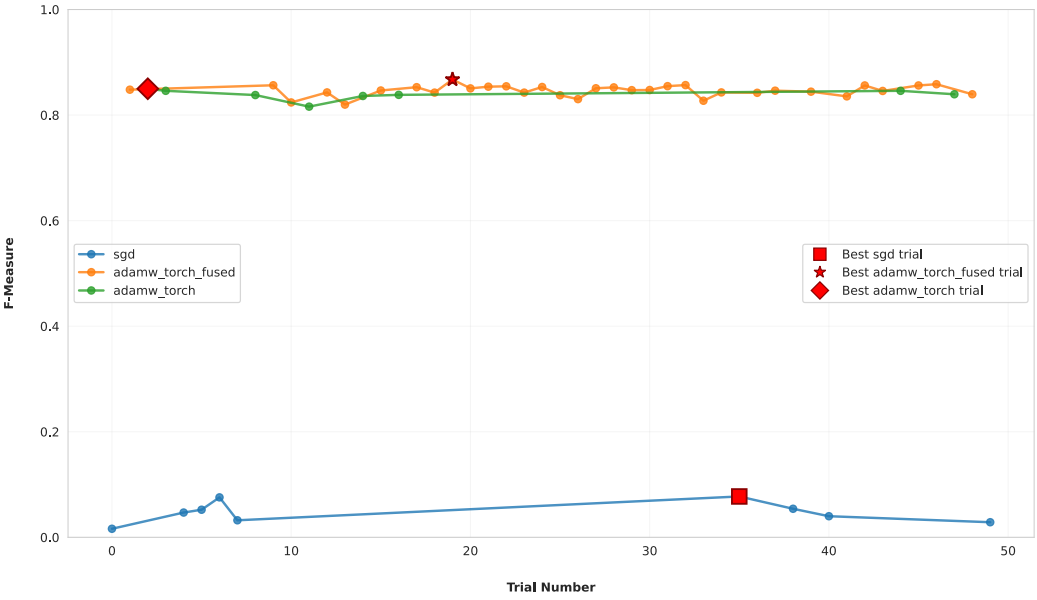


Figure 8. Optimization results per trial of each optimizer of the ModernBERT model on the dataset D2.

3.2.2. Fine-Tuning Results

After the thorough hyperparameter tuning, the models went through a fine-tuning process, i.e., their pre-trained weights were adjusted using the optimal configurations identified for each dataset to enhance their performance further. Table 6 presents these optimal hyperparameters identified in the previous step for the best two-performing models for each dataset. The fine-tuned models underwent evaluation using the test set to check their ability to handle unseen data. Overall, the outcomes confirmed the effectiveness of the previous process, with fine-tuned models exhibiting high results across key metrics

on both datasets. Table 7 summarizes the performance metrics achieved by the fine-tuned BERT on the D1, showing an F-Measure score of 0.89, which indicates a strong balance between the identification of key entities and the identification accuracy of such entities. This result is also supported by a precision of 0.89 and a recall of 0.90, showing that the model is able to identify the access control policy components. Additionally, we analyzed the performance on a per-entity basis, and as depicted in Table 8 and Figure 9, the model achieved an F-measure of 0.96 for identifying the "action" entity. It also demonstrated optimal performance for the "subject" entity, achieving an F-Measure of 0.91. The score for the "resource" entity was lower at 0.81, but that level appears quite solid. We also reported the averages of this entity's analysis, with an overall average regarding the F-Measure of 0.89, which demonstrates strong model performances despite the class imbalance.

Table 6. Fine-tuning hyperparameters.

Hyperparameter	BERT (D1)	ModernBERT (D2)
Learning rate	2.561×10^{-5}	4.198×10^{-5}
Epochs	9	9
Weight decay	0.136	0.237
Batch size	8	8
Label smoothing factor	0.052	0.138
Dropout rate	0.339	0.159
Optimizer	adamw_torch	adamw_torch_fused

Table 7. Results of BERT fine-tuning on dataset D1.

Metric	BERT (D1)
Precision	0.890
Recall	0.902
F-Measure	0.896

Table 8. Results of the per-entity analysis of the fine-tuned BERT on D1.

Entity	Precision	Recall	F-Measure
ACTION	0.955	0.972	0.964
RESOURCE	0.806	0.821	0.813
SUBJECT	0.912	0.912	0.912
Micro Avg	0.890	0.902	0.896
Macro Avg	0.891	0.902	0.896
Weighted Avg	0.890	0.902	0.896

ModernBERT fine-tuned on the D2 also demonstrated a similar level of efficacy, achieving an overall F-Measure value of 0.85 (Table 9). This indicates that the models effectively identify access control components in the second dataset, despite potential complexities due to the identification of five distinct entities. A further breakdown of its performance revealed that the model, with a gap between recall (0.98) and precision (0.73), learned to include more entities, even if not entirely relevant, rather than excluding important ones, but at the cost of a higher number of false positives. Instead, a detailed analysis of the per-entity of D2 is provided in Table 10 and Figure 10. They show that ModernBERT is able to identify the majority classes, specifically the "action" and "subject" entities, supported by the F-Measure values of 0.95 and 0.93, respectively. Conversely, the "condition" entity, likely attributable to the limited availability of training instances, presents an F-Measure of 0.32. Despite facing the same challenges, with fewer represented

elements in the training set, the "purpose" entities yield an F-Measure value of 0.84. For the average results, the macro average (which treats all classes equally) presents a value of 0.76; however, it is impacted by the poor performance with the "condition" entities. Comparatively, weighted and micro averages F-Measure scores are both 0.86, supported by the higher performance of the majority classes.

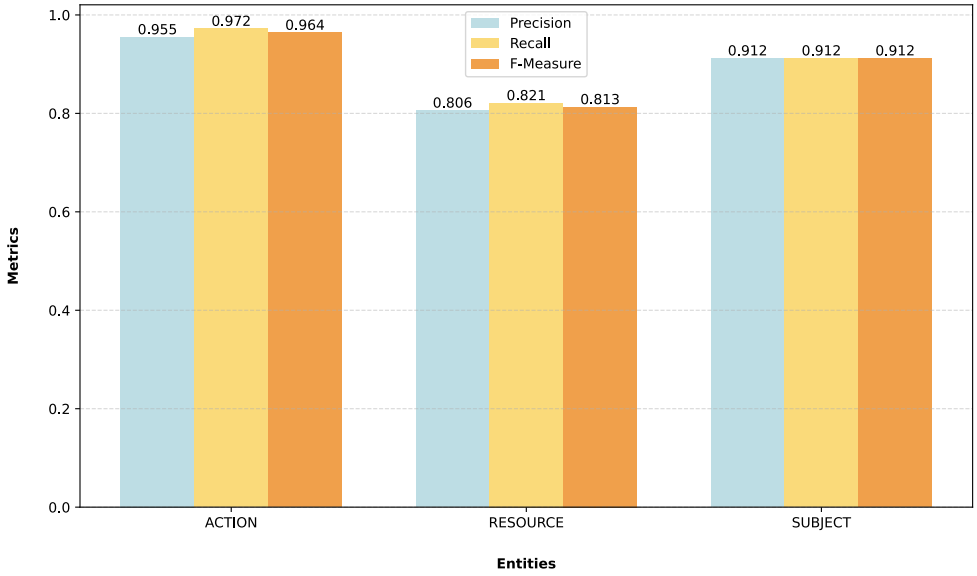


Figure 9. Results obtained by BERT for each entity present in D1.

Table 9. Results of ModernBERT fine-tuning on dataset D2.

Metric	ModernBERT (D2)
Precision	0.734
Recall	0.986
F-Measure	0.842

Table 10. Results of the per-entity analysis of the fine-tuned ModernBERT on D2.

Entity	Precision	Recall	F-Measure
ACTION	0.942	0.963	0.953
RESOURCE	0.733	0.783	0.757
SUBJECT	0.932	0.941	0.937
CONDITION	0.333	0.308	0.320
PURPOSE	0.917	0.786	0.846
Micro Avg	0.840	0.864	0.852
Macro Avg	0.771	0.756	0.762
Weighted Avg	0.841	0.864	0.852

Per-Dataset Analysis

Given the results obtained in the previous Section, we further examined the differences between the "condition" and "purpose" entities for the D2. We performed a granular analysis to understand the underlying factors contributing to their differential performance, particularly considering that the number of these two entities within the dataset is almost similar, as emerged in Section 3.1 (102 elements for the "condition" and 127 for the "purpose" in training). To investigate the significant performance disparity, we performed a disaggregated analysis on the test set by reporting the F-Measure for each entity calculated independently for each of the five source datasets. The results in Table 11 reveal

that the model presents a stable and high performance across all data sources where it is present. ModernBERT is able to perfectly identify the "purpose" entity in iTrust T2P, while it achieved a strong score (0.71) on iTrust Acre. This consistency suggests the model generalizes well for this entity across varied origins.

In contrast, the "condition" entity exhibits considerable variability in performance across the different sources. The model collapsed to a zero score on the iTrust Acre dataset—a complete failure to recognize any instances within that specific data subset. At the same time, it achieved low scores on the other ones, particularly on the Collected ACP, which contained the most "condition" examples. This variability significantly underscores the difficulty of the model to identify this entity, and may suggest to us that the linguistic expression of the "condition" is challenging to recognize.

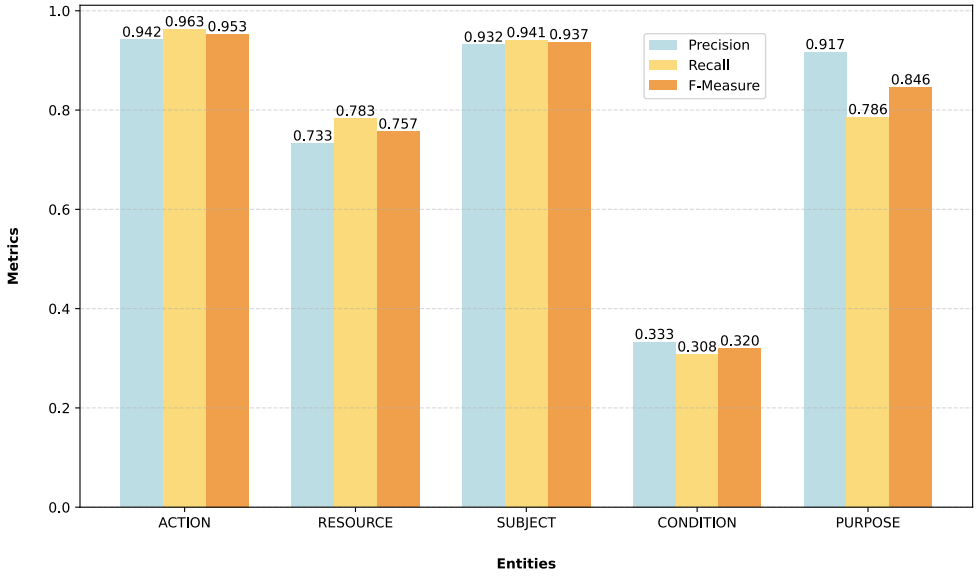


Figure 10. Results obtained by ModernBERT for each entity present in D2.

Table 11. Results of the per-dataset analysis of the fine-tuned ModernBERT on D2.

Source Dataset	Condition (F-Measure)	No. of Elements	Purpose (F-Measure)	No. of Elements
iTrust Acre	0.000	2	0.714	8
iTrust T2P	–	–	1.000	5
Collected ACP	0.571	5	1.000	1
CyberChair	0.400	2	–	–
IBM	0.200	4	–	–
Total elements		13		14

3.2.3. Explainability Results

This Section delves into the model's prediction interpretability, providing insight into the textual characteristics that meaningfully affected the decision-making process while extracting NLACP components. For BERT, we examined a sample case from the test set that was accurately classified (as shown in Listing 2), investigating how token-wise attention mechanisms facilitated the identification of subjects, actions, and resources inherent in access control policies. Figure 11a reports the matrix attribution score, where the model appropriately identified "*registrar* " as the subject. Across the distributed scores, the verb "enter" emerged as the most influential one (0.404). For the action entity, the highest score was attributed to the word itself; however, BERT also paid significant attention to tokens,

such as "*registrar*" and "*the*", indicating that it learned the subject-action structure ("*the registrar enters...*"). Finally, the resource in this sentence is the compound word "student id." For the former, the highest score is attributed to itself and to the action "*enter*." Meanwhile, the token "*id*" relies on itself and on the previous token ("*student*"). This result shows us that the model has identified "*id*" as a continuation of the entity "*student*."

Figure 11b, instead, shows an analysis of a misclassified instance (Listing 3), where the model correctly identified the multi-token "lab technician" as the subject of the NLACP. Additionally, it also classified the term "update" as the action, with the greater influence of the previous token "can". However, BERT struggled to distinguish a slightly more complex resource, resulting in incomplete extraction. As can be seen, "*status*" was identified as the beginning of the resource, relying mainly on the preceding compound verb ("*can update*"). Starting with the next token, classified as outside the previous entity ("*of*"), a cascading failure occurred. This was primarily due to the negative influence of "*status*," thus interrupting the entity's identification. Once the entity chain is interrupted, the final token "*procedure*" also received an incorrect classification, primarily because the attribution score of the preceding tokens "*status of*" are both negative.

Listing 2. Correctly classified sample from the D1 test set used for BERT explainability.

```
Original (tokens and labels):
the [O] system [O] requests [O] that [O] the [O] registrar [B-SUBJECT] enter
    [B-ACTION] the [O] student [B-RESOURCE] id [I-RESOURCE]

Predicted:
the [O] system [O] requests [O] that [O] the [O] registrar [B-SUBJECT] enter
    [B-ACTION] the [O] student [B-RESOURCE] id [I-RESOURCE]
```

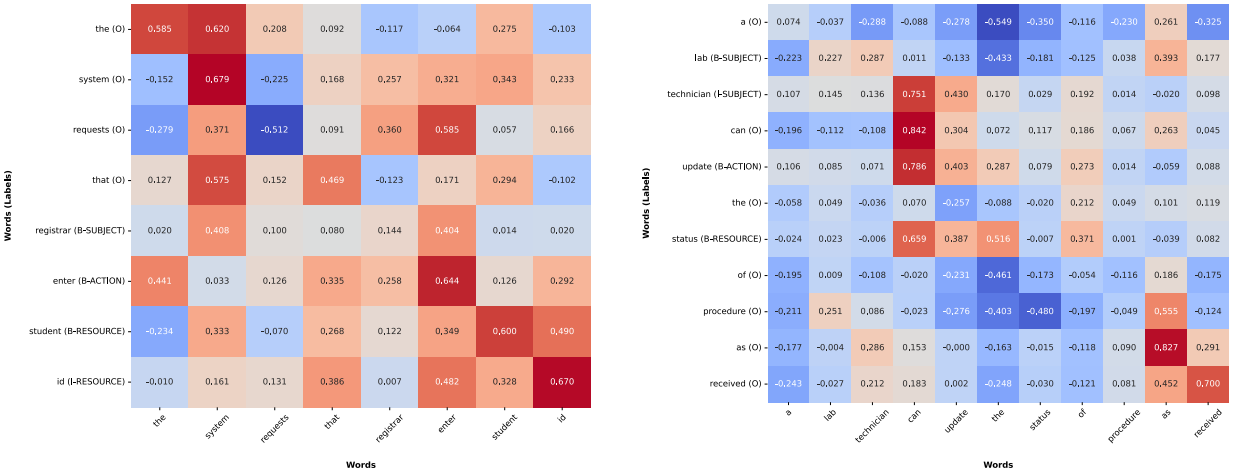
Listing 3. Misclassified sample from the D1 test set used for BERT explainability.

```
Original:
  a [O] lab [B-SUBJECT] technician [I-SUBJECT] can [O] update [B-ACTION] the [O]
      status [B-RESOURCE] of [I-RESOURCE] a [I-RESOURCE] lab [I-RESOURCE]
      procedure [I-RESOURCE] as [O] received [O]

Predicted:
  a [O] lab [B-SUBJECT] technician [I-SUBJECT] can [O] update [B-ACTION] the [O]
      status [B-RESOURCE] of [O] a [O] lab [O] procedure [O] as [O] received [O]
```

We also performed the same analysis (Listing 4) for ModernBERT fine-tuned with D2. Figure 12a illustrates a correctly classified instance from the test set of D2, where the model recognized the subject, the action, the resource, and the purpose. To handle out-of-vocabulary words, these models often employ subword tokenization strategies, breaking down words into meaningful subunits while maintaining intact frequent ones, and breaking down rare words into more common ones. In this case, we can observe a typical example of this process for the "*lhcp*" subject, where the model segments the acronym into its constituent subwords, yet correctly predicts them as beginning and within the subject entity, despite the initial labeling scenario considering it as a single entity. This specific sentence, randomly sampled from the test set, belongs to the subset of iTrust T2P, where (as shown in Table 11 and Listing 5) the model perfectly performed on the purpose entities. In this case, it is possible to observe that the duo "*view details*" was predicted as such. Specifically, for the "*view*" token, ModernBERT relied on the preceding adverb "*then*".

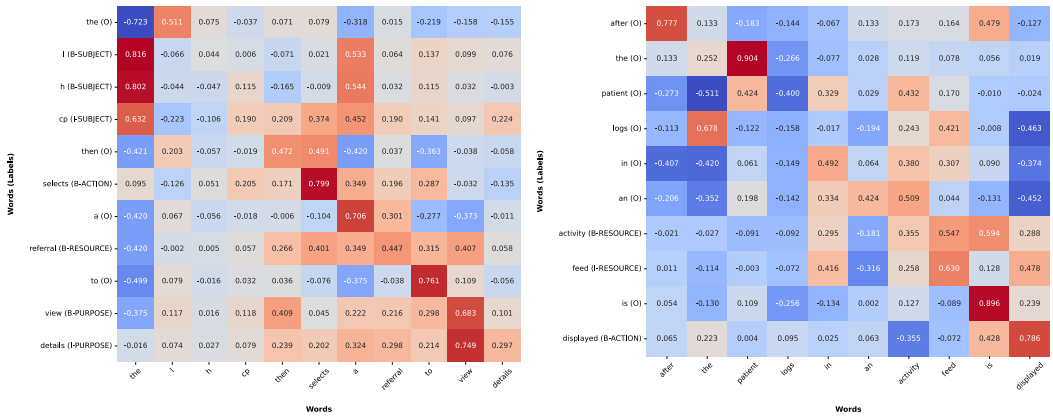
While for "*details*" the single most influential one was "*view*", demonstrating that the model treats him as a direct continuation of the previous token.



(**a**) (**b**)

Figure 11. (**a**) Matrix attribution score for a correctly classified test sample using BERT, illustrating the contribution of each input token to the model's final classification decision. (**b**) Matrix attribution score for a misclassified test sample using BERT, illustrating the contribution of each input token to the model's final classification decision.

As emerged in the previous Section, ModernBERT struggled to consistently identify the condition entity across diverse datasets. The case shown in Figure 12b exemplifies this challenge, revealing a misclassified instance, from the iTrust Acre subset, where the model failed to accurately delineate the condition element within an NLACP. In this case, the condition statement "*after the patient logs in*" was predicted completely wrong, starting from the "*after*" token, which presents a high self-attribution score, considering it as an outside entity. The following most contributing score comes from the verb "*is*", which is entirely unrelated. The cascading failure process, initiated by this initial misclassification, is confirmed by the "*logs*" and "*in*" tokens, where the attribution scores are a mix of positive and negative influences from surrounding words.



(**a**) (**b**)

Figure 12. (**a**) Matrix attribution score for a correctly classified test sample using ModernBERT, highlighting how each input token contributes to the model's final classification decision. (**b**) Matrix attribution score for a misclassified test sample using ModernBERT, highlighting how each input token contributes to the model's final classification decision.

Listing 4. Correctly classified sample from the D2 test set used for ModernBERT explainability.

```
Original (tokens and labels):
the [O] lhcp [B-SUBJECT] then [O] selects [B-ACTION] a [O] referral [B-RESOURCE]
    to [O] view [B-PURPOSE] details [I-PURPOSE]

Prediction:
the [O] l [B-SUBJECT] h [I-SUBJECT] cp [I-SUBJECT] then [O] selects [B-ACTION] a
    [O] referral [B-RESOURCE] to [O] view [B-PURPOSE] details [I-PURPOSE]
```

Listing 5. Misclassified sample from the D2 test set used for ModernBERT explainability.

```
Original (tokens and labels):
after [B-CONDITION] the [I-CONDITION] patient [I-CONDITION] logs [I-CONDITION] in
     [I-CONDITION] an [O] activity [B-RESOURCE] feed [I-RESOURCE] is [O]
    displayed [B-ACTION]

Prediction:
after [O] the [O] patient [O] logs [O] in [O] an [O] activity [B-RESOURCE] feed
    [I-RESOURCE] is [O] displayed [B-ACTION]
```

3.2.4. Discussion

In this study, we conducted a multi-stage evaluation of several transformer-based models designed for extracting elements of access control policies from NLACPs. The initial hyperparameter tuning phase showed that the choice of optimizer was the strongest determining factor in model performance. A further investigation found that this factor significantly increased the F-Measure (the optimization measure) to 0.85 in the first experiments, in accordance with the AdamW optimizer preference over Stochastic Gradient Descent (SGD). This evidence is further supported by the investigation of later experiments where the use of models in tandem with SGD failed to produce a significant value for the F-Measure.

For the fine-tuning phase, the BERT model on the D1 (subject, action, and resource) achieved a significant F-Measure of 0.89, excelling in finding the action entity (0.96) and the subject (0.91), albeit with some limitations in the extraction of the resources (0.81). On the other hand, ModernBERT was used to handle the more complex 5-entity extraction on the D2 dataset, achieving an overall F-Measure score of 0.84. The further per-entity analysis showed a discernible performance difference, particularly in identifying the "condition" entities with an F-Measure of 0.32. Despite the purpose and condition presenting approximately the same number of instances in the dataset, their results are inconsistent; however, the per-dataset analysis revealed that the difficulty in identifying this entity stems not only from data scarcity but can also be attributable to the linguistic complexity across different sources.

The explainability stage confirmed the performance and limitations observed in the qualitative analysis. By studying a misclassified sample of BERT on D1, we found that it struggled with the multi-token resource entity ("*status of a lab procedure*"); in fact, the preposition "*of*" acted as a sort of chain-breaker. While ModernBERT on a D2 sample completely missed the condition ("*after the patient logs in*"), in this case, it failed from the first word. This error may have occurred because the conditional clause began with a common word. Furthermore, in the expanded context window, the word "*patient*" received the most influence from the previous word, and when combined ("*the patient*"), they also form a common non-conditional clause. A possible reason behind this model error is that

this policy does not use a classic form to express a condition (e.g., "*if*") that must be verified, but rather a set of common words.

3.2.5. Challenges in Operationalizing Policy Within Formal Frameworks

Once identified, these components can be translated into a formal, structured policy language by performing a process known as operationalization. However, the direct translation is a challenging task due to the complex and verbose structure of these policy languages, which often include intricate syntax, conditional logic, and hierarchical dependencies. Let us consider an example policy from the iTrust Acre dataset, as in Box 1. In this case, the extracted condition "*if answer to security question is correct*" maps to the block condition as in Listing 6, but even if this entity is correctly identified, there is a significant semantic gap that prevents its direct translation into the formal language. Nevertheless, the proposed method could be integrated with a template-based approach that enables the creation of a structured output aligned with formal requirements. Returning to the previous example, once the condition in the sentence is identified, it could be mapped into a conditional policy template.

Box 1. iTrust Acre policy example.

if answer to security question is correct, allow user to change their password

Listing 6. Example of XAMCL statement translation of a condition for an access control policy.

```
<Condition>
        <Apply FunctionId="boolean-equal">
                <Attribute Reference="urn:oasis:names:tc:xacml:1.0:subject:
security-answer-correct" DataType="boolean"/>
                <AttributeValue DataType="boolean">true</AttributeValue>
        </Apply>
</Condition>
```

4. Related Work

In [13], the authors presented SPARCLE, a tool for authoring and parsing privacy policies that utilizes natural language parsing technology to identify policy elements, enabling users to create, modify, and visualize policies. It includes a set of grammars that can parse natural language privacy policy rules and identify policy elements. Xiao et al. [11] proposed Text2Policy, a tool that automatically extracts access control policies from natural language software documents, addressing the challenge of manual extraction. For ACP rules, authors identified the subject, action, and resource elements based on matched semantic patterns and inferred the policy effect by checking for negative expressions. The system employs shallow parsing to annotate sentences with phrases, clauses, and grammatical functions, utilizing a domain-specific dictionary that facilitates the association of verbs with predefined semantic classes, thereby aiding in the identification of the correct action and its corresponding semantic meaning. Authors in [12] created a tool called ACRE (Access Control Rule Extraction), which utilizes NLP techniques to extract access control rules from natural language documents automatically. Once they determine that a sentence contains an ACR, they extract the subject, action, and resource elements from the sentence representation using a relation extraction algorithm. The process begins with a small, well-known set of ACR patterns and then expands those patterns to identify other, closely related patterns. The algorithm iterates until no new items or patterns are discovered. After that, they train the naïve Bayes classifier by using the identified patterns in the other documents, achieving a maximum F1-score of 98%. The paper [17] proposes a technique to identify access control policy sentences from natural language policy documents. The

study employs a combination of NLP and machine learning techniques to extract sentences that exhibit ACP content and to identify various types of features, including dependency relations. To identify the ACP sentences, they utilized inherent relations that occur in these types, such as subjects, objects, and verbs. Researchers in [18] introduced semantic role labeling as a method to identify ACPs within unrestricted natural language documents, aiming to automate the process of translating security policies into machine-readable formats. After the ACP identification, they used a semantic parser to identify the predicate-argument structure. It involves detecting semantic arguments associated with a verb (or predicate) and classifying them into specific roles (e.g., who did what to whom). The proposed SRL-based approach achieved an average F1-score of 75% for correctly identifying ACP elements. Alohaly et al. [19] proposed a two-phase framework to extract Attribute-Based Access Control (ABAC) constraints from unstructured natural language access control policies using a BiLSTM model. They treated the constraints identification phase as a sequence labeling task after labeling the presence of constraints in NLACPs using the BIO (Beginning, Inside, Outside) format. The study achieved an F1 score of 0.91 in detecting at least 75% of each constraint expression. Authors in [20] propose a sequence labeling model based on the hybrid neural network RoBERTa-BiLSTM-CRF to mine content attributes from unstructured text for attribute-based access control (ABAC) in a big data environment. Specifically, they proposed a model comprising three modules: word embedding, context encoding, and inference, and utilized the RoBERTa pre-training model for dynamic word embeddings. The authors tested the approach on the publicly available CLUENER2020 data set, achieving an F1-measure of 80%. In [11], the authors propose an automated approach to extract ABAC policies from natural-language documents in healthcare systems, utilizing BERT and semantic role labeling to identify access control policy sentences and extract rules. Once they identified the sentences, they used the semantic role labeling sub-module to label the subject-values, subject attribute-values, object-values, and object attribute-values. Unlike other works, Heaps et al. [21] used an automated approach to extracting access control information from user stories. After identifying statements containing access control information, they performed a named entity recognition task to predict whether a word in a user story represents an "actor" (or end user), a "data object", or an "operation". The approach achieved an F1-score of 79.8% for the entity recognition task. Instead, in [22], authors developed an algorithm for mining ABAC policies from operation logs and attribute data, which can handle incomplete information about entitlements and produce policies with over-assignments. The results obtained show that the mined policy is sufficiently similar to the original ABAC policy, making it a suitable starting point for policy administrators. In [23] proposed a multi-objective evolutionary approach for inferring attribute-based access control policies from logs of authorized and denied requests. The approach incorporates a domain-specific phenotypic representation, custom genetic operators, and an incremental strategy for learning a policy in an incremental manner.

Unlike prior works that rely on grammar-based parsing, shallow semantic patterns, or traditional machine learning classifiers, our approach leverages a state-of-the-art, end-to-end transformer architecture, eliminating the need for handcrafted rules. Additionally, while other studies have employed deep learning models such as BiLSTMs or hybrid architectures, our work offers a more comprehensive analysis of the model's entire lifecycle, from hyperparameter optimization to fine-tuning analysis. Finally, the most significant contribution of our work lies in the application of state-of-the-art explainability techniques to diagnose model failures.

5. Conclusions and Future Work

In this research, we evaluated several transformer-based models for the automated extraction of key components from access control policies defined in natural language. To perform our analysis, we relied on a dataset comprising various sources of requirements specifications from different sectors, such as healthcare and conference management systems. Our evaluation is based on two different versions, covering an extraction level of 3 entities and another of 5 entities, including subject, action, resource, condition, and purpose entities. The investigation was conducted in several stages, starting with a hyperparameter optimization phase. This first phase was crucial for optimizing the model's performance, as the selection of the optimizer had a significant impact on the result, with those based on AdamW outperforming Stochastic Gradient Descent. We then conducted a fine-tuning phase, during which the BERT model demonstrated excellent performance in extracting action and subject entities in a 3-entity extraction paradigm, achieving an F-Measure value of 0.89. In the more complex 5-entity extraction task, ModernBERT proved to be the most promising model, with an overall F-Measure of 0.84, albeit with significant variability in some individual components, such as condition and purpose. Finally, we introduced a state-of-the-art explainability step using layer-wise integrated gradients to gain insight into the decision-making process of these deep models. The goal of this phase was to delve deeper into the attribution scores of individual tokens, particularly for cases of misclassification that emerged during the qualitative analysis. However, this attribution method reveals which tokens are considered critical for a given prediction, but in order to provide a deeper analysis for this task, it is also important to understand why the model misclassified a specific entity. For this reason, we plan to integrate perturbation-based methods, where we will modify or mask the first misclassified token and record the subsequent prediction, to understand how these changes affect the model. Also, in future work, we plan to explore the capabilities of larger and more sophisticated models, such as the Llama and Mistral series. Our hypothesis is that a larger number of parameters and extended contextual windows could improve the identification of complex dependencies, which, as we observed in this study, pose a challenge in terms of conditions and purposes. Furthermore, using few-shot learning techniques with these large language models could help generalize from limited examples and classify rare entities more accurately. Another critical area for future research will involve a data augmentation strategy to enrich the training datasets, particularly for the proposed D2, where the models exhibit performance degradation for rare entities, such as conditions and purposes. This phase could involve a synthetic data generation task to enhance policies with these entities while preserving both the semantic and syntactic forms of the sentences.

Author Contributions: Conceptualization, L.P., F.M. (Fabio Martinelli), A.S. and F.M. (Francesco Mercaldo); Methodology, L.P., F.M. (Fabio Martinelli), A.S. and F.M. (Francesco Mercaldo); Software, L.P.; Validation, L.P. and F.M. (Francesco Mercaldo); Investigation, L.P., F.M. (Fabio Martinelli), A.S. and F.M. (Francesco Mercaldo); Resources, F.M. (Fabio Martinelli), A.S. and F.M. (Francesco Mercaldo); Data curation, L.P.; Writing—original draft, L.P. and F.M. (Francesco Mercaldo); Writing—review & editing, L.P., F.M. (Fabio Martinelli), A.S. and F.M. (Francesco Mercaldo); Supervision, F.M. (Fabio Martinelli), A.S. and F.M. (Francesco Mercaldo); Funding acquisition, F.M. (Fabio Martinelli). All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially supported by EU DUCA, EU CyberSecPro, SYNAPSE, PTR 22-24 P2.01 (Cybersecurity) and SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the EU - NextGenerationEU projects, by MUR—REASONING: foRmal mEthods for computAtional analySis for diagnOsis and progNosis in imagING—PRIN, e-DAI (Digital ecosystem for integrated analysis of heterogeneous health data related to high-impact diseases: innovative model of care and research), Health Operational Plan, FSC 2014–2020, PRIN-MUR-Ministry of Health,

Progetto MolisCTe, Ministero delle Imprese e del Made in Italy, Italy, CUP: D33B22000060001, FORESEEN: FORmal mEthodS for attack dEtEction in autonomous driviNg systems CUP N.P2022WYAEW and ALOHA: a framework for monitoring the physical and psychological health status of the Worker through Object detection and federated machine learning, Call for Collaborative Research BRiC—2024, INAIL.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

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

References

1. Cheng, Y.; Xu, M.; Zhang, Y.; Li, K.; Wu, H.; Zhang, Y.; Guo, S.; Qiu, W.; Yu, D.; Cheng, X. Say What You Mean: Natural Language Access Control with Large Language Models for Internet of Things. *arXiv* **2025**, arXiv:2505.23835. [CrossRef]
2. Tang, A. Implementation of Formal Semantics and the Potential of Non-Classical Logic Systems for the Enhancement of Access Control Models: A Literature Review. *arXiv* **2023**, arXiv:2308.12983. [CrossRef]
3. Mondragon, J.; Rubio-Medrano, C.; Cruz, G.; Shastri, D. "I Apologize For Not Understanding Your Policy": Exploring the Specification and Evaluation of User-Managed Access Control Policies by AI Virtual Assistants. *arXiv* **2025**, arXiv:2505.07759.
4. Bella, G.; Castiglione, G.; Santamaria, D.F. An automated method for the ontological representation of security directives. *arXiv* **2023**, arXiv:2307.01211. [CrossRef]
5. Fatema, K.; Debruyne, C.; Lewis, D.; OSullivan, D.; Morrison, J.P.; Mazed, A.A. A semi-automated methodology for extracting access control rules from the european data protection directive. In Proceedings of the 2016 IEEE Security and Privacy Workshops (SPW), San Jose, CA, USA, 22–26 May 2016; pp. 25–32.
6. Gillioz, A.; Casas, J.; Mugellini, E.; Abou Khaled, O. Overview of the Transformer-based Models for NLP Tasks. In Proceedings of the 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 6–9 September 2020; pp. 179–183.
7. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631.
8. Zhang, Z.; Al Hamadi, H.; Damiani, E.; Yeun, C.Y.; Taher, F. Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access* **2022**, *10*, 93104–93139. [CrossRef]
9. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic attribution for deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 3319–3328.
10. Petrillo, L.; Martinelli, F.; Santone, A.; Mercaldo, F. TALLM: A framework for Text Analysis using Large Language Models. In Proceedings of the 2025 IEEE 13th International Conference on Healthcare Informatics (ICHI), Rende, Italy, 18–21 June 2025; pp. 663–664.
11. Xiao, X.; Paradkar, A.; Thummalapenta, S.; Xie, T. Automated extraction of security policies from natural-language software documents. In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, Cary, NC, USA, 11–16 November 2012; pp. 1–11.
12. Slankas, J.; Xiao, X.; Williams, L.; Xie, T. Relation extraction for inferring access control rules from natural language artifacts. In Proceedings of the 30th Annual Computer Security Applications Conference, New Orleans, LA, USA, 8–12 December 2014; pp. 366–375.
13. Brodie, C.A.; Karat, C.M.; Karat, J. An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In Proceedings of the Second Symposium on Usable Privacy and Security, Pittsburgh, PA, USA, 12–14 July 2006; pp. 8–19.
14. Jayasundara, S.H.; Arachchilage, N.A.G.; Russello, G. Ragent: Retrieval-based access control policy generation. *arXiv* **2024**, arXiv:2409.07489. [CrossRef]
15. Palmer, M.; Gildea, D.; Kingsbury, P. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.* **2005**, *31*, 71–106. [CrossRef]
16. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
17. Narouei, M.; Khanpour, H.; Takabi, H. Identification of access control policy sentences from natural language policy documents. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy, Philadelphia, PA, USA, 19–21 July 2017; pp. 82–100.
18. Narouei, M.; Takabi, H.; Nielsen, R. Automatic extraction of access control policies from natural language documents. *IEEE Trans. Dependable Secur. Comput.* **2018**, *17*, 506–517. [CrossRef]

19. Alohaly, M.; Takabi, H.; Blanco, E. Towards an automated extraction of abac constraints from natural language policies. In Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection, Charleston, SC, USA, 15–17 July 2019; pp. 105–119.
20. Zhu, Z.; Ren, Z.; Du, X. Unstructured text ABAC attribute mining technology based on deep learning. In Proceedings of the 2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST), Guangzhou, China, 10–12 December 2021; pp. 34–39.
21. Heaps, J.; Krishnan, R.; Huang, Y.; Niu, J.; Sandhu, R. Access control policy generation from user stories using machine learning. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy, Calgary, AB, Canada, 19–20 July 2021; pp. 171–188.
22. Xu, Z.; Stoller, S.D. Mining attribute-based access control policies from logs. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy, Vienna, Austria, 14–16 July 2014; pp. 276–291.
23. Medvet, E.; Bartoli, A.; Carminati, B.; Ferrari, E. Evolutionary inference of attribute-based access control policies. In Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization;, Guimarães, Portugal, 29 March–1 April 2015; pp. 351–365.