information

Article

Multi-Task Romanian Email Classification in a Business Context

Alexandru Dima [1,2], Stefan Ruseti [1], Denis Iorga [2,3], Cosmin Karl Banica [2] and Mihai Dascalu [1,2,*]

1 Computer Science & Engineering Department, University Politehnica of Bucharest, 313 Splaiul Independentei, 060042 Bucharest, Romania; alexandru.dima1609@stud.acs.upb.ro (A.D.); stefan.ruseti@upb.ro (S.R.);
2 Research Technology, 19D Soseaua Virtutii, 060782 Bucharest, Romania; denis.iorga@drd.unibuc.ro (D.I.); cosmin@research-technology.ro (C.K.B.)
3 Interdisciplinary School of Doctoral Studies, University of Bucharest, 4-12 Bulevardul Regina Elisabeta, 030018 Bucharest, Romania
* Correspondence: mihai.dascalu@upb.ro

Abstract: Email classification systems are essential for handling and organizing the massive flow of communication, especially in a business context. Although many solutions exist, the lack of standardized classification categories limits their applicability. Furthermore, the lack of Romanian language business-oriented public datasets makes the development of such solutions difficult. To this end, we introduce a versatile automated email classification system based on a novel public dataset of 1447 manually annotated Romanian business-oriented emails. Our corpus is annotated with 5 token-related labels, as well as 5 sequence-related classes. We establish a strong baseline using pre-trained Transformer models for token classification and multi-task classification, achieving an F1-score of 0.752 and 0.764, respectively. We publicly release our code together with the dataset of labeled emails.

Keywords: multi-task classification; token classification; email classification corpus; natual language processing; language models

check for updates

Citation: Dima, A.; Ruseti, S.; Iorga, D.; Banica, C.K.; Dascalu, M. Multi-Task Romanian Email Classification in a Business Context. *Information* **2023**, *14*, 321. https://doi.org/10.3390/info14060321

Academic Editor: Kostas Stefanidis

Received: 12 May 2023
Revised: 31 May 2023
Accepted: 1 June 2023
Published: 3 June 2023

1. Introduction

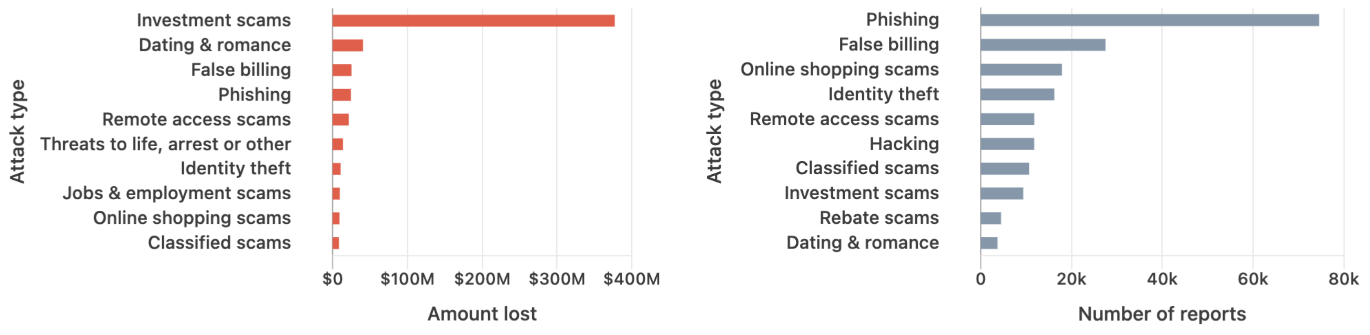
1.1. Overview

Email classification systems are essential for handling and organizing communication, especially in a business context where an increasing volume of emails is observed. As such, email service providers, for instance, Outlook or Google Email, rely on such systems to help users classify their received emails into multiple categories. Besides standard categories generally made available by all service providers (e.g., "Spam"), high variability is encountered among providers. For example, Outlook uses a "Focused" tab for the most important email messages and an "Other" tab for the rest. Google uses more descriptive categories, namely: (a) "Social" for messages from social networks and media-sharing sites, (b) "Promotions" for deals, offers, and other promotional emails, (c) "Updates" for notifications, confirmations, receipts, bills, and statements, and (d) "Primary" for important messages and messages that are not classified in any of the other categories. This classification can help users focus on the important emails firsts, thus decreasing the time spent on managing email each day. Unlike the spam category, these categories depend on the email client and are thus nonstandard.

Spam (https://www.malwarebytes.com/spam; accessed on 12 May 2023) emails refer to any kind of unwanted email or unsolicited digital communication usually sent in bulk. There are many types of spam emails; some are relatively harmless, like marketing messages for unsolicited goods, while others are harmful. Harmful spam can include messages that can spread malware, trick the receiver into disclosing personal information, or compel the receiver into paying in order to get out of trouble. Even though people generally assume that spam and scammers do not have a significant influence or reach in today's society, statistics from the FBI (https://www.gasa.org/post/fbi-internet-crime-report-2022; accessed on 12 May 2023) and the Australian government (https://www.scamwatch.gov.au/scam-statistics?scamid=all&date=2022; accessed on 12 May 2023) suggest the contrary.

We observe in Figure 1a that harmful spam has a big impact when considering the amount of money lost if the attacker is successful. When comparing the previous attack types with the ones present in Figure 1b, we notice that the most common attack types may not necessarily be the most damaging in terms of monetary impact, but this does not undermine their inherent danger. Another important thing to note is that email attacks are generally the third most common and harmful attack channel (see Figure 2). Besides the "Spam" category being covered by most email service providers, we observed a lack of consensus regarding the categories of harmful emails, each source having a slightly different definition. As such, we consider that the best categories for harmful spam are the following:

- Phishing Emails
  - These are the types of emails that cybercriminals use to trick the receiver into offering personal sensitive information (e.g., credit card, login credentials). They are very dangerous because they mislead the receiver into thinking that following the requests of the attacker is the best course of action;
  - A common phishing technique is email spoofing. These emails mimic an email from a legitimate sender (e.g., Google, Apple, PayPal) and ask the receiver to take action (e.g., payment of an invoice, request to reset a password, request for updated billing information).
- Scams
  - These emails usually trick the user into thinking an important reward is received for completing an action. We give a few examples of types of scams that a receiver could be subjected to;
  - Current events scams are emails that use hot new topics to get attention. For example, during the COVID-19 pandemic, scammers sent spam messages offering remote jobs that paid in Bitcoin or messages offering financial relief for small businesses. Ultimately to complete the scam, the scammers would ask for bank account details;
  - Another type of scam is advance-fee scam emails. These scam emails promise a big financial reward; however, the receiver must first pay a comparatively small advance to be eligible to obtain the reward. Usually, the advance is a processing fee needed to unlock a larger amount of money. A small variation on this idea involves the scammer pretending to be a family member of the receiver who is in trouble and in need of money.
- Malspams
  - These are the types of emails that are used to infect the receiver's device with malware. Usually, this is done by clicking a link to a malicious website or opening an attachment that contains malware (e.g., ransomware, trojan, bots, crypto-miners, spyware, keyloggers). A common technique is to use a PDF, Word, or Powerpoint file with a malicious script that runs when the attachment is opened.



(**a**) Top-10 Attacks by the Amount Lost (in A$) (**b**) Top-10 Attacks by the Number of Reports

Figure 1. The Reach and Damages Produced by Attackers.

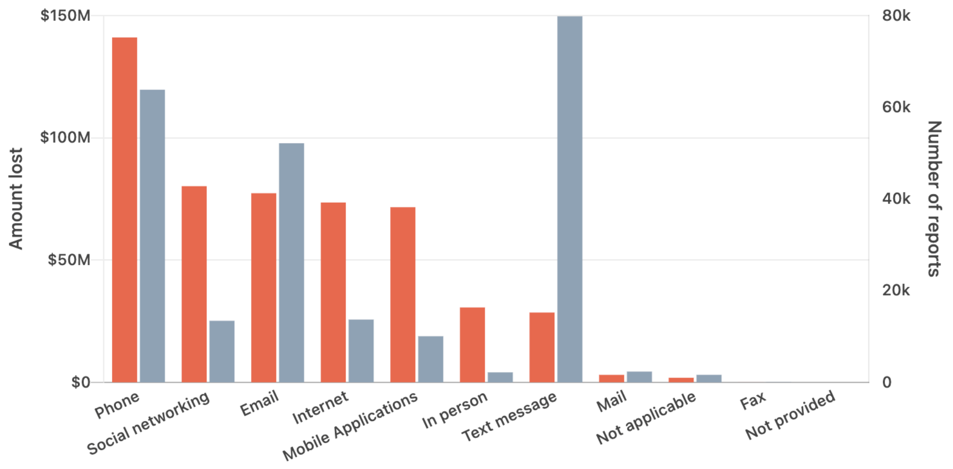


Figure 2. Attack Delivery Channels (orange denotes Amount lost, while gray reflects the Number of reports).

The previous analysis leads to three observations. First, service providers offer different classes for email classification, while others do not have extra classes at all. Second, the classes are not constructed from a business standpoint. Thirdly, spam emails (both relatively harmless and harmful) can sometimes pass through the email provider's filters into the main inboxes.

1.2. Business Context in Romania

The previous observations and statistics are also relevant to the local context in Romania. While domestic reports and statistics for spam do not exist, there are reports concerning spam emails in international reports. Considering that Romania was in 2010 in the top 10 spam email senders (https://www.zf.ro/business-hi-tech/romania-este-in-top-10-la-nivel-mondial-dupa-numarul-de-spam-uri-expediate-6499280, https://www.romania-insider.com/romania-ranks-fifth-in-top-of-spam-source-countries-panda-security-report; accessed on 30 May 2023) the situation improved considerably since in 2021 the country was in the lowest tier for both senders and receivers of spam (https://securelist.com/spam-and-phishing-in-2021/105713/; accessed on 30 May 2023). However, the situation can still be improved as Romania is the 51st country in the world regarding spam-sending (https://talosintelligence.com/reputation_center/email_rep#spam-country-senders; accessed on 30 May 2023). Moreover, the context is even more complex and problematic since the number of unsolved cases regarding computer crimes quadrupled from the start of April 2021 to the end of March 2023 (ro. "infracțiuni informatice") (https://recorder.ro/politia-romana-e-coplesita-de-criminalitatea-informatica-peste-21-000-de-dosare-zac-nerezolvate/; accessed on 30 May 2023).

Considering all of the previously mentioned information, we consider that accurately identifying both relatively harmless and harmful spam is an important task for email classification in a business context. Additionally, there is a high need for categories focusing on business-related metrics. By business-related metrics, we refer to any email that supports workers' productivity in a company. This can span both straight-forward categories (e.g., whether the emails are business-related or personal) or more particular categories (e.g., whether the email entails an action from the reader, whether the email is automatically generated or written by a human)

1.3. Related Work

1.3.1. Public Datasets for Email Classification

There are several directions when it comes to the classification of emails. These directions can be summed up into three types of tasks. The first one refers to spam classification. This type of task requires a discriminator model to differentiate between spam and ham emails. Usually, "spam" refers to any type of spam message for this task—i.e., both harmful and relatively harmless spam (according to our prior categorization). The second type of task refers to harmful spam classification. This type of task requires a discriminator model to differentiate between ham emails and harmful spam emails. "Harmful spam" only refers to one type of harmful email for this task—for example, only phishing emails or only fraud emails. The third type of task refers to "normal" classification. This type of task implies the classification of only ham emails in different classes. For this, a large variety of possible classes can appear. Considering these directions, various public datasets can be used to develop email classification systems.

One example of a public email dataset useful for developing an email classification system is the Enron dataset [1]. The dataset contains approximately 500.000 generated by employees of the Enron Corporation. The Federal Energy Regulatory Commission obtained it while investigating Enron's collapse. This dataset has two columns: one representing the path of a certain email message which contains information related to the user and the mailbox directory from which the email was extracted, and the email message with both the header metadata and the email body. This dataset does not have annotated labels. Other public datasets contain annotated labels, for instance, a dataset that contains emails weakly separated into 4 classes: Crime, Entertainment, Politics, and Science (i.e., one email may have multiple labels, only one of which is accurate) [2]. Considering business-related datasets, subsets of the Enron datasets were annotated for business email classification. There have been two variations of annotation using subsets of Enron, one using only two labels (i.e., "Business" and "Personal") [3], and another with six classes (i.e., "Business", "Somehow Business", "Mixed", "Somehow Personal", "Personal", and "Cannot Determine") [4]. The latter also annotated a subset of the Avocado Research Email Collection [5] with the same classes.

Other datasets are specifically designed for spam classification. For example, SpamAssassin Corpus [6] is a dataset containing approximately 4200 ham messages and 1900 spam messages. Similarly to the Enron corpus, the messages contain the email header metadata and the email body. Yet another example is Spambase [7], a dataset composed of 4600 examples. Unlike the previous datasets, each example is based on statistics extracted from the used emails; thus, it does not include the body or headers of the emails. The statistics used are word frequencies of 48 selected words, character frequencies of 6 characters, and a few statistics based on consecutive capital letters found in the body.

Other datasets use a filtered subset of emails from the Enron corpus augmented with spam emails collected from other corpora (e.g., the SpamAssassin corpus) or personally-collected spam emails. Two such datasets are worth mentioning. The first is the Enron-Spam dataset [8], containing the ham emails from 6 employees, which had many messages in the Enron corpus augmented with spam messages as described previously. Each employee is considered a different slice of the dataset, the first three having a ratio of approximately 3:1 ham to spam emails; in contrast, the last three have a ratio of approximately 1:3 ham to spam. Each slice adds up to approximately 5000–6000 emails. The second dataset is the TREC 2007 Public Corpus Dataset [9] that contains approximately 25,000 emails from the Enron corpus, augmented with an additional 50,000 spam emails; each email, consisting of header and body, has a corresponding label (i.e., ham, spam).

Two main datasets are generally used in literature when considering harmful spam emails. The first is Nazario's Phishing Corpus [10], which contains approximately 4500 phishing emails. The documents are stored using the mbox format, each containing emails with header metadata and body. At present, the dataset is not available at its original location; however, it can still be retrieved by following the instruction in a GitHub repository (https://github.com/diegoocampoh/MachineLearningPhishing, accessed on 12 May 2023), namely by using Wayback Machine to find a time when the original link (http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus; accessed on 12 May 2023) was working. The second dataset is the CLAIR collection of fraud emails [11], which contains approximately 4000 fraudulent emails. The dataset contains both the metadata header and the body of the emails, and the emails can be more exactly classified as advance-fee scam emails.

1.3.2. Existing Approaches in Email Classification

Email classification tasks (or classification in general) can be modeled as single-label (i.e., binary or multi-class; one class/label for each entry) or multi-label (i.e., multiple labels can be assigned for each entry); in both cases, labels are from a finite set. We will present examples of each in the following paragraphs. From our observations, most approaches tackle single-label classification. We present works on spam detection, harmful spam detection, and "normal" email classification in this specific order. It is also important to mention that part of the presented approaches used datasets that are not publicly available, so their results cannot be easily compared with other approaches. Another important distinction is that part of existing approaches for detecting phishing emails require opening the links present in the body of the email and investigating the DOM (Document Object Model) and the CSS (Cascading Style Sheets) of the corresponding page.

Alghoul et al. [12] used a Multi-Layer Perceptron (MLP) model with features engineered based on selected word and letter frequencies present in the email body for identifying spam emails. While it was not mentioned, the features seem to be the same as the ones used in Spambase, meaning that this may be the dataset used for this paper. Li et al. [13] adopted a semi-supervised learning approach, using labeled and unlabelled data to detect whether an email is spam. The authors applied three classification algorithms: Naive Bayes, KNN (K-Nearest Neighbours), and C4.5 (tree-based classifier). Their approach relied on features engineered using header and body statistics and multi-view co-training in an ensemble manner. It is important to mention that their features did not involve the words from the body, as most features were related to the header of the email. Unfortunately, their dataset is not publicly available. Each classifier used a distinct dataset based on the initially labeled dataset and the examples from the unlabeled dataset, which diminishes the classification error when added to the training set of the classifier. Sharaff and Gupta [14] have attempted spam classification using features engineered from word frequencies, from the Spambase dataset, with tree-based classifiers (i.e., extra-tree, decision tree, and random forest). The authors considered PSO (Particle Swarm Optimization) variants and GAs (Genetic Algorithms) to select the most relevant features. Pan et al. [15] converted the spam email classification problem to graph classification using GCNs (Graph Convolutional Networks). They used three node types for learning (i.e., word, text, and topic nodes), where the topics were mined from the email bodies using LDA (Latent Dirichlet Allocation) [16]. Saleh et al. [17] developed a spam classification model using a Negative Selection Algorithm (NSA) [18] and a knowledge base created from keywords extracted from ham and spam emails from the Enron-Spam dataset.

Yasin and Abuhasan [19] tackled phishing classification by engineering features based on certain word frequencies. The authors tried the following classification algorithms:

SVM (Support Vector Machine), C4.5, Random Forest, and MLP, out of which the best results were obtained with Random Forest. An unusual detail in their approach is that they had three types of emails: ham, spam, and phishing; however, they did not have a clear separation between spam and phishing since the examples came from different datasets. Also, they considered both ham and spam as being legitimate emails. Niu et al. [20] used an SVM-based classifier for identifying phishing emails. Unlike other already mentioned approaches, they constructed features using the emails' body and header. However, most features generated from the body relied on the links from the email body.

Egozi and Verma [21] and Harikrishnan et al. [22] also used classical Machine Learning (ML) approaches (e.g., SVM, Trees, Logistic Regression) for phishing detection on a dataset that is no longer available. The former engineered their features based on statistics constructed from words, stop-words, and punctuation frequency/occurrences, while the latter used TF-IDF encoding (Term Frequency—Inverse Document Frequency) with NMF (Non-negative Matrix Factorization) and SVD (Singular Value Decomposition) for feature extraction and dimensionality reduction. Another important mention in the latter approach is that the vocabulary for TF-IDF was built not only on the training data but also on the testing data, a methodology we consider incorrect.

Fang et al. [23] attempted a more novel approach to phishing classification by using a multi-level RCNN (Region-Based Convolutional Neural Network) model with attention. They used the header and body of the email and created a character-level vector and a word-level vector using Word2Vec [24] for each of them. This resulted in vectors that were used for the RCNN model. Similar to previously discussed models, their dataset is currently unavailable. Alhogail and Alsabih [25] constructed GCNs from the body of the emails for their public dataset and afterward trained the model for the phishing classification task. Baccouche et al. [26] addressed the classification of both spam and fraud messages using Long-Short Term Memory (LSTM) [27] models and by constructing pseudo-labels from common bigrams between a YouTube comments dataset and the CLAIR fraud dataset.

Harsha Kadam and Paniskaki [28] introduce a multi-label classification for emails (without spam detection) using techniques like SVM, GRU (Gated Recurrent Unit), CNN (Convolutional Neural Network), and Transformer [29]; nevertheless, we find their methodology improper. Their labels were predefined topics on which customers of a certain company left questions and feedback. Sharaff and Nagwani [30] also attempted to solve a multi-label classification without tackling spam detection. The authors used a public dataset, which did not contain spam emails, and treated the problem as an unsupervised learning problem. By using LDA on the email dataset, they extracted frequent terms and divided them manually into multiple categories. Afterward, they clustered the emails with multiple algorithms (K-Means, Agglomerative Clustering, and NMF) and used LDA to extract the N most frequent terms in each cluster. The N terms were the label for each cluster, which was used to map the cluster into the manually defined categories. However, it is worth mentioning that they did not consider an objective accuracy metric.

Jlailaty et al. [31] performed business process identification on a small private dataset by using TF-IDF features and clustering emails using K-Means and Word2Vec. Alkhereyf and Rambow [4] attempted business and personal email classification using classical ML approaches, namely SVM and Extra-Trees [32] on the two datasets with classes. They used two feature sets, namely the encoded emails and features constructed by considering the thread relations of the emails. In a follow-up paper, Alkhereyf and Rambow [33] explicitly included thread relations using LSTMs and improved their feature encoding by using GraphSAGE [34]. Šošić and Graovac [35] also attempted business email classification on the same datasets by using BiLSTMs with Attention.

1.4. Research Objectives

Our research focuses on developing a versatile email classification system designed to categorize emails into various classes, including those pertaining to business-related and spam-related categories. Considering the previous works on email classification, our research objective is threefold: (1) introduce a dataset in Romanian for email classification from a business standpoint, (2) create automated models to curate and extract only the relevant content from the email body (i.e., remove signatures, disclaimers), and (3) fine-tune a pre-trained Transformer model on the curated data for multiple classification tasks, namely token classification with five possible labels (e.g., "Initial Addressing", "Final Addressing", "Signature", "Disclaimer", and "Personal Identifiable Information") and text classification for five categories (e.g., "Is Automatically-Generated", "Needs Action from User", "Is SPAM", "Is Business-Related", and "Type of Writing Style").

Following the aforementioned objectives, our main contributions are the following:

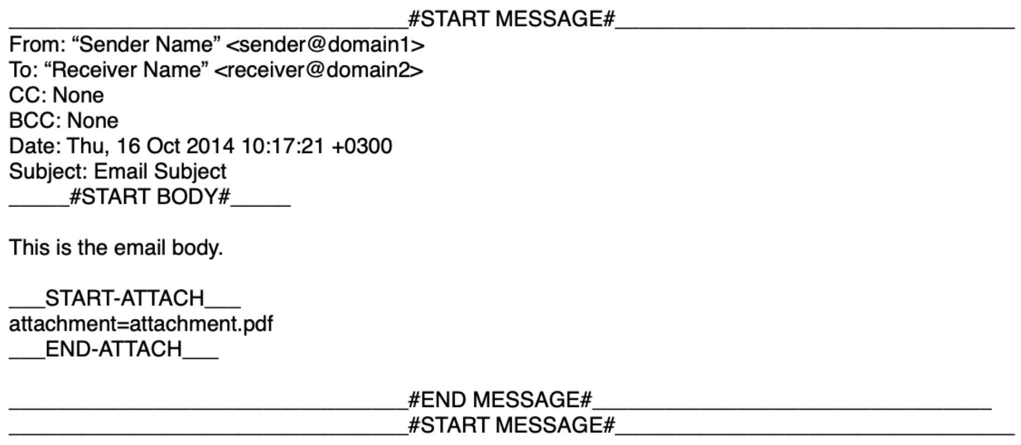
- Publish an anonymized version of a dataset of 1447 manually labeled emails on multiple business-related criteria. The dataset is available at: https://huggingface.co/datasets/readerbench/ro-business-emails (accessed on 12 May 2023);
- Introduce a strong baseline for the curation model that supports follow-up in-depth classifications;
- Develop a strong baseline for multi-task email classification consisting of the identification of "Personal Identifiable Information" and the five previously mentioned text classification categories. The entire codebase was open-sourced at: https://github.com/research-technology-ai/ro-business-emails (accessed on 12 May 2023).

2. Method

2.1. Dataset

2.1.1. Acquisition and Preprocessing

The initial raw data used to create the published dataset originated from 9.94 GB of text that represented the mailboxes of employees from 9 Romanian companies. The raw data contained a total of 515 useful mailboxes. We refer to mailboxes as the directories which contain emails and are displayed in an email client (e.g., "Inbox", "Sent", "Spam", "Trash", "Archive"). By useful mailbox, we consider directories not already marked as dangerous/useless by the email client; in other words, all directories except those like "Junk", "Spam", or "Trash". Each of the 515 mailboxes was initially saved in a text file containing all the emails in that particular mailbox. Each email comprises three components: header, body, and attachments. A mock email can be seen in Figure 3.


```
___________________________________#START MESSAGE#___________________________________
From: "Sender Name" <sender@domain1>
To: "Receiver Name" <receiver@domain2>
CC: None
BCC: None
Date: Thu, 16 Oct 2014 10:17:21 +0300
Subject: Email Subject
_____#START BODY#_____

This is the email body.

___START-ATTACH___
attachment=attachment.pdf
___END-ATTACH___

___________________________________#END MESSAGE#___________________________________
___________________________________#START MESSAGE#___________________________________
```

Figure 3. An example of an email from the raw data text files.

The data were pre-processed in multiple steps. First, we separated each email into its three components, using regular expressions constructed based on the format the data was saved into. The attachments were kept intact, while the header was subjected only to a few data regularization techniques (e.g., replacing null values and saving only the email addresses, not the names). However, the body needed more thorough processing. We started with an initial cleaning that entailed the use of regular expressions for removing HTML (HyperText Markup Language) and CSS artifacts that were present in the body and also the email thread (i.e., an email conversation containing a series of messages and replies) delimiters (i.e., repetition of characters like "_", "=", or "−"). An important problem arose when the email body was not just a simple message but a thread. These threads had to be separated into individual emails to capture important relations in the dataset for future experiments. The threads had different kinds of separators before each email; since the data was in the Romanian language, these were either in Romanian or English, depending on the language of the system where the email originated from. We identified the following types of separators:

- Simple separators
    - These are separators that do not contain any information about the message. A few examples of this type of separator are the following:
        - "-Mesaj redirecționat-"
        - "-Mesaj original-"
        - "Begin forwarded message:"
- Header-aware separators
    - These separators include information from the header of that specific message. The following are sample separators included in this category:
        - "În <date> <user name> <user email> a scris:"
        - "La <date> <user name> <user email> a scris:"
        - "On <date> <user name> <user email> wrote:"
        - "<date> <user name> <user email>:"
        - "Quoting <user name> <user email>:"
- Header-only separators
    - These separators include portions from the header of that specific message, thus being similar to the header of the original email. However, there is variability in the name of the fields and which field is present in the separator.

We separated the individual messages from threads using regular expressions tailored to the previously mentioned separator types and then divided each message into its header and body. This was done similarly to the main email; however, we constructed a regular expression that would accept a variable number of fields from a predefined list of possible fields since the fields from the header could have a large variability. As such, all the messages contained in threads were extracted. When separating the threads, we also saved a reference for each separated message that points to the previous messages in the thread in order to recreate any thread if need be.

Having the email data processed with the previously described methodology, the only remaining steps were the removal of duplicate and invalid emails. We defined a duplicate as any email where all the fields saved in our dictionary were identical to the ones of another email. While this procedure could not remove all the duplicates, this does not impact the final dataset with manually verified emails. Regarding invalid emails, we defined an email as being invalid if any of the following conditions appear:

- The "from" field in the header is missing or null
- The "date" field in the header is missing or null
- The email's body and attachment list are both empty

In the end, all these processed emails added up to roughly 400,000 emails, out of which 100,000 were soft duplicates (i.e., emails where the "from" and "date" fields are identical, but the rest of the fields were slightly different, either because of formatting or because of the loopholes in the email thread separation). As previously mentioned, these possible duplicates do not impede the results and contributions of this paper.

2.1.2. Proposed Dataset

As previously mentioned, our aim is to build a dataset for email classification. To this end, we used Label Studio (https://labelstud.io/; accessed on 12 May 2023), an open-source annotation platform for manual annotation. An important detail is that even though the raw data came from Romanian companies, a few emails are not in Romanian. For the purpose of this paper, the emails in the subset selected for the annotation process were all written in Romanian, with the exception that an email could have multiple languages (e.g., words in the message are in English or the message is in Romanian, but the disclaimer in English). An overview of the manual annotation procedure is depicted in Figure 4.

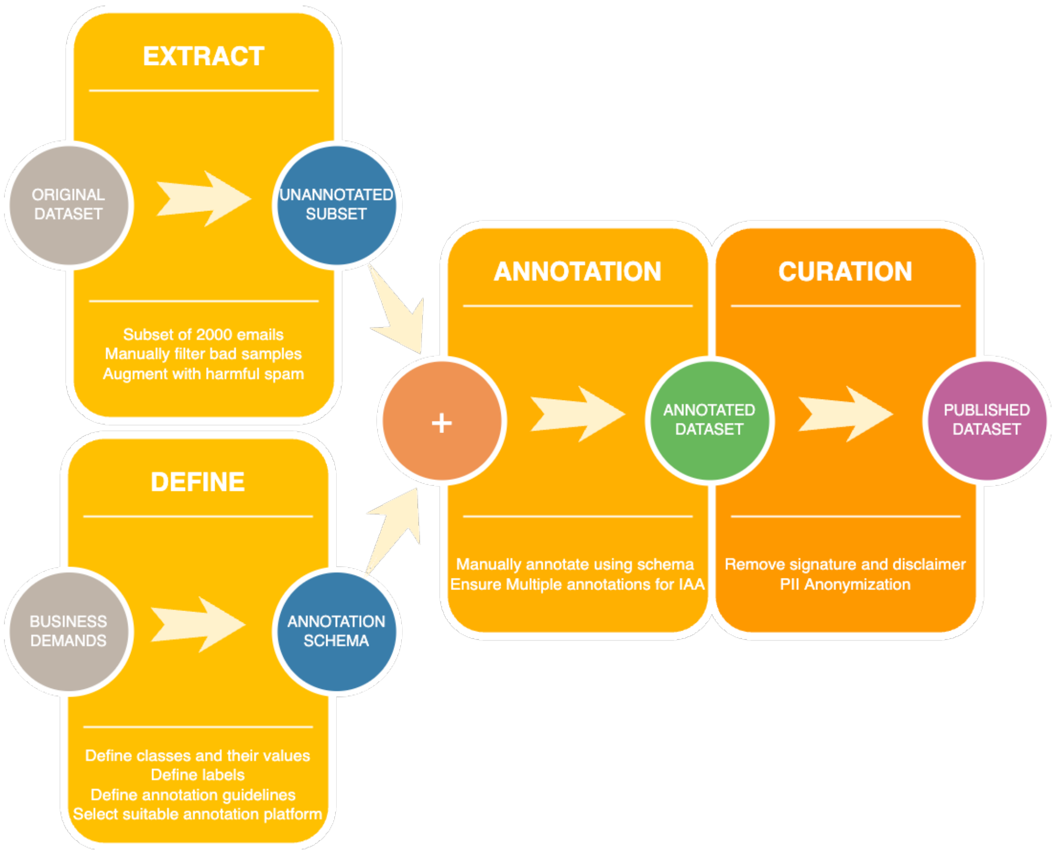

EXTRACT
ORIGINAL DATASET
UNANNOTATED SUBSET
Subset of 2000 emails
Manually filter bad samples
Augment with harmful spam
DEFINE
BUSINESS DEMANDS
ANNOTATION SCHEMA
Define classes and their values
Define labels
Define annotation guidelines
Select suitable annotation platform
+
ANNOTATION
ANNOTATED DATASET
Manually annotate using schema
Ensure Multiple annotations for IAA
CURATION
PUBLISHED DATASET
Remove signature and disclaimer
PII Anonymization

Figure 4. Manual Annotation Procedure Overview.

A subset of 1447 emails was selected for annotation purposes using the following steps:

1. We randomly sampled 2000 of the emails from the original collection;
2. We manually checked all emails and eliminated the ones that were written in a different language than Romanian; if an email was written in multiple languages, out of which one was Romanian, then that email was kept; in case of soft-duplicates, only one message was kept;
3. Lastly, we also added a few selected harmful spam emails retrieved from our mail inboxes to the dataset in order to ensure a higher frequency of these messages.

Unlike the previously mentioned datasets containing annotations on whether an email was business-related or personal, we introduced more annotations (or metrics) that provide value in a business context. Towards this end, we defined two subtasks for the annotation process of the selected emails. Considering the available data in an email (i.e., header, body, and attachments), the first subtask was a multi-task classification in a business context. The classes deemed important are the following:

- Is Automatically Generated:
  - This class has boolean values (i.e., True or False);

- The label is True when it can be deduced that the email was generated by an application (e.g., generated by a language model or generated by filling in personalized data on a predefined template) and False when it was composed by a human user.

- Needs Action from the User:
  - This class has boolean values (i.e., True or False);
  - This label is True if the email specifies the need for the person who received it to perform a specific action; otherwise, the label is False.
- Is SPAM
  - This class has boolean values (i.e., True or False);
  - This label is True if the email can be considered relatively harmless or harmful spam according to the classification made in Section 1. For this annotation task, we considered the following as relatively harmless spam: marketing email (e.g., promotional offers, ads for products, surveys) and newsletters (i.e., news from organizations about their products, services, and offers). Harmful spam emails include phishing, scam, or malspam emails; we opted to combine these two sub-categories since harmful spam had a low frequency in our dataset;
- Is Business-Related
  - This class has boolean values (i.e., True or False).
  - This label is True if the email is about a subject that can be related to the company's activity (e.g., tasks, notifications, equipment, financial transactions, legal matters); otherwise, it is False.
- Type of Writing Style
  - This class has one of the following three values: "Formal", "Neutral", and "Informal".
  - This class specifies the type of language used in the redaction of the email.

An important thing to mention is that the last class is much more subjective than the previous; as such, the opinion of the annotator is more likely to induce bias. The statistics for the multi-task classification can be seen in Figures 5 and 6. We present the number of examples annotated with "True" for the classes with binary values while we show the number of examples annotated with each of the three possible values for the writing style class.

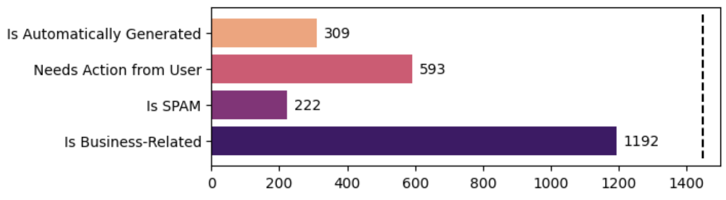


Figure 5. Multi-task Classification—Binary Classes Statistics.

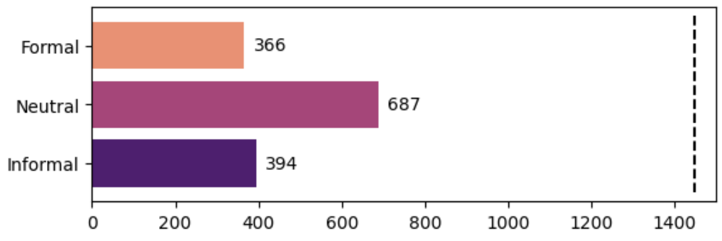


Figure 6. Multi-task Classification—Writing Style Class Statistics.

The second subtask was a token classification one, which deals with the identification of text fragments that could hinder the classifier (i.e., signature, disclaimer), as well as other relevant token information. For this, we defined the following labels:

- Initial Addressing:
  - The selected text fragment represents an initial addressing formulation (e.g., "Salut", "Stimate client"—eng. "Hello", "Esteemed client"). This is generally at the beginning of the message and has the role of initializing the communication act;
- Final Addressing:
  - The selected text fragment represents a final addressing formulation (e.g., "O zi buna!", "Mulțumesc!"—eng. "Have a good day!", "Thank you!"). This is generally at the end of the message and has the role of finalizing the communication act. Typically if there happens to be other text after it, it is usually the signature of the sender or a disclaimer;
- Signature:
  - The selected text fragment represents the signature of the sender. A signature is a fragment of text at the end of the communication, generally between the final addressing and the disclaimer (if both exist) that contains the contact information of the writer. Common information here includes the writer's name, his position in the company, the name of the company, the address of the company, phone number, fax, emails, and other similar information;
- Disclaimer:
  - The selected text fragment represents the disclaimer, a text which usually contains clauses and legal considerations in relation to the email and its content. Usually, only companies and institutions have a disclaimer for the emails sent by their employees. When it appears, the disclaimer is generally the last part of the email body;
- Personal Identifiable Information:
  - The text fragment contains personal information like name, surname, bank accounts, address, and car registration identifiers;
  - Information can only be considered personal if it relates to a person; in other words, a business address is not PII, but the address of "John Doe" is PII.

It is important to mention that each example must have a label for each class for the former subtasks (i.e., multi-task email classification), while any of the previously described labels may be missing for the latter subtasks (i.e., token classification). Moreover, one or more labels can overlap for the latter subtask. For example the fragment "Goodbye Josh and have a great day!" would have the label "Initial Addressing", but "Josh" would also have the label "Personal Identifiable Information". Figure 7 presents the total annotated entities for each label type in the token classification task.

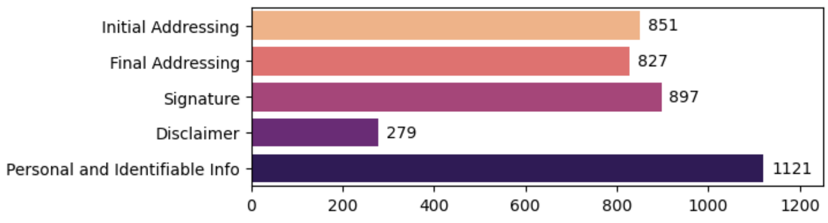
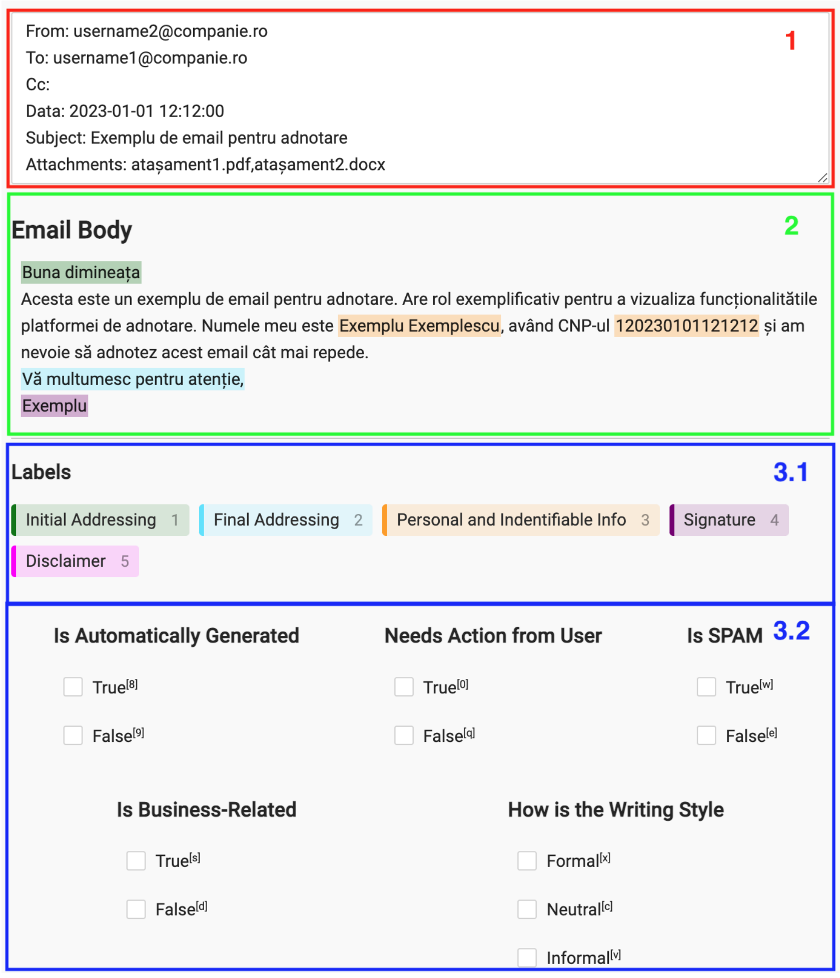


Figure 7. Token Classification—Label Statistics.

With the previous subtask defined, we constructed the annotation interface and started the labeling process. An example of our annotation interface can be viewed in Figure 8. It is separated into three parts. The first part, colored in red (1), contains the header data of the email (i.e., the fields 'from', 'to', and 'subject'). The second part, colored in green (2), contains the body of the email. The third part in blue (3) is split into two sections: the token classification labels and the multi-task classification classes. The workflow of an annotator was to read the header and the body of the email, click on the relevant token classification label, and then select the corresponding text portion. The granularity of the selection was set at the word level.



1

From: username2@companie.ro
To: username1@companie.ro
Cc:
Data: 2023-01-01 12:12:00
Subject: Exemplu de email pentru adnotare
Attachments: atașament1.pdf,atașament2.docx

2

Email Body

Buna dimineața
Acesta este un exemplu de email pentru adnotare. Are rol exemplificativ pentru a vizualiza funcționalitătile platformei de adnotare. Numele meu este Exemplu Exemplescu, având CNP-ul 120230101121212 și am nevoie să adnotez acest email cât mai repede.
Vă multumesc pentru atenție,
Exemplu

3.1

Labels

Initial Addressing 1 | Final Addressing 2 | Personal and Indentifiable Info 3 | Signature 4 | Disclaimer 5

3.2

Is Automatically Generated
☐ True[8]
☐ False[9]

Needs Action from User
☐ True[0]
☐ False[q]

Is SPAM
☐ True[w]
☐ False[e]

Is Business-Related
☐ True[s]
☐ False[d]

How is the Writing Style
☐ Formal[x]
☐ Neutral[c]
☐ Informal[v]

Figure 8. Label Studio Annotation Interface.

The emails were labeled by one of the seven annotators, except for 40 emails which were annotated by all to assess the quality of the annotations. For this, we performed an IAA (Inter-Annotator Agreement) procedure. Two metrics were used to assess IAA for the multi-task classification task, namely Fleiss' kappa [36,37] and Krippendorff's alpha [38,39]. It is generally agreed upon that Fleiss' kappa is the best metric to be used when data is measured using a nominal scale, and there are more than two annotators. Krippendorff alpha usually has the same values as Fleiss' kappa if the data is measured using the nominal scale and has no missing values (i.e., our case). Nonetheless, the latter offers more flexibility since it can also be used for data using the ordinal, interval, and ratio scales. Because of this, it should be a more sound method for the annotations in the "How is the Writing Style" category since those are measured on the ordinal scale. As such, we computed both metrics, with the mention the alpha was computed for the nominal scale in the case of binary classes, while it was computed for the ordinal scale when applied to the "Writing Style" class. The IAA for these annotations is presented in Table 1.

Table 1. Multi-Task Classification—IAA.

Category	Fleiss' Kappa	Krippendorff's Alpha
Is Automatically Generated	0.736	0.737
Needs Action from User	0.383	0.385
Is SPAM	0.734	0.735
Is Business-Related	0.618	0.619
How is the Writing Style	0.367	0.586

Both IAA values fall in the range $[-1, 1]$. Thus, both the values for kappa and alpha can be interpreted according to Table 2. It can be observed that the annotation procedure led to a substantial agreement in almost all classes. The only outliers are "Needs Action from User" and "How is the Writing Style". Regarding the latter, Krippendorff's alpha holds a higher importance due to the fact it is measured using an ordinal scale. Because of this, we can conclude that this label reaches moderate agreement between annotators. This is relatively expected since this class is more subjective than the previous ones. As for the "Needs Action from User" class, this class can also be rather subjective since some emails can implicitly suggest that the user needs to complete an action, which is harder to extract compared to explicit instructions.

Table 2. Kappa (and Alpha) Value Interpretation [40].

Value	Interpretation
<0.00	Poor agreement
0.00 to 0.20	Slight agreement
0.21 to 0.40	Fair agreement
0.41 to 0.60	Moderate agreement
0.61 to 0.80	Substantial agreement
0.81 to 1.00	Almost perfect agreement

For the token classification task, which is similar to a NER (Named Entity Recognition) task, the best metric for assessing the IAA is the pair-wise F1 score [41]. This is because Fleiss' kappa can severely underestimate the true agreement in this case. As such, we computed the pair-wise F1 score between all annotator pairs using the "seqeval" package (https://huggingface.co/spaces/evaluate-metric/seqeval; accessed on 12 May 2023) and then averaged the results to obtain the overall F1 score for each label category. This package is used for computing the accuracy, precision, recall, and F1 score for NER-like tasks. The IAA for these annotations can be seen in Table 3. We can observe that the only label which has a slightly weaker IAA is "Disclaimer". This is because disclaimers can be very long spans of text in general, and we observed that some annotators correctly identified the start of the disclaimer, but they relatively disagreed on where to end. Some emails, like newsletters, contained information on how to unsubscribe, or other useless information for our task after the legal and confidential information in the disclaimer. However, this did not represent an impediment to our tasks, since we are interested in where the signatures and disclaimers start, rather their ending.

Table 3. Token Classification—IAA.

Label	Overall F1 Score
Initial Addressing	0.937
Final Addressing	0.823
Signature	0.687
Disclaimer	0.413
Personal Identifiable Info	0.737

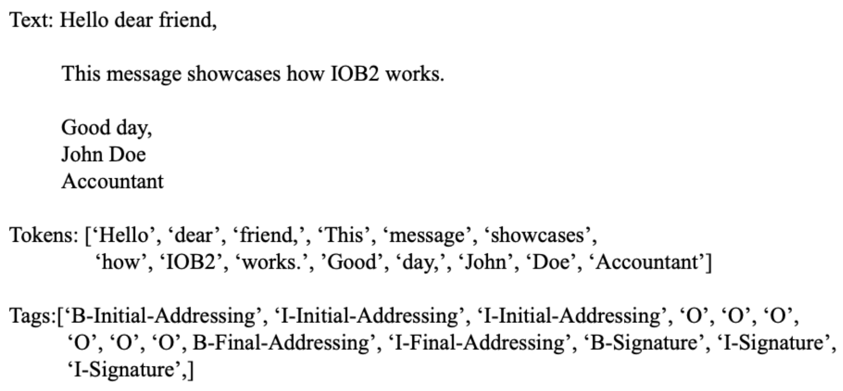
2.2. Email Classification

2.2.1. Curation

The curation of the dataset requires only the useful content of the email, meaning that we had to identify the signatures and disclaimers to eliminate them. To achieve this, we trained an ML model to identify these elements in the text. Since the signatures and the disclaimers have a relatively fixed position in an email (i.e., they are after the final addressing if it appears), we have used the following 4 labels to train our model: “Initial Addressing”, “Final Addressing”, “Signature”, and “Disclaimer”. We also used the two types of addressing because they depend on the span’s placement in the email, thus having a similar bias to the signature and disclaimer. Token classification is a task that requires codifying the labels and the text into an IOB (Inside–Outside–Beginning) format. We considered a slight variation called IOB2 with the following properties:

- The text is split into words (which shall be referred to as tokens).
- Each word has a corresponding tag.
- There are two important tags for each annotated label: the “B” tag marks that the respective token is the first in the corresponding label, while the “I” tag signifies that the respective token is not the first in the labeled sequence.
- All tokens which do not correspond to a label receive the “O” tag.

An example of an annotation with this format is presented in Figure 9.


Text: Hello dear friend,

This message showcases how IOB2 works.

Good day,
John Doe
Accountant

Tokens: [‘Hello’, ‘dear’, ‘friend,’, ‘This’, ‘message’, ‘showcases’,
‘how’, ‘IOB2’, ‘works.’, ’Good’, ‘day,’, ‘John’, ‘Doe’, ‘Accountant’]

Tags:[‘B-Initial-Addressing’, ‘I-Initial-Addressing’, ‘I-Initial-Addressing’, ‘O’, ‘O’, ‘O’,
‘O’, ‘O’, ‘O’, B-Final-Addressing’, ‘I-Final-Addressing’, ‘B-Signature’, ‘I-Signature’,
‘I-Signature’,]

Figure 9. IOB2 scheme example.

The curation step is necessary because signatures and disclaimers can be large blocks of text. Our architecture for this task comprises of a pre-trained Transformer encoder backbone with a token classification head. The token classification head receives each encoded token and applies dropout on them, then passes them through a dense layer, thus obtaining logits for each token. Afterward, these are combined using a CRF (Conditional Random Field) to get the final IOB2 tag for each token. This is similar to the PII head in Figure 10, but here the token classification head deals with all 4 labels and also contains a CRF layer (unlike the PII one).

Since most Transformer encoders have a limit of 512 sub-word tokens, the signatures, and disclaimers would create a significant impediment for the following tasks if they would not have been eliminated. To keep only the important parts of the email, we trimmed it from the start and the end. For the beginning of the email, we removed everything before the initial addressing formulation, if it exists. Few exceptions were encountered—e.g., artifacts from the preprocessing of the raw data. We adopted the following approach for the ending of the email: we chose the smallest index between the first token after the final addressing, the first token of the signature, and the first token of the disclaimer, and afterward trimmed everything starting with that token until the end of the email. We have chosen this approach because there is no certainty with regard to which of the three labels are present (if there are any) in the email. This procedure resulted in two curated datasets, one created from our annotations and one created by using the ML model trained on detecting the 4 labels. With this process, the average length of the Transformer-tokenized emails decreased by a fourth.

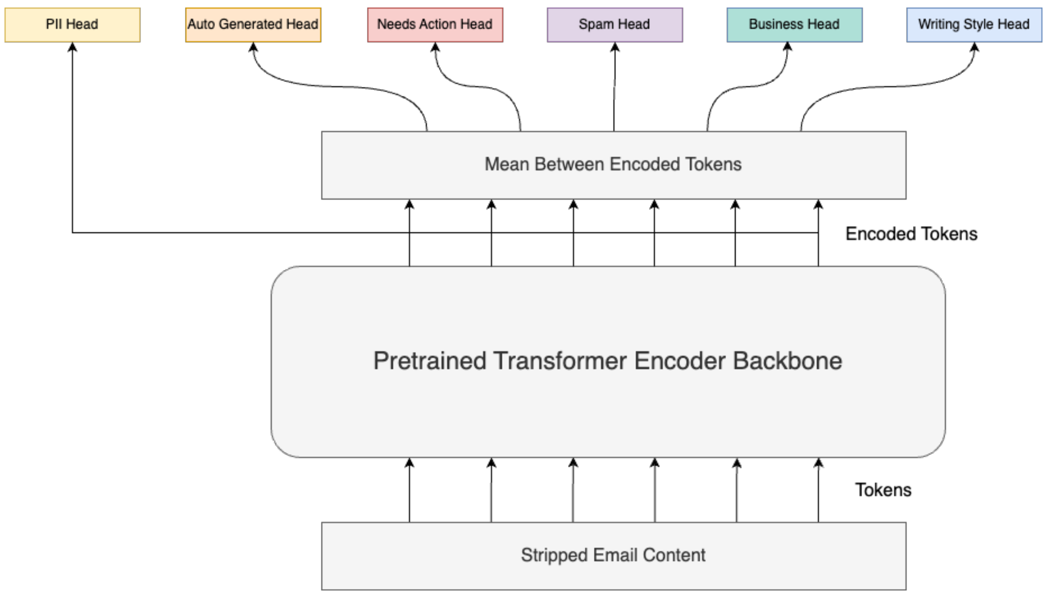
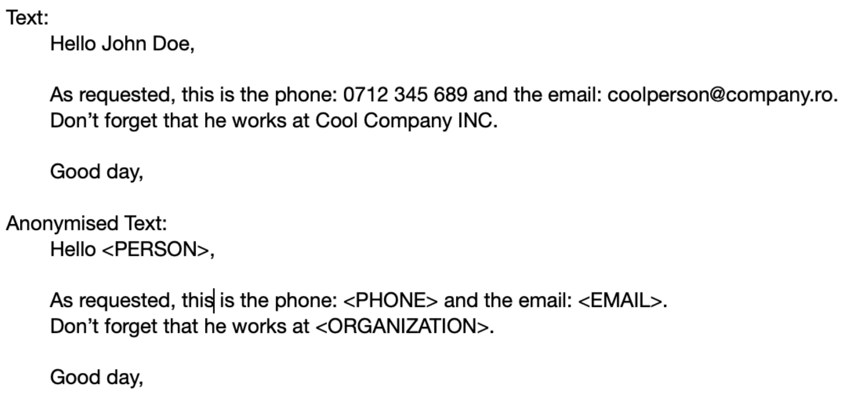


Figure 10. Multi-Task Classification Architecture.

Lastly, we also anonymized the dataset in terms of PII for its publication. To this end, we started from the annotated PII and a ro_core_news_lg model from spaCy (https://spacy.io/models/ro; accessed on 12 May 2023) for NER, and we ran the model on the PII spans. We manually verified all the outputs of the NER model and corrected them when needed so as to ensure that the generated entities corresponded to their respective PII span. Afterward, we replaced the PII spans in the email body with their respective entity. We also removed the original PII labels from the annotations, thus obtaining a body similar to the example in Figure 11. This version of the dataset is the publicly available one, containing only the anonymized email body and the multi-class classification classes. To better assess the size difference in the emails before and after the curation process, we show the average length of the emails (both in words and characters) in Table 4.


Text:

Hello John Doe,

As requested, this is the phone: 0712 345 689 and the email: coolperson@company.ro.
Don't forget that he works at Cool Company INC.

Good day,

Anonymised Text:

Hello <PERSON>,

As requested, this is the phone: <PHONE> and the email: <EMAIL>.
Don't forget that he works at <ORGANIZATION>.

Good day,

Figure 11. Anonymized Email Example.

Table 4. Datasets Size Comparisons.

Type	Dataset Version	Mean	Std	Min	25%	50%	75%	Max
Words	Not Curated	139.72	203.41	1.00	27.00	57.00	160.50	1481.00
	Published	105.17	187.11	1.00	16.00	35.00	97.00	1474.00
Characters	Not Curated	1105.03	1718.37	6.00	187.50	402.00	1182.50	13,508.00
	Published	835.05	1611.14	3.00	98.00	224.00	667.50	13,175.00

2.2.2. Multi-Head Classification

With the curated datasets available, we focused on the email classification task. Here, we trained a multi-task Transformer model (see Figure 10) to predict each of the previously discussed classes, as well as to identify the tokens which form a PII text span. To this end, we opted for a multi-head classification architecture to make the model more robust and less likely to overfit by fine-tuning it on multiple tasks simultaneously. Each classification head uses different weights, but the same input representation which is computed by the backbone. Each classification head has a distinct loss function and to achieve a unified objective function we define the final loss as the sum of all the others.

The Transformer backbone is used in the same manner as in the curation phase but with a more complex classification head. The first part of the classification head is the token classification part (PII Head) which considers each of the encoded tokens outputted by the backbone, applies dropout on them, and finally passes them through a dense layer. Afterward, the final IOB2 tags are obtained using a Softmax function on the logits to obtain the probabilities of the IOB2 encoded labels, followed by an argmax operation. In this case, since we only have one token classification head with only one type of label (PII), we simplified the IOB2 encoding by only leaving 'I' and 'O' tokens, thus replacing the 'B' with 'I'. This simplifies the models' training on this head while also eliminating the need to have a CRF layer.

The second part of the classification head focuses on the text classification tasks. Because of this, we need to combine the representations of each individual token into a representation of the entire input sample. There are two often-used methodologies for this: (a) using the embedding of the first token (e.g., <CLS>) as a representation of the entire input or (b) aggregating all of the embeddings with a function like the arithmetic mean. We chose the latter approach since it takes into account the contribution of every token for the classification decision. Not only that, but it can also be beneficial since some of our classes can depend on the entire context (like "Needs Action from User"). After obtaining the mean (i.e., the condensed representation of the entire input), the embeddings are used by each of the remaining classification heads. Each individual classification head is comprised of a dropout layer followed by a dense layer. We applied a function to convert the logits into the class probabilities to get the result for each head; depending on the head, this can be either a Softmax (for the Writing Style Head) or the Sigmoid function (for the others). Additionally, we used a weighted loss function for the Writing Style Head so that each value would have importance inversely proportional to its frequency.

3. Results

We randomly partitioned the annotated dataset into three splits: training, validation, and test, each split containing 60%, 20%, and 20% of the original dataset, resulting in 868 training, 289 validation, and 290 test examples. We tried to keep the distribution as similar as possible to the unpartitioned version by using the MultilabelStratifiedShuffleSplit class from the "iterative-stratification" Python package (https://pypi.org/project/iterative-stratification/; accessed on 12 May 2023). The algorithm used for stratifying multilabel data implemented in this class was introduced by Sechidis et al. [42]. The distributions for each class are presented in Table 5. Since the dataset was unbalanced for specific tasks (e.g., "Is Automatically Generated", "Disclaimer"), we experimented with a weighted loss strategy computed based on the percentages of each class; this strategy was kept if the overall macro F1 score improved.

Table 5. Dataset Partitions Annotation Distributions.

Annotation Type	Unpartitioned Dataset	Train	Validation	Test
Is Automatically Generated	21.35%	21.42%	20.06%	22.41%
Needs Action from User	40.98%	40.09%	40.83%	43.79%
Is SPAM	15.34%	16.93%	12.80%	13.10%
Is Business-Related	82.37%	81.68%	83.39%	83.44%
How is the Writing Style—Formal	25.29%	26.03%	25.95%	22.41%
How is the Writing Style—Neutral	47.47%	48.61%	44.98%	46.55%
How is the Writing Style—Informal	27.22%	25.34%	29.06%	31.03%
Initial Addressing	58.81%	58.64%	58.82%	59.31%
Final Addressing	57.15%	57.02%	57.43%	57.24%
Signature	61.99%	61.75%	62.28%	62.41%
Disclaimer	19.28%	19.12%	19.37%	19.65%
Personal Identifiable Info	77.47%	73.04%	94.11%	74.13%

For the following experiments, we selected 5 models as the pre-trained backbone encoders:

- Two BERT [43] models pre-trained for the Romanian language, RoBERT [44] (https://huggingface.co/readerbench/RoBERT-large; https://huggingface.co/readerbench/RoBERT-base; accessed on 12 May 2023). This was chosen since an encoder pre-trained in the Romanian language should adequately process text in Romanian.
- Two multilingual RoBERTa models [45] (xlm-roberta-base/large; https://huggingface.co/xlm-roberta-base and https://huggingface.co/xlm-roberta-large; accessed on 12 May 2023). These were chosen since some emails can be written in multiple languages (including Romanian) and because disclaimers and signatures can also be written in English, thus an encoder that can process multiple languages should help.
- A multilingual Longformer model [46] (https://huggingface.co/markussagen/xlm-roberta-longformer-base-4096; accessed on 12 May 2023). Since emails, especially with the signature and disclaimers included, can exceed the normal 512-token limit of encoders, we hypothesized that a model that can process the entire context could improve results compared to the others.

A similar hyperparameter search procedure was implemented for each experiment, namely variating the learning rate between 5×10^{-4} and 5×10^{-6}, as well as variating whether the training set should be reshuffled at each epoch or not. We hypothesized that, since larger examples were generally not truncated but split into continuous smaller examples, this would aid the training process by trying to keep them grouped together.

The first task was the token classification used for the curation of the datasets. Here, about 250 examples had an input bigger than 512 tokens, if tokenized with the Transformer tokenizer from the RoBERT model and the XLM RoBERTa models. Since we opted not to truncate our data, resulting in losing important information, we split our examples into 512 token chunks. For this task, we divided the tokens while considering their labels so that we would split the tokens while a label was not ongoing. In the eventually that a label would be too long to be split in the previously described way, then we would truncate the tokens until the end of that label and continue the next chunk afterward. For the Longformer model, since it could process up to 4096 tokens, we did not split the examples and truncated the longer examples instead. This happened for only three emails, which was a negligible loss. For each model, we experimented with multiple constant learning rates during fine-tuning and saved the best results. Afterward, we experimented with whether adding a CRF layer would improve the results. We fine-tuned the previous best architecture with a final CRF layer, which indeed improved the result, thus our final architecture contained a CRF layer. We also experimented with multiple reduction variants for it, namely: sum, mean, and token-mean. The results of these comparisons are presented in Table 6, while a more detailed overview of the best model can be seen in Table 7. The hyperparameters for the best model were 5×10^{-5} learning rate, shuffling at each epoch, and token-mean. For this task, we used the Precision, Recall, and F1 score computed by the seqeval package.

Table 6. Token Classification—Fine-tuned Models Overall Comparison (bold marks the best results).

Model	F1-Score Train	F1-Score Validation
RoBERT Base	0.750	0.724
RoBERT Large	**0.851**	0.725
XLM RoBERTa Base	0.777	0.704
XLM RoBERTa Large	0.786	**0.739**
XLM Longformer Base	0.547	0.687
XLM RoBERTa Large with CRF(sum)	0.557	0.618
XLM RoBERTa Large with CRF(mean)	0.704	0.735
XLM RoBERTa Large with CRF(token_mean)	**0.909**	**0.765**

Table 7. Token Classification—Best Model Detailed Results.

Label	Metric	Train	Validation	Test
Initial Addressing	Precision	0.982	0.911	0.921
	Recall	0.990	0.901	0.964
	F1	0.986	0.906	0.942
Final Addressing	Precision	0.846	0.698	0.713
	Recall	0.898	0.751	0.793
	F1	0.871	0.723	0.751
Signature	Precision	0.870	0.728	0.628
	Recall	0.961	0.820	0.822
	F1	0.913	0.771	0.712
Disclaimer	Precision	0.69	0.392	0.260
	Recall	0.776	0.465	0.422
	F1	0.732	0.425	0.322
Metric		**Train**	**Validation**	**Test**
Overall Precision		0.882	0.743	0.690
Overall Recall		0.937	0.797	0.825
Overall F1		0.909	0.765	0.752

Afterward, we created two versions of the dataset, one which was curated manually using the annotations and the other using the best model of the previously fine-tuned models, namely XLM Roberta Large with a CRF layer and token_mean reduction. We refer to the former dataset as MCD (Manually Curated Dataset), whereas the latter is ACD (Automatically Curated Dataset). Using these two datasets, we used the same 5 back-bones with the previously described multi-head approach. For this task, we changed the chunking procedure by ignoring the labels and assuring a 128-token overlap between the chunks. We chose this approach because, in a real-world scenario, we do not have annotated labels; as such, we cannot chunk with respect to them. Consequently, we used the model output to recombine the chunks into one annotation and utilized the combined examples for evaluation. We fine-tuned the models first using the MCD and with the ACD augmented with the MCD for the training split. Essentially, we combined both training splits into one and validated/tested them with the ACD splits. We compared the two variants of the models on the ACD to simulate the real-world scenario. Here, we experimented with adding a CRF layer for the PII labels, as well as with a weighted loss function for each task. The results are displayed in Tables 8 and 9. For the best model from Table 8, the best hyperparameters were: 5×10^{-5} learning rate, without reshuffling, and with a weighted loss for the writing style category.

Table 8. Multi-Task Classification—Fine-tuned Models Overall Comparison (bold marks the best results).

Model	F1 Train	F1 Validation MCD	F1 Validation ACD
RoBERT Base	0.850	0.766	-
RoBERT Large	0.831	0.770	-
XLM RoBERTa Base	0.805	**0.785**	**0.753**
XLM RoBERTa Large	0.798	0.740	-
XLM Longformer Base	**0.900**	0.750	-
XLM RoBERTa Base	0.826	0.746	0.743

Table 9. Multi-Task Classification—Best Model Detailed Results.

Task	Train	F1 Validation ACD	F1 Test ACD
Personal Identifiable Information	0.353	0.185	0.237
Is Automatically Generated	0.861	0.807	0.783
Needs Action from User	0.787	0.740	0.685
Is SPAM	0.909	0.821	0.857
Is Business-Related	0.937	0.928	0.936
How is the Writing Style	0.529	0.467	0.449

Lastly, we anonymized the MCD as previously described in Section 2.2, thus obtaining the publicly available dataset. Afterward, we fine-tuned the 5 backbone models for multi-task classification modifying the architecture by eliminating the PII head since the email had been anonymized. Here, we tried the same variations as with the previous experiment except for CRF since we did not have any token classification for this task. The results on the publicly available dataset are displayed in Tables 10 and 11. The hyperparameters for the best model were: 2.5×10^{-5} learning rate, without reshuffling and with the weighted loss on the writing style category.

Table 10. Public Dataset Results—Models Comparison (bold marks the best results).

Model	F1 Train	F1 Validation	F1 Test
RoBERT Base	0.774	0.763	-
RoBERT Large	0.800	0.755	-
XLM RoBERTa Base	0.809	**0.770**	0.764
XLM RoBERTa Large	0.685	0.658	-
XLM Longformer Base	**0.900**	0.750	-

Table 11. Public Dataset Results—Best Model Detailed Results.

Task	Train	F1 Validation	F1 Test
Is Automatically Generated	0.817	0.785	0.820
Needs Action from User	0.811	0.796	0.739
Is SPAM	0.917	0.828	0.873
Is Business-Related	0.938	0.920	0.927
How is the Writing Style	0.561	0.524	0.462

4. Discussion

First, we focus on the results of the token classification task for the automated dataset curation. It is important to note when inspecting Table 6 that even though the XLM Longformer model could process the input examples without the need for chucking, inherently being capable of predicting the labels in a matter more similar to how the human annotators approached this task, it obtained by far the worst result for this task. This suggests that the chunking process did not impede the token classification task, meaning the model does not need the entire context to learn to predict the labels, unlike how a human annotator may need it to make an accurate annotation. Considering the results in Tables 3 and 7, the model also struggled to identify the "Disclaimer" label because the IAA for it was much weaker than for the other labels. However, considering the results in Table 8, it is highly probable that our original observation and intuition in regards to the fact that annotators disagreed on where to end the "Disclaimer" as opposed to mislabeling an important part of the body as a disclaimer was correct. The other labels with a higher IAA also achieved better results on the classification task.

Regarding the second task, it is interesting to observe that the best backbone for the previous task obtained the worst results on this one. We speculate that this is because the curated dataset contains fewer English words; as such, a multilingual model is less effective. Also, in the case of classification tasks, the number of training signals is much lower than for token classification (one per email compared with one per token), so smaller models can be more effective. In addition, as previously mentioned, it is important to note that the automated curation slightly affected the results; however, the models obtained a relatively similar result on the test set. This confirms that the trimmed results still provide enough valuable information, despite the automated curation differing slightly from the manual one.

Considering the results in Tables 1 and 9, we observe that the model struggled to predict the correct spans for PII that had a higher IAA than other labels (e.g., Disclaimer). Whilst it may be possible for a model to consider the entity type (NER) of the tokens in order to achieve a better result for PII identification, it seems unlikely that increasing the IAA on this task could increase performance. In contrast, the performance for writing style was most likely influenced by the disagreement among annotators. Stricter guidelines and a reannotation may increase the model's performance by a large margin for this task.

The results on the publicly available dataset partially illustrate the previously mentioned point. The anonymization essentially replaced the spans in the text with their NER-like counterpart. This resulted in a considerable upgrade to all three models, RoBERT large being the only one negatively impacted by this change, mostly caused by the NER-like tags being in English. We observe by comparing Tables 8 and 10 that the introduction of the NER-like tokens had a generally positive impact.

5. Conclusions and Future Work

This paper introduced a novel dataset and baseline models for automatic email curation and classification. The dataset and the source code for the email classification model were publicly released. Our dataset generally has a moderate to substantial inter-annotator agreement (0.57 kappa and 0.61 alpha), indicating that it can be successfully used for training Machine Learning models. The automatic curation model achieved good results (0.752 F1-score), and it is even more effective in practice because of the way in which the text is removed until the end of the email. The email classification model also achieved good results, while anonymizing the dataset using NER-like tokens slightly improved the average results on the remaining classes. The F1-score achieved on the published dataset is 0.764.

In terms of future research, one direction is to improve the dataset by establishing stricter guidelines for classes such as "How is the Writing Style"; increasing the sample size with better IAA would ensure the generalizability of our model. We also aim to add other classes to our dataset, including "Is Harmful SPAM", for which we already started collecting examples. Moreover, we are looking into using the proposed dataset and the unannotated data (i.e., the original 400,000 processed emails) to improve our models in a semi-supervised manner.

Author Contributions: Conceptualization, A.D., S.R. and M.D.; methodology, A.D. and S.R.; software, A.D.; validation, A.D., S.R. and M.D.; formal analysis, A.D.; investigation, A.D.; resources, C.K.B. and M.D.; data curation, A.D. and D.I.; writing—original draft preparation, A.D.; writing—review and editing, S.R., D.I. and M.D.; visualization, A.D.; supervision, S.R. and M.D.; project administration, M.D. and C.K.B.; funding acquisition, C.K.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the "Innovative Solution for Optimizing User Productivity through Multi-Modal Monitoring of Activity and Profiles—OPTIMIZE"/"Soluție Inovativă de Optimizare a Productivității Utilizatorilor prin Monitorizarea Multi-Modala a Activității și a Profilelor—OPTIMIZE" project, Contract number 366/390042/27.09.2021, MySMIS code: 121491.

Institutional Review Board Statement: Ethical review and approval were not required for this study as it solely involved the analysis of business-related information available in an enterprise environment. As regards the use of sensitive information, informed consent was obtained to use the data, and all identified sensitive data from the corpus was removed.

Informed Consent Statement: The data utilized in the present study was obtained and subjected to processing with the informed consent of companies' proprietors.

Data Availability Statement: The dataset can be found at https://huggingface.co/datasets/readerbench/ro-business-emails (accessed on 12 May 2023), while the code used for training the multi-task classification model on the published dataset can be found at https://github.com/research-technology-ai/ro-business-emails (accessed on 12 May 2023).

Acknowledgments: We would like to thank Teodor Neacsu, Cosmin Oprescu, Andreea Dutulescu, Emanuel Tertes, Pavel Betiu, and Bogdan Gutu for their support in building the corpus used in our experiments.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
CRF	Conditional Random Field
CSS	Cascading Style Sheets
DOM	Document Object Model
GA	Genetic Algorithms
GCNs	Graph Convolutional Networks
GDPR	General Data Protection Regulation
GRU	Gated Recurrent Unit
HTML	HyperText Markup Language
IAA	Inter-Annotator Agreement
IOB	Inside–Outside–Beginning
PII	Personal Identifiable Information
KNN	K-Nearest Neighbour
LDA	Latent Dirichlet Allocation
LSTM	Long-Short Term Memory
MCD	Manually Curated Dataset
ML	Machine Learning
MLP	Multi-Layer Perceptron
NER	Named Entity Recognition
NMF	Non-negative Matrix Factorization
NSA	Negative Selection Algorithm
PSO	Particle Swarm Optimization
RCNN	Region-Based Convolutional Neural Network
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF-IF	(Term Frequency—Inverse Document Frequency

References

1. Klimt, B.; Yang, Y. The enron corpus: A new dataset for email classification research. In Proceedings of the Machine Learning: ECML 2004: 15th European Conference on Machine Learning, Pisa, Italy, 20–24 September 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 217–226.
2. Srirag, D. Text Classification on Emails. 2020. Available online: https://www.kaggle.com/datasets/dipankarsrirag/topic-modelling-on-emails (accessed on 3 May 2023).
3. Jabbari, S.; Allison, B.; Guthrie, D.; Guthrie, L. Towards the Orwellian nightmare: Separation of business and personal emails. In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, Sydney, Australia, 17–18 July 2006; pp. 407–411.
4. Alkhereyf, S.; Rambow, O. Work hard, play hard: Email classification on the Avocado and Enron corpora. In Proceedings of the TextGraphs-11: The Workshop on Graph-Based Methods for Natural Language Processing, Vancouver, BC, Canada, 3 August 2017; pp. 57–65.
5. Oard, D.; Webber, W.; Kirsch, D.; Golitsynskiy, S. *Avocado Research Email Collection*; Linguistic Data Consortium: Philadelphia, PA, USA, 2015.
6. Mason, J. The Apache SpamAssassin Public Corpus. Available online: https://spamassassin.apache.org/old/publiccorpus/ (accessed on 3 May 2023)
7. Dua, D.; Graff, C. UCI Machine Learning Repository. 2017. Available online: http://archive.ics.uci.edu/ml (accessed on 2 May 2023).
8. Metsis, V.; Androutsopoulos, I.; Paliouras, G. Spam filtering with naive bayes-which naive bayes? In Proceedings of the CEAS, Mountain View, CA, USA, 27–28 July 2006; Volume 17, pp. 28–69.
9. Cormack, G.V.; Lynam, T.R. TREC 2005 Spam Track Overview. In Proceedings of the TREC, Gaithersburg, MD, USA, 15–18 November 2005; pp. 274–500.
10. Nazario, J. Phishing Corpus. 2007. Available online: http://monkey.org/~jose/wiki/doku.php (accessed on 12 May 2023).
11. Radev, D. CLAIR Collection of Fraud Email. 2008. Available online: http://aclweb.org/aclwiki (accessed on 12 May 2023).
12. Alghoul, A.; Al Ajrami, S.; Al Jarousha, G.; Harb, G.; Abu-Naser, S.S. Email classification using artificial neural network. *Int. J. Acad. Eng. Res.* **2018**, *2*, 8–14.
13. Li, W.; Meng, W.; Tan, Z.; Xiang, Y. Design of multi-view based email classification for IoT systems via semi-supervised learning. *J. Netw. Comput. Appl.* **2019**, *128*, 56–63. [CrossRef]
14. Sharaff, A.; Gupta, H. Extra-tree classifier with metaheuristics approach for email classification. In Proceedings of the Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018, Bangkok, Thailand, 20–21 October 2018; pp. 189–197.
15. Pan, W.; Li, J.; Gao, L.; Yue, L.; Yang, Y.; Deng, L.; Deng, C. Semantic graph neural network: A conversion from spam email classification to graph classification. *Sci. Program.* **2022**, *2022*, 1–8. [CrossRef]
16. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent dirichlet allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
17. Saleh, A.J.; Karim, A.; Shanmugam, B.; Azam, S.; Kannoorpatti, K.; Jonkman, M.; Boer, F.D. An intelligent spam detection model based on artificial immune system. *Information* **2019**, *10*, 209. [CrossRef]
18. Forrest, S.; Perelson, A.S.; Allen, L.; Cherukuri, R. Self-nonself discrimination in a computer. In Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 16–18 May 1994; pp. 202–212.
19. Yasin, A.; Abuhasan, A. An intelligent classification model for phishing email detection. *arXiv* **2016**, arXiv:1608.02196.
20. Niu, W.; Zhang, X.; Yang, G.; Ma, Z.; Zhuo, Z. Phishing emails detection using CS-SVM. In Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou, China, 12–15 December 2017; pp. 1054–1059.
21. Egozi, G.; Verma, R. Phishing email detection using robust nlp techniques. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 7–12.
22. Harikrishnan, N.; Vinayakumar, R.; Soman, K. A machine learning approach towards phishing email detection. In Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA AP), Tempe, AZ, USA, 21 Match 2018; Volume 2013, pp. 455–468.
23. Fang, Y.; Zhang, C.; Huang, C.; Liu, L.; Yang, Y. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. *IEEE Access* **2019**, *7*, 56329–56340. [CrossRef]
24. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
25. Alhogail, A.; Alsabih, A. Applying machine learning and natural language processing to detect phishing email. *Comput. Secur.* **2021**, *110*, 102414. [CrossRef]
26. Baccouche, A.; Ahmed, S.; Sierra-Sosa, D.; Elmaghraby, A. Malicious text identification: Deep learning from public comments and emails. *Information* **2020**, *11*, 312. [CrossRef]
27. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
28. Harsha Kadam, S.; Paniskaki, K. Text Analysis for Email Multi Label Classification. Master's Thesis, University of Gothenburg, Gothenburg, Sweden, 2020.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.

30. Sharaff, A.; Nagwani, N.K. ML-EC2: An algorithm for multi-label email classification using clustering. *Int. J. Web-Based Learn. Teach. Technol. (IJWLTT)* **2020**, *15*, 19–33. [CrossRef]
31. Jlailaty, D.; Grigori, D.; Belhajjame, K. Business process instances discovery from email logs. In Proceedings of the 2017 IEEE International Conference on Services Computing (SCC), Honolulu, HI, USA, 25–30 June 2017; pp. 19–26.
32. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
33. Alkhereyf, S.; Rambow, O. Email classification incorporating social networks and thread structure. In Proceedings of the Twelfth Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 1336–1345.
34. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
35. Šošić, M.; Graovac, J. Effective Methods for Email Classification: Is it a Business or Personal Email? *Comput. Sci. Inf. Syst.* **2022**, *19*, 1155–1175. [CrossRef]
36. Fleiss, J.L. Measuring nominal scale agreement among many raters. *Psychol. Bull.* **1971**, *76*, 378. [CrossRef]
37. Fleiss, J.L.; Levin, B.; Paik, M.C. *Statistical Methods for Rates and Proportions*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
38. Krippendorff, K. *Content Analysis: An Introduction to Its Methodology*; Sage Publications: Thousand Oaks, CA, USA, 2018.
39. Krippendorff, K. Computing Krippendorff's Alpha-Reliability. *Computing* **2011**, *1*, 25–2011.
40. Landis, J.R.; Koch, G.G. The measurement of observer agreement for categorical data. *Biometrics* **1977**, *33*, 159–174. [CrossRef] [PubMed]
41. Brandsen, A.; Verberne, S.; Lambers, K.; Wansleeben, M.; Calzolari, N.; Béchet, F.; Blache, P.; Choukri, K.; Cieri, C.; Declerck, T.; et al. Creating a dataset for named entity recognition in the archaeology domain. In Proceedings of the Conference Proceedings LREC 2020, Marseille, France, 11–16 May 2020; The European Language Resources Association: Luxembourg, 2020; pp. 4573–4577.
42. Sechidis, K.; Tsoumakas, G.; Vlahavas, I. On the stratification of multi-label data. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, 5–9 September 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 145–158.
43. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
44. Masala, M.; Ruseti, S.; Dascalu, M. Robert–a romanian bert model. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 6626–6637.
45. Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *arXiv* **2019**, arXiv:1911.02116.
46. Sagen, M. Large-Context Question Answering with Cross-Lingual Transfer. Master's Thesis, Uppsala University, Department of Information Technology, Uppsala, Sweden, 2021.