



**Project Title: Advanced Multi-Label Learning
Techniques (AMULET)**

Project ID: HFRI-FM17-514

Principal Investigator: Grigorios Tsoumakas

Host Institution: Aristotle University of Thessaloniki

Project Website: <https://amulet.csd.auth.gr>

**Deliverable D3.1 – Technical Report on Local
Multi-label Explanations**

Team AMULET
GRIGORIOS TSOUMAKAS
IOANNIS MOLLAS
NIKOLAOS MYLONAS



1 Introduction

The first task of the third work package concerns local multi-label explanations. Multi-label models are widely used both in the industry and academia, due to the number of domains they are applicable on. Explaining the decisions of such models however is not a straightforward task, mainly due to their complexity. Hate speech detection is one such domain, which affects online users on a daily basis. Explaining the decisions of models that aim to moderate hate speech on social media, can be a valuable tool. Applying this in multi-label context is even more challenging due to the lack of high quality data sets, to test our techniques on.

To alleviate this issue, we developed a multi-label hate speech detection data set called ETHOS. ETHOS retains a balance between the multiple labels, while also covering a wide spectrum of topics for each one of the hate speech labels. These two properties are very scarce in the literature, and as such we believe it is a valuable tool towards hate speech detection.

Furthermore, since the focus of this work package is interpretability of text classification tasks, a domain where transformer models are commonly employed, we studied how attention computed by transformers can be used as interpretation for the model’s decisions. To that end, we performed an investigation to discover the optimal interpretation extraction process from attention.

Finally, one last activity related to this task, was extending a recent work on local interpretability of Random Forest models, to be applicable in multi-label tasks. The extension retains the main properties of the original work, offering informative rules and a variety of representations regarding the labels that the explanation covers.

2 ETHOS: a Multi-label Hate Speech Detection Dataset

Hate speech (HS) is a type of derogatory public speech directed at specific individuals or groups of people based on characteristics such as race, religion, ethnic origin, national origin, gender, disability, sexual orientation, or gender identity¹. This phenomenon occurs verbally or physically (e.g., speech, text, gestures), fostering the formation of racism and ethnocentrism. Because of the social consequences associated with HS, many countries consider it an unlawful act, especially when violence or hatred is promoted [1]. Although freedom of speech and expression is an essential human right, it is in contradiction with laws that protect individuals from HS. As a result, almost every country has responded by developing matching regulatory frameworks, while the Data Mining and Machine Learning (ML) research community have lately conducted research related to techniques that attempt to remedy such occurrences, providing data sets and ML models [2].

¹https://en.wikipedia.org/wiki/Hate_speech

To overcome the key weaknesses of the existing data set collections of HS instances, we introduce a small, yet fairly, informative dataset, ETHOS, that does not suffer from issues such as imbalanced or biased labels (e.g., gender), produced appropriately following a carefully designed protocol. Considering the popular approaches of mining similar datasets for tackling with HS problem, we assume that an appropriate pre-process of initially collected data could improve in general their overall utilisation under ML or AI products, improving the total fitness of data quality, blending data mining techniques related with the field of Active Learning [3], such as query strategy and crowdsourcing platforms. The overview of the proposed annotation protocol is visualised through a flow chart in Figure 1. The finally obtained dataset is the outcome of a 3-stage process, which we describe shortly in the current Section.

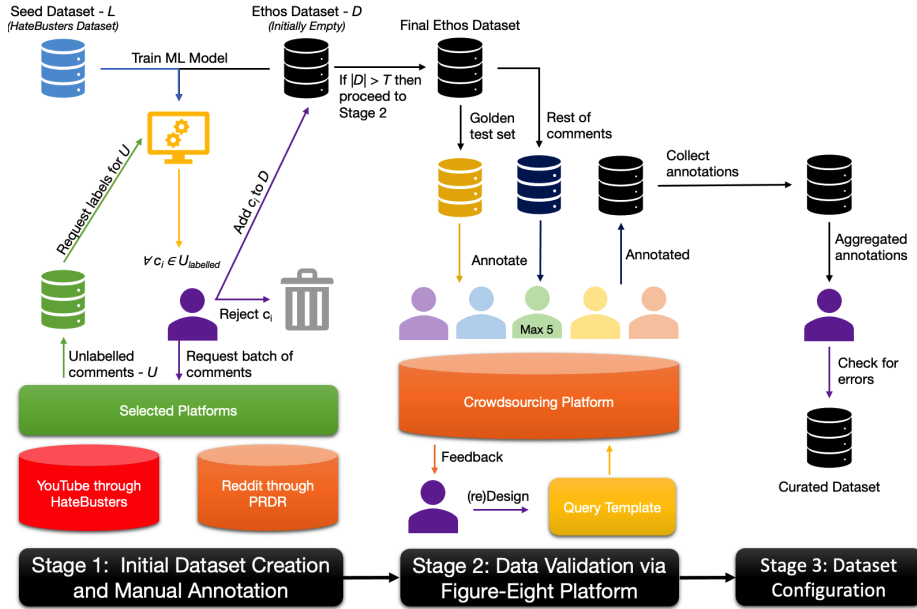


Figure 1: Dataset creation stages flowchart

2.1 Initial Dataset Creation and Manual Annotation

The first three procedures, mentioned as “Platform Selection & Data Collection”, “Data Prediction” and “Manual Data Annotation”, could be seen as the initial stage (Stage 1) which is executed until a stopping criterion is satisfied regarding the cardinality of the collected instances, based on the original available HS dataset which operates as the input. This stage works like a “stream”, specifically for groups of comments that we have already collected, annotating their weak labels’ predictions through a predefined ML classifier, before an active selection and manually annotation takes place over some unlabelled (U)

mined examples.

2.1.1 Platform Selection & Data Collection

To create this dataset (D), initially $D = \emptyset$, a data collection protocol has been designed. We chose the platforms of Hatebusters² and Reddit through the Public Reddit Data Repository³ to collect our data. Hatebusters Platform collects new data daily via the YouTube Data v3 API.

After these new data have been collected, the Hatebusters Platform performs the classification process. The locally retained pre-trained ML model predicts the class of each comment, exporting a ‘hate’ score. Currently, this model is a Support Vector Machine (SVM) model with a linear kernel embedded with the well-known vectorization technique of the term frequency-inverse document frequency (TF-IDF). Instead of transforming the output of the SVM learner to confidence score, we kept its inherent property to compute the distance from the decision boundary. Through this, lower time overheads and more faithful decisions are drawn.

After granting access to Hatebusters’ SQL database, based on the input data, this first part was to query the Hatebusters’ database for comments already annotated by the corresponding users, without spending any monetisation resources. These comments were deemed to be accurate, and they were the first group of comments to be manually annotated. The second part concerns the enrichment of the gathered comments, by querying Hatebusters’ database with a specific frequency (e.g. daily) for a time period – in our case this was equal to two months – with various queries. Based on the data obtained each previous day, the applied query strategy had been updated concerning only them. For example, when we received a sufficient amount for all categories of HS, except for one category, the queries in the Hatebusters’ database were updated to make comments specific to the residual category. Later on, we will show the categories and the amount of comments we have received.

The final part of the data collection process was based on a public Reddit data archive, which provides batches of files regarding Reddit comments on a monthly basis. The files of this directory were processed through a JSON crawler for selecting comments from specific subreddits for particular time periods. The discovery of subreddits incorporating different HS contents has been investigated^{4,5}, we distinguished the next entities:

- **Incels**, this subreddit became known as a place where men blamed women for their unintended celibacy, often promoting rape or other abuse. Those posts had a misogynistic and sometimes racist content.
- **TheRedPill**, which is devoted to the rights of men, containing misogynous material.

²<https://hatebusters.org>

³<https://files.pushshift.io/reddit/comments/>

⁴https://en.wikipedia.org/wiki/R/The_Donald

⁵<https://en.wikipedia.org/wiki/Incel>

- **The_Donald**, a subreddit where the participants create discussions and memes supportive of U.S. President Donald Trump. This channel has been described as hosting conspiracy theories and racist, misogynous, Islamophobic, and antisemitic content.
- **RoastMe**, in this subreddit, reddit users can ask their followers to ‘roast’ (insult) them.

While some of these subreddits were suspended and shut down by Reddit at the end of 2017 due to their context, it was possible to access comments from these subreddits by selecting files from the archive for October 2017 and earlier.

2.1.2 Data Prediction

The next process of Stage 1 is the “Data Prediction”. For each batch of comments extracted from the first part, the assignment of some useful labels to the available unlabelled set ($U^{current}$) is triggered through an ML model trained on an expanded version ($L \cup D$) of the Hatebusters’ dataset (L) and the new data annotated on Stage 3 (D). Per each iteration of the previous part, we were performing a grid search among a bunch of classification methods in the currently expanded dataset, obtaining the best algorithm through a typical 10-fold-CV process so as to be set as the annotator of the ($U^{current}$).

The selected bunch consisted of various ML models: SVMs, Random Forests (RF), Logistic Regression (LR), as well as simple or more complex architectures of Neural Networks (NNs). In addition to the classifier tuning, some TF-IDF vectorization techniques – with word or char n -grams (n from 1 to 13) – were also examined in this search.

2.1.3 Manual Data Annotation

By the end of the “Data Prediction” phase, the “Data Annotation” process is initiated. In the sense of active learning concept, a hybrid combination of query strategy has been employed in order to pick informative comments for manual annotation. The mentioned query strategy combines appropriately both concepts of Uncertainty Sampling and Maximum Relevance with predefined ranges of accepted confidence values based on the expected labels of the classifier we had trained [4]. More specifically, we were annotating the comments within the $[.4, .6]$ probability range, while we were examining few comments in the ranges $[.0, .1] \cup [.9, 1.0]$ to detect any major misclassification. Eventually, only comments with specific labels and content were added to the new dataset (D), preserving both the *balance of the labels* and the *diversity of the comments per label*. The latter asset stems directly from the existence of the human factor, since the class probabilities that are produced by any ML classifier just express its confidence independently of the underlying content. This kind of filtering is adequately addressed here by the human factor.

At the end of this process, if the number of comments collected is not more than a targeted threshold (T) – in our case $T = 1.000$ – we update the D , and

Stage 1 will be repeated to request new unlabelled comments. Otherwise, Stage 2 will be triggered. Despite the limited cardinality of the exported dataset, the adopted actively sampling process eliminates defects of redundancy, maintaining the both informativeness of each label, and reducing at the same time overfitting phenomena. Therefore, an in-depth evaluation stage regarding several learning models has been conducted in Section 4. The use of Query-by-Committee, another one popular active learning strategy, might insert practical difficulties in practice, and thus was not investigated in that analysis. The reason for this choice is twofold: independent classifiers are needed for properly formatting such a committee, which constitutes a hard task under the shortage of large amounts of data, while the corresponding stage of hyperparameter tuning would induce more computational overhead.

2.2 Data Validation via Figure-Eight Platform

The second stage will begin when T – in our case 1.000 – comments have been collected. Moreover, Hatebusters’ dataset is discarded, since it does not further contribute to our protocol. After a number of different experiments on the Figure-Eight platform, we settled on the next process. Firstly, given a specific comment, we ask the contributors to identify whether that comment *contains HS or not*. In a positive scenario, we raise 3 more questions: whether the comment *incites violence*, defining violence as “the use of physical force to injure, abuse, damage, or destroy”, and whether the comment includes *directed* or *generalized* HS. The case of targeting a single person or a small group of people is defined as directed HS, whereas the case of targeting a class/large group of people is described as generalised HS. Finally, we ask the contributors to pick *one* or *more* from the following *HS categories*, which, according to their opinion, better reflect(s) the content of the comments. The categories of HS concern gender, race, national origin, disability, religion and sexual orientation.

After testing the platform with 40 questions, we executed the task for the whole D , collecting in total 5.360 judgements. Almost every comment was therefore annotated by five different annotators. The level of expertise of the annotators was the 3rd, on a scale of 3 levels. “The 3rd level annotators are the smallest group of the most experienced, most accurate, contributors” according to the Figure-Eight System. We also computed the Fleiss’ kappa, a statistical measure for assessing the reliability of agreement of annotators, and we present the results in Table 1. A kappa value greater than 0.75 implies good agreement, while kappa values greater than 0.90 indicate perfect agreement [5].

	Contains Hate Speech	Violence	Directed vs Generalized	Gender	Race	National Origin	Disability	Sexual Orientation	Religion
Fleiss’ Kappa	0.814	0.865	0.854	0.904	0.931	0.917	0.977	0.954	0.963

Table 1: Reliability of annotators agreement per label

2.3 Dataset Configuration

The final stage regards dataset configuration. Taking as input the results from the Stage 2, the dataset takes its final form. Examining the annotated data one last time manually, we checked for any misclassification. Few errors occurred on some of the most disambiguous examples, assuring us about the quality of the annotators that participated. The use of representative test questions that follow a more realistic label distribution than the uniform could be useful to the overall process. This might be improved further by incorporating an interactive procedure that alerts annotators to mislabelled samples and/or allows them to provide feedback when they disagree.

Despite the inherent uncertainties introduced by the human factor, crowd-sourcing is the sole viable technique for gathering the required information regarding the label space. Furthermore, given the semantic overlap of label space encountered during HS detection, the assumption of obtaining cheap labels is violated. Given the idiomatic expressions and highly unstructured nature of the comments posted on social media platforms, this becomes especially clear when examined in a multi-label fashion. To address this, additional human supervision, as stated at this stage, is required, while the active sampling process, which aims to create a balanced dataset, is clearly justified.

2.4 ETHOS Dataset Overview

Two datasets⁶ were the product of the above operation. “ETHOS_Binary.csv”, the first one, includes 998 comments and a label on the presence or absence of hate speech content (*isHate*). The second file, called “ETHOS_Multi_Label.csv”, includes 433 hate speech messages along with the following 8 labels: (*violence*, *directed_vs_generalized*, *gender*, *race*, *national_origin*, *disability*, *sexual_orientation*, *religion*).

For every comment c_i , N_i annotators voted for the labels that we set. The label *isHate* was the result of summing up the positive votes $P_{1,i}$ of the contributors, divided by N_i , so its values are within the range of $[0, 1]$. We measured the *violence* label by summarising the positive votes of the contributors $P_{2,i}$ to the question: “Does this comment incite violence?”, which was divided by $P_{1,i}$ to be normalised to $[0, 1]$. Likewise, the value of the label *directed_vs_generalized* was determined by summarising the annotators replied *directed* $P_{3,i}$ to the question, “Is this comment targeting a specific individual (directed) or a group/class of people (generalized)?”, divided by $P_{1,i}$. Finally, we accumulated the votes of the $P_{1,i}$ contributors for each of the 6 hate speech categories, and dividing them by $P_{1,i}$, we obtained six independent labels.

This dataset achieves to create balanced labels. In particular, it maintains balance between the two classes of *isHate* label, almost perfect balance between the 6 labels of hate speech categories, while it has a fair ratio between the rest of the labels (Figure 2). In Table 2, the balance between hate speech categories

⁶<https://github.com/intelligence-csd-auth-gr/ETHOS-Hate-Speech-Dataset.git>

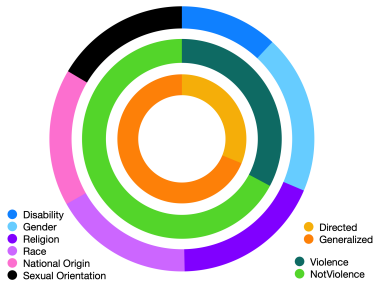


Figure 2: Ratio of labels

	V-D	nV-D	V-G	nV-G	Sum
Gender	14	22	13	37	86
Race	4	13	12	47	76
National Origin	5	11	18	40	74
Disability	12	15	8	18	53
Religion	11	8	24	38	81
Sexual Orientation	11	15	11	36	73
	57	84	86	216	443

Table 2: Correlation of HS categories with (not) violence (nV - V) and directed/generalized (D - G) labels

(last column) and their correlation with violence and directed/generalized labels is further portrayed.

2.5 Dataset Baseline Evaluation

In order to evaluate ETHOS, after pre-processing the data, we used a variety of algorithms in multi-label scope to present the baseline performance in this dataset. For the purpose of providing the unbiased performance of each algorithm we performed nested-CV evaluation, using a variety of parameter setups, for every algorithm except NNs, where we applied 10-fold-CV. In addition, we binarise the values of each label, which are initially discrete in a range of $[0,1]$, to the $\{0,1\}$ classes using the rule “*If value $\geq 0.5 \rightarrow 1$ Else value $\rightarrow 0$* ”.

The algorithms handling Multi Label Learning (MLL) can be either problem transformation or adaptation techniques [6]. MLkNN [7] and MLARAM [8], as well as Binary Relevance (BR) and Classifier Chains (CC) [9] with base learners like Logistic Regression (LR), SVMs and Random Forests (RF), are utilised. We used FastText (FT) embeddings for our Neural Networks (NNs) and designed models inspired by classic MLL systems, such as BR and CC. Specifically, NNBR is an NN containing BiLSTMs, an attention layer, two feed forward and an output layer with 8 outputs in a BR fashion. NNCC is inspired by the CC technique, but during its output, each label is given as input for the next label prediction.

In the evaluation of MLL systems, a very common measure is the Hamming loss (symmetric difference between the ground truth labels and the predicted ones). Furthermore, subset accuracy (symmetric similarity), as well as F_1 -score (micro), are contained here. We present our results in Table 3. The superior performance of neural-based approaches compared to classical ML models is observed. Specifically, NNBR achieves the highest score in 12 out of 13 metrics.

	F_1	Subset	Hamming
	Micro	Accuracy	Loss
MLkNN	53.74	26.53	0.1566
MLARAM	18.71	7.15	0.2948
BR	56.76	26.28	0.1395
CC	58.23	31.4	0.1606
NNBR	74.87	48.39	0.0993
NNCC	55.47	26.61	0.1378

Table 3: Performance of selected models on MLL HS (P: Precision, R: Recall, AP: Average Precision)

3 Optimum Attention Analysis for Interpretability of Transformers in Text Classification

Attention-based interpretability techniques, are much faster than other commonly employed methods. However, no conclusive study was performed, showcasing the optimal interpretation extraction process from attention. As a consequence, attention is often overlooked when discussing interpretability techniques for transformer models, and specifically for text classification. The main contribution of this research is an extensive attention analysis to identify the optimal interpretation extraction procedure. Furthermore, we propose a novel metric for evaluating feature importance based interpretability methods, while we also propose a different strategy regarding the evaluation process of faithfulness based metrics, that is better suited for text-classification interpretability tasks performed by transformer models.

3.1 Attention-based interpretations

Several works used attention information to produce interpretations in the past, while others state that attention cannot be used for such task. However, the process of extracting those interpretations is not always straightforward. Attention information exists in the form of matrices in each encoder/decoder layer and their attention heads [10].

Studying the corresponding literature, we gathered different ways researchers handle this attention information. The most common approach is averaging [11, 12, 13] or summing [14, 15] the attention heads. Similarly, averaging [15] and multiplying [13] are the most common operations applied on layers. Regarding how the final interpretation is produced, some common approaches are selecting the row corresponding to the $[CLS]$ token (*From*) [13, 12], selecting the maximum value from each column (*Max Columns*) [15] or averaging the columns of the attention matrix (*Mean Columns*) [16]. Two out of these approaches for extracting the interpretation from the attention matrix, *From* and *Mean Columns*, are presented in Figure 3.

Inspired by the literature, we further present two similar novel strategies *To*,

	[CLS]	I	Need	Attention	[SEP]		[CLS]	I	Need	Attention	[SEP]		[CLS]	I	Need	Attention	[SEP]		[CLS]	I	Need	Attention	[SEP]
[CLS]	0.17	0.14	0.32	0.35	0.01		0.17	0.14	0.32	0.35	0.01		0.17	0.14	0.32	0.35	0.01		0.17	0.14	0.32	0.35	0.01
I	0.06	0.23	0.30	0.39	0.01		0.06	0.23	0.30	0.39	0.01		0.06	0.23	0.30	0.39	0.01		0.06	0.23	0.30	0.39	0.01
Need	0.05	0.08	0.68	0.18	0.00		0.05	0.08	0.68	0.18	0.00		0.05	0.08	0.68	0.18	0.00		0.05	0.08	0.68	0.18	0.00
Attention	0.08	0.07	0.15	0.69	0.01		0.08	0.07	0.15	0.69	0.01		0.08	0.07	0.15	0.69	0.01		0.08	0.07	0.15	0.69	0.01
[SEP]	0.18	0.17	0.19	0.17	0.30		0.18	0.17	0.19	0.17	0.30		0.18	0.17	0.19	0.17	0.30		0.18	0.17	0.19	0.17	0.30
	0.17	0.14	0.32	0.35	0.01		0.17	0.06	0.05	0.08	0.18		0.20	0.20	0.20	0.20	0.20		0.11	0.14	0.33	0.36	0.07

F: From [CLS]
T: To [CLS]
MR: Mean Rows
MC: Mean Columns

Figure 3: Interpretation extraction methods from an attention matrix

which uses the attention scores each token has towards $[CLS]$, and *Mean Rows*, that averages the rows of the attention matrix to produce the interpretation. The logic behind the first strategy is that since the $[CLS]$ token is used by the Transformer for its final decision regarding the instance, the attention scores towards it should provide reasoning for that decision. On the other hand, *Mean Rows* is examined for completeness sake. Both strategies are also visible in Figure 3.

To the best of our knowledge, no research has been conducted, concerning the optimal way to extract interpretation from attention matrices for text classification tasks. Towards discovering the best approach to extract accurate interpretations from attention, we combine all these different strategies per layer, head and interpretation extraction method. The available options for attention heads found in the literature were, averaging and summing, while multiplying and averaging are commonly used for layers. We, additionally, experiment with summing attention layers as an available option, as well as selecting specific attention heads or layers, following the logic that different attention heads or layers hold different types of information. An additional strategy for selecting the most informative heads from each layer, inspired by a recent study [17], is also employed. Finally, we use 6 different strategies to derive the interpretation from the attention matrix, namely, *From*, *To*, *MeanRows*, *MeanColumns*, discussed earlier as well as *MaxRows* and *MaxColumns*, which selects the maximum value from each row/column of the attention matrix.

3.2 Ranked Faithful Truthfulness

To evaluate the different methods, we developed a novel measure based on two others regularly used in the literature. These two, namely Faithfulness and Truthfulness, albeit informative, do not fully represent the quality of the produced explanations. Specifically, Faithfulness considers solely the token with the highest importance when evaluating the explanation, while Truthfulness assigns

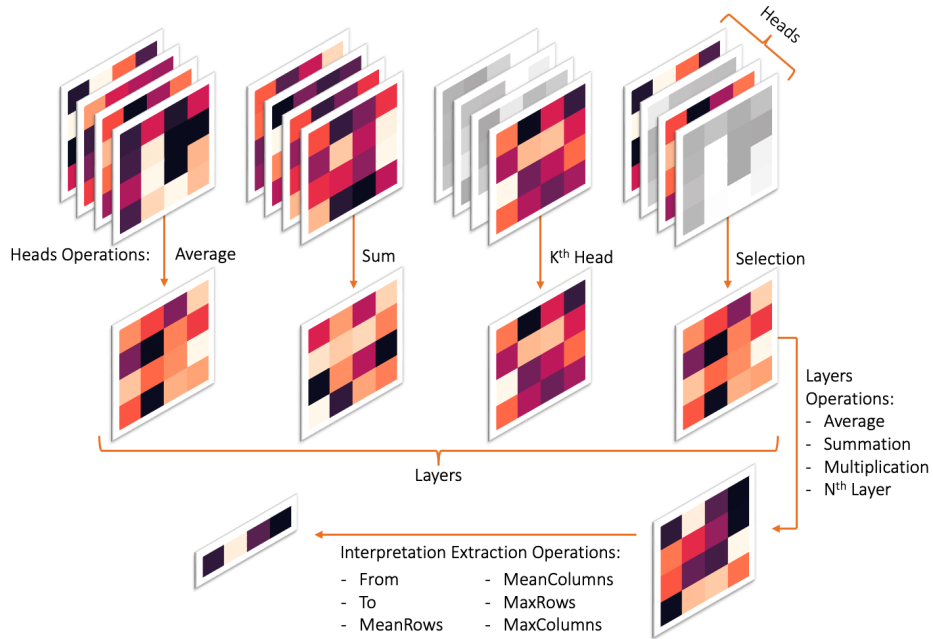


Figure 4

the same penalty for each token regardless of its importance score. As such, we propose a new FI evaluation metric called Ranked FT (Faithful Truthfulness), that not only considers each token when evaluating the quality of the explanation, but also assigns a different penalty to each one based on their importance to the explanation.

$$RankedFT = \frac{1}{|L|} \sum_l^{ |L| } RankedFT_l \quad (1)$$

The mathematical formulation for *RankedFT*, across all labels can be found in Equation 1, which corresponds to the mean of the values computed for each label (*RankedFT_l*). To calculate *RankedFT_l* for a specific label *l*, we first iterate through the examined instances *X* that are subject to prediction *l*. For each instance $x_e \in X, P(x_e)^l > 0.5$, we perform *N* modifications by iteratively removing one token at a time (modify). Then, we compare the changes in the model’s prediction, regarding label *l* between each modified instance and the original one as seen in Equation 2.

$$\text{compare}(x_e, x'_e, w_{e,i}, l) = \begin{cases} P(x_e)^l - P(x'_e)^l, & \text{If } w_{e,i} > 0, \\ P(x'_e)^l - P(x_e)^l, & \text{If } w_{e,i} < 0, \\ -1 \times |P(x_e)^l - P(x'_e)^l|, & \text{If } w_{e,i} = 0 \end{cases} \quad (2)$$

$w_{e,i}$ denotes the importance score of the removed token provided by the FI technique, and the comparison is done based on its sign. We further penalize the outcome of this comparison, according to the ranking of the token’s importance score. This whole process is depicted in Equation 3. Higher values on this metric indicate better performance.

$$RankedFT_l = \frac{1}{|X|} \sum_e \sum_i \frac{compare(x_e, modify(x_e, i), w_{e,i}, l)}{penalty(z_e, i)} \quad (3)$$

3.2.1 Token removal process in Transformers

Faithfulness-oriented interpretability metrics, including *RankedFT*, evaluate the performance of the method based on how the model’s decision changes when the most important element of the input is removed. When dealing with textual inputs, this element refers to the most influential token to the decision. In transformers, however, removing a word from the input sequence results in the order of words changing, which in turn affects the context and how attention is computed. In Figure 5, we show an example of such change in attentions, with image (a) representing the attentions of the initial sequence, (b) the one after the most important token is removed, and (c) when replaced with the *[UNK]*.

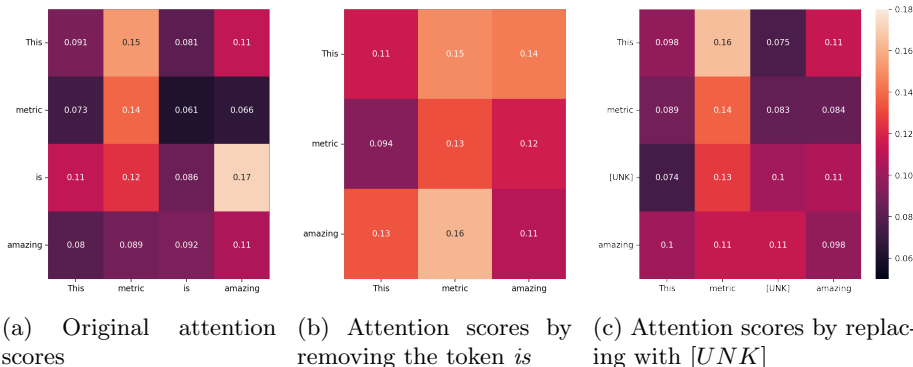


Figure 5: Example of attention with token removal and replacement with *[UNK]*

Completely removing a token from the sequence (Figure 5b), affects the attentions between the remaining ones more, in contrast to replacing it with *[UNK]* (Figure 5c). This can be identified by analyzing the attentions between the tokens of the sequence in each of these two cases compared to the original, as seen in Figure 5. For example, the attention the token *metric* has towards *amazing* is 0.066 in the original sequence. By removing the token *is* the attention increases to 0.12, which is to be expected since the context of the sequence changed and these tokens are next to each other now. On the other hand,

replacing *is* with *[UNK]*, only slightly increases the attention to 0.084, since that change does not affect the position of the examined tokens.

Based on the example, simply removing the token not only nullifies its influence in that sequence but also affects the relations between the other tokens, by shifting their context. Transformers are contextual models, which means that the context of each examined token affects the final decision. As a result, instead of removing each token from the sequence in order to measure and assess its influence on the final decision, we replace it with *[UNK]* during the Ranked FT and Faithfulness evaluation processes. This way, we nullify the influence of the replaced token while minimally affecting the context of the sequence.

4 Local Multi Label Explanations for Random Forests

LionForests (LF) is a local explainability technique that uses rules in order to provide explanations for the decisions of a Random Forests (RF) model. A key advantage of LF is that it does not need to meddle with the architecture of the examined model, as it distils the interpretation from the knowledge already present in RF. This in turn means that the explanations are provided without any demerits in the model’s performance or complexity. LF can be used in binary or multi-class classification and regression problems without any significant adjustments.

LF provides explanations for each instance. The main step of the algorithm behind the interpretation extraction process is the estimation of the minimum number of paths across the different estimators of RF that cover the examined instance. This step identifies the main set of rules upon which LF builds the interpretation for that instance through feature and path reduction and feature-range formulation. The estimation of the minimum number of paths is not a straightforward task, especially since it needs to comply with LF’s main property, namely *conclusiveness*. This property requires the rules produced by an explainability technique to be free of misleading or erroneous elements. Since multi-label classification problems can be delegated to a series of different binary problems, one for each of the examined labels, we will further discuss how LF computes the minimum number of paths for binary tasks while retaining the *conclusiveness* property.

LF is applicable in single-label or multi-class classification and regression tasks. This work aims to extend the scope of problems LF is applicable, to also include multi-label classification. In multi-label classification, the predictions come in matrices of size $|L|$, with L denoting the set of available labels. Explanations in these scenarios concern either the whole predicted label set $L_p \subseteq L$, subsets of it $L'_p \subseteq L_p$, or even each one of the labels $l \in L_p$ comprising it separately. We employ three different strategies that allow LF to export multi-label explanations, which differ on how we calculate the quorum discussed earlier, based on the subset of L that we want our explanation to cover.

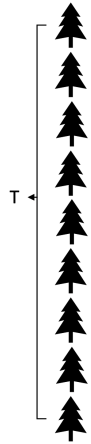
	<u>Prediction:</u>	<u>Per Label:</u>			<u>All:</u>	<u>Frequent Set:</u>
RF:	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]	[1, 1, 1, 1, 1]
	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]	[0, 0, 0, 0, 0]
	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]	[0, 1, 0, 0, 1]
	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]	[0, 1, 0, 0, 0]
	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]	[1, 1, 0, 1, 1]
	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]	[0, 1, 0, 1, 1]
		8→5	6→5	7→5	6→5	6→5

Figure 6: Running example

We also introduce an example, which we are going to use throughout the different strategies. Consider an RF model with $9 = |T|$ estimators, predicting $5 = |L|$ different labels for a given input. Based on the theory presented before, the quorum equals $\lfloor \frac{9}{2} + 1 \rfloor = \lfloor 5.5 \rfloor = 6$. For a given instance, RF predicts the following label set $[0, 1, 0, 1, 1]$. From each $t \in T$ tree estimator, we extract the path and the prediction for this instance. Then, based on the strategy, we proceed to the appropriate reduction and eventually the formulation of the final rule interpretation. In Figure 6, the predicted label sets from each t tree are visible.

4.1 Explaining each predicted label separately

The first step of this strategy (LF-1) is the extraction of all possible paths from the tree predictors comprising the RF model. Then, an iterative process for each predicted label l takes place, that first identifies the paths T which vote for its prediction. The next step is the reduction of T to the number denoted by the quorum, obtaining the minimum number of paths T' . The rule building steps remain the same as those in the original technique, namely feature aggregation and handling of the categorical features. After formulating a rule for each predicted label l , we use these rules as an explanation for the examined instance.

If we look at the example in Figure 6, focusing on the second column *Per Label*, we can see how LF selects and reduces the paths to the quorum. It identifies the paths that voted for each of the three predicted labels. If the number of paths exceeds the quorum, the LF reduction strategies are used to

decrease them to the bare minimum (quorum). Treating each label separately can result in smaller feature sets in the final interpretation. This is because LF has a greater number of possible paths to reduce to the minimum.

4.2 Explaining all the predicted label set

This strategy is largely similar to the previous one, with the main difference being that instead of an iterative process for each label ($LF-a$), this time a single process is executed for the whole predicted label set. This in turn means that LF must now identify the paths T that vote for the whole predicted label set, greatly reducing the number of available paths to be reduced in the following step, if possible. Furthermore, during the path reduction step, each produced path set must cover the whole prediction, limiting the number of paths LF can safely remove from T in order to obtain T' . It is worth noting that, due to the above conditions, the final rule obtained after applying the rule-building steps is very specific to the examined instance. There is a possibility that the number of recognized paths covering all predicted labels will be less than the quorum. This prevents us from further decreasing them, but also prohibits us from using them alone to form the final rule. In this scenario, regardless of their vote, we use all the paths.

Connecting this strategy with the running example of Figure 6, we focus on the third column, *All*. Only 6 paths include all the predicted labels at the same time. LF will use the reduction strategies to decrease those pathways to 5 (quorum). However, because there is so little room for reduction, their effectiveness is limited, and therefore, we might not observe the desired feature reduction.

4.3 Explaining frequent label subsets

This strategy provides explanations for subsets ($LF-p$) of the predicted label set that frequently appear inside the examined data set. These subsets are identified with the use of association rules and specifically the *fpgrowth* algorithm. Then, an iterative process comparable to the one present in the first strategy is performed. For each subset, the paths T that vote for all the labels present inside it are identified and then reduced to T' , before the rule building steps that formulate the final rule for this subset are implemented. The final explanation for the frequent subsets is an aggregation of the rules built by the aforementioned process.

In case of larger label sets, as well as a large set of predicted labels, the number of activated subsets can be very high. Therefore, the end-user is given an option to limit the number of subsets. Hence, if the activated subsets are X and the user asks for $N < X$, the first N subsets and their explanation will be provided, ordered based on the support of the subset across the label sets of the training data set.

In the example (Figure 6), the last column presents the explanation of one identified subset $[0, 1, 0, 1, 0] \subset [0, 1, 0, 1, 1]$, the paths which cover this set, and

the removed path.

4.4 Experiments

We carried out a set of experiments to compare the performance of our strategies to state-of-the-art techniques frequently used in the literature. We performed three distinct sets of experiments to provide a fair comparison. The first compares our various strategies to each other in order to gain insight into their effectiveness with multi-label data sets. The second focuses on techniques that explain the entire predicted label set, pitting our second strategy against similar methods described in the literature. The third and final set compares our first strategy to two different state-of-the-art competitors typically used in the same task, providing explanations for each label separately. To further the reliability of our results, we performed a 10-fold cross validation.

The experimental procedure conducted for this work, showed that the multi label extension retains LF’s main property *conclusiveness*, while also providing informative rules. Furthermore, the time response of the method was found to be close or faster, in some cases, to other techniques in the literature that do not comply with the *conclusiveness* property.

References

- [1] K. Dinakar, R. W. Picard, H. Lieberman, Common sense reasoning for detection, prevention, and mitigation of cyberbullying (extended abstract), in: Q. Yang, M. J. Wooldridge (Eds.), Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015, AAAI Press, 2015, pp. 4168–4172.
URL <http://ijcai.org/Abstract/15/589>
- [2] M. Jirotko, B. C. Stahl, The need for responsible technology, Journal of Responsible Technology 1 (2020) 100002. doi:<https://doi.org/10.1016/j.jrt.2020.100002>.
URL <http://www.sciencedirect.com/science/article/pii/S2666659620300020>
- [3] M. Sharma, D. Zhuang, M. Bilgic, Active learning with rationales for text classification, in: R. Mihalcea, J. Y. Chai, A. Sarkar (Eds.), NAACL HLT 2015, Denver, Colorado, USA, May 31 - June 5, 2015, The Association for Computational Linguistics, 2015, pp. 441–451. doi:[10.3115/v1/n15-1047](https://doi.org/10.3115/v1/n15-1047).
URL <https://doi.org/10.3115/v1/n15-1047>
- [4] O. G. R. Pupo, A. H. Altalhi, S. Ventura, Statistical comparisons of active learning strategies over multiple datasets, Knowl. Based Syst. 145 (2018) 274–288. doi:[10.1016/j.knosys.2018.01.033](https://doi.org/10.1016/j.knosys.2018.01.033).
URL <https://doi.org/10.1016/j.knosys.2018.01.033>

- [5] M. Inc., Kappa statistics for attribute agreement analysis, Available at <https://support.minitab.com/en-us/minitab/18/help-and-how-to/quality-and-process-improvement/measurement-system-analysis/how-to/attribute-agreement-analysis/attribute-agreement-analysis/interpret-the-results/all-statistics-and-graphs/kappa-statistics/> (2021/04/17) (2021).
- [6] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, *International Journal of Data Warehousing and Mining (IJDWM)* 3 (3) (2007) 1–13.
- [7] M.-L. Zhang, Z.-H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, *Pattern recognition* 40 (7) (2007) 2038–2048.
- [8] F. Benites, E. Sapozhnikova, Haram: A hierarchical aram neural network for large-scale text classification, in: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), IEEE Computer Society, USA, 2015, pp. 847–854. doi:10.1109/ICDMW.2015.14.
- [9] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Springer, Bled, Slovenia, 2009, pp. 254–269.
- [10] E. Reif, A. Yuan, M. Wattenberg, F. B. Viegas, A. Coenen, A. Pearce, B. Kim, Visualizing and measuring the geometry of bert, *Advances in Neural Information Processing Systems* 32 (2019).
- [11] Y. Wang, H.-Y. Lee, Y.-N. Chen, Tree transformer: Integrating tree structures into self-attention, 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)*, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1061–1070. doi:10.18653/v1/D19-1098. URL <https://aclanthology.org/D19-1098>
- [12] B. Mathew, P. Saha, S. M. Yimam, C. Biemann, P. Goyal, A. Mukherjee, Hatexplain: A benchmark dataset for explainable hate speech detection, in: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 14867–14875. URL <https://ojs.aaai.org/index.php/AAAI/article/view/17745>
- [13] H. Chefer, S. Gur, L. Wolf, Transformer interpretability beyond attention visualization (2021). arXiv:2012.09838.

- [14] B. Hoover, H. Strobel, S. Gehrmann, exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 187–196. doi:10.18653/v1/2020.acl-demos.22. URL <https://aclanthology.org/2020.acl-demos.22>
- [15] L. Schwenke, M. Atzmueller, Show me what you’re looking for: visualizing abstracted transformer attention for enhancing their local interpretability on time series data, in: The International FLAIRS Conference Proceedings, Vol. 34, 2021.
- [16] K. Clark, U. Khandelwal, O. Levy, C. D. Manning, What does BERT look at? an analysis of BERT’s attention, in: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Florence, Italy, 2019, pp. 276–286. doi:10.18653/v1/W19-4828. URL <https://aclanthology.org/W19-4828>
- [17] M. Behnke, K. Heafield, Losing heads in the lottery: Pruning transformer attention in neural machine translation, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 2664–2674. doi:10.18653/v1/2020.emnlp-main.211. URL <https://aclanthology.org/2020.emnlp-main.211>