Three Approaches to Client Email Topic Classification

Branislava Šandrih Todorović, Katarina Josipović
NLB DigIT / Serbia
{branislava.sandrih.todorovic, katarina.kovacevic}@nlbdigit.rs
Jurij Kodre
NLB d.d. / Slovenia
jurij.kodre@nlb.si

Abstract

This paper describes a use case that was implemented and is currently running in production at the Nova Ljubljanska Banka, that involves classifying incoming client emails in the Slovenian language according to their topics and priorities. Since the proposed approach relies only on the Named Entity Recogniser (NER) of personal names as a language-dependent resource (for the purpose of anonymisation), that is the only prerequisite for applying the approach to any other language.

1 Introduction

Together with Nova Ljubljanska Banka's (NLB) Centre of Excellence, Belgrade IT company **NLB DigIT** has a mission to incorporate smart, data-driven IT solutions to various aspects of everyday work in different Bank's business sectors. One such case is the classification of client emails sent to the Bank's Contact Centre (*Kontakt Centar* in Slovenian, dubbed KC) with respect to their topic (e.g. accounts/loans/cards, etc.) and priority (high, low, medium).

In this paper we present the whole procedure, from having only the plain Outlook files to the models assigning topics and priorities to emails in real time. Section 2 mentions some of the previously published scientific articles that inspired this research. We propose and discuss three different approaches for the modelling of emails in Section 3. Afterwards, we explain the process of preparing the dataset in Section 4: selection of optimal classification schemes and manual annotation, followed by certain cleaning steps and anonymisation of personal information. Having the prepared dataset, we trained and exhaustively evaluated different NLP models for the case of topic classification. We explain the training process in more detail in Section 5, where we also show evaluation results first on a validation and then on a separate test set. Finally, we close with some final remarks, ideas for further improvement and conclusions in Section 6.

2 Related Work

The problem of email classification has been an active area of research for several decades, with numerous studies focusing on developing effective algorithms to accurately categorise incoming emails based on their content. Researchers still experiment with different techniques and approaches in order to improve the existing SPAM and Phishing email classifiers. Iqbal and Khan (2022) achieved the 98.06% accuracy using the binary Support Vector Machine (SVM) classifier for the case of SPAM, whilst Shuaib et al. (2018) developed the optimal SPAM classifier using Rotation Forest algorithm, achieving the accuracy of 94.2%. Ali et al. (2021) experimented with feature engineering and RNN/CNN architectures, concluding that RNN provides the highest 94.9% accuracy. The binary SVM classifier also proved to be the best Phishing email detector for Sundararaj and Kul (2021), with 87.85% accuracy. SVM proved to be the optimal classifier in all our experiments, which we clarify in the coming sections.

3 Methodology

In this section we will describe three different approaches that we hypothesised to be appropriate for the multi-class email topic classification:

3.1 BERT "all-in-one" approach

BERT (Devlin et al., 2019) is a pre-trained deep learning model developed by Google for natural language processing applications such as question answering and language inference. The model works by training on a massive dataset of text,

Proceedings of Recent Advances in Natural Language Processing, pages 1015–1022
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_109

learning the relationships between words and their meanings. Once pre-trained, BERT model can be fine-tuned on a specific task using a smaller dataset. This step allows the model to adapt to the specific task and improve its performance.

The BERT "all-in-one" approach refers to training each of the models on all samples at once by fine-tuning an off-the-shelf BERT language model for the Slovenian language. Having the same general pre-processing pipeline for all cases of email classification (described in Subsection 4), we propose to fine-tune the SloBERTa (Ulčar and Robnik-Šikonja, 2021) model for the Slovenian language on the whole dataset. The first step is to ensure that only Slovenian emails are present in the dataset, using an off-the-shelf language detection tool. After performing a pre-defined text processing procedure, two different models are to be trained separately: TOPICSLOBERTA (for the topic classification) and PRIORSLOBERTA (for the priority classification). The same would hold for any other BERT model.

There are certain drawbacks with this method. First, the performance of the final model depends on the underlying BERT model. If the BERT model itself is trained on data that comes from a domain that very much differs from the lexica of client emails, one cannot expect too much from the classification outcome. Fine-tuning of BERT models demands all class labels to be well covered in the means of number of representative samples, and to be well balanced, which represents one potential issue with client emails. Additionally, despite there not being any official lower boundary in the means of number of instances on which a BERT model should be fine-tuned, it is well known that for the case of deep neural network models holds the "more the merrier" rule, which could also be one of the performance limitations in our case. Finally, from the technical point of view, fine-tuning of this model demands strong computing resources, which could also represent one of the reasons against using this approach.

3.2 Waterfall-1 approach

Topics of the client emails to the KC are not uniformly distributed: clients commonly ask questions about their accounts, cards, mobile banking application, and less frequently they refer to KC regarding loans. This results in class imbalance. Similarly, the more classes there are in the multi-classification setup, the harder the problem is. This especially holds if the topics are related, which strongly holds in this case.

One peculiar case is with emails in which clients report phishing attempts. These emails contain completely different vocabulary from the regular account- or card-based queries. Hence, we propose the WATERFALL-1 approach, displayed in Figure 1, whose fundamental idea is to separately train a classifier for the dominant classes (dubbed as "major") in the dataset from a classifier trained on less frequent classes (dubbed as "minor"). The whole procedure is illustrated in the process of inference. First, as in the previous method, only emails written in the Slovenian language should be taken into consideration, ensured by the LANG-DETECT component. Since emails that report phishing attempts can be easily differentiated from other categories (due to the different vocabulary) the next step in the procedure is retrieving prediction from the KCPHI, binary classifier that identifies such emails. If such an email is detected, the inference ends there. If this was not the case, the model checks whether an email belongs to any of the major categories, predicted by the KCMAJOR. This classifier is still multi-class, but theoretically, since it would be trained on a smaller number of balanced classes, it should be more reliable. If any of the major classes is predicted, the inference ends there. This classifier has (number of major classes + 1) outputs, where this additional class represents emails from other, non-major categories. If this additional class is predicted, then the inference is pushed to the bottom of the approach, where the KCMINOR MODEL tries to determine which of the minor class labels it should assign to the input email. This classifier also outputs the *Other/Can't decide* category which covers samples that were not classified in any of the major or minor categories.

It is also important to note that putting KCMAJOR classifier above the KCMINOR statistically gives higher probabilities to the more frequent classes. We also propose to include the *SPAM* category in the KCMINOR step. Despite all email servers having spam filters nowadays, some junk still arrives to the inbox. This case is not that common, but since it is still possible, we propose to add it to the bottom classifier as an additional SPAM-detector. One potential drawback of this approach, however, is that spam emails can be in any language, and this procedure would immediately dis-

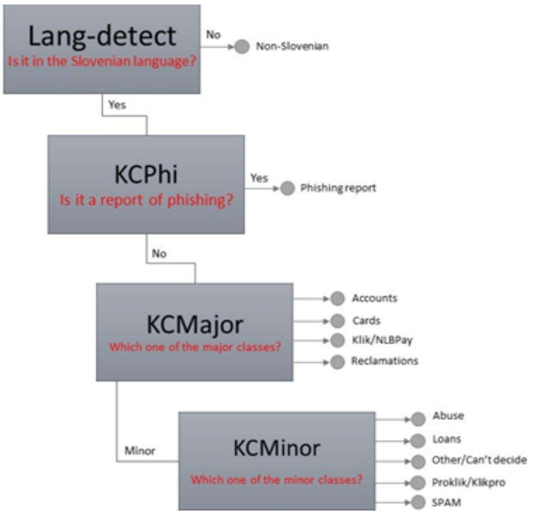

Figure 1: WATERFALL-1 approach

card them as foreign ones. If in practice these emails would get forwarded to KC staff that deal with non-Slovenian emails, they would potentially be the ones receiving junk mail from time to time.

3.3 Waterfall-2 approach

Another perspective to put at multi-class classification task is to divide it into smaller wholes, giving priority to the classes that are of higher importance to be dealt with. As opposed to the WATERFALL-1 approach, in the WATERFALL-2 MODEL (shown in Figure 2), SPAM filter is configured up as a zero-layer. This should fix the WATERFALL-1's SPAM-related potential issue. However, this would demand having a descent amount of SPAM emails to train a satisfactorily performing binary classifier. Afterwards, as in the WATERFALL-1, language detection is performed. Next, KCPHI classifier checks whether an email reports phishing. If this is not the case, KCABUSE classifier checks whether the email reports abuse of an account, card, or mobile banking application. This classifier is put high in the inference process since these emails have the highest priority. Similarly, if abuse is not detected, KCRECLAMATIONS checks whether a client communicates a reclamation. Next steps are the same as in the WATERFALL-1 approach, having KCMAJOR followed by the KCMINOR multi-class classifiers. Yet, the number of classes for both classifiers is smaller than in the WATERFALL-1, since three categories (spam, abuse, reclamations) were given higher priorities by being escalated to the top.

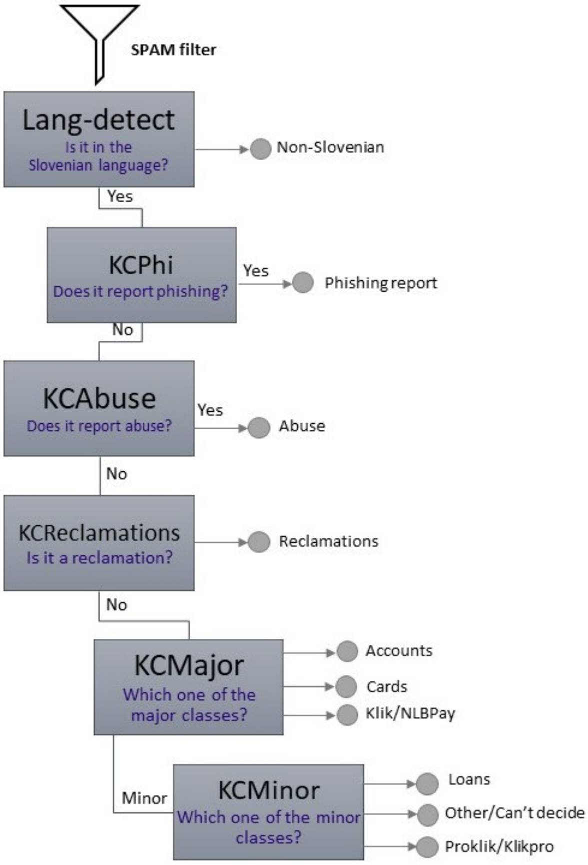

Figure 2: WATERFALL-2 approach

This approach breaks the large classification problem into smaller bits, and consequently, instead of training one large multi-classifier on imbalanced data, it trains several binary or multi-class classifiers on less training samples, but better balanced. Despite this approach being theoretically the most promising, the strongest drawback is a need for many data samples: the more training data for each of the components separately, the better. For the sake of being as reliable as possible, each classifier needs to see many positive instances, but also many negative ones.

As an example, let's observe a zero-layer binary SPAM classifier trained on training samples annotated as SPAM. Since there is already a SPAM filter in any email server, we would not expect too many of such emails in the training set. If we want a balanced sub-sample, if there are n emails annotated as SPAM in the original dataset, we would sample exactly (or nearly) n negative samples from the remaining set of class labels. Regardless of the distribution of the sampled negative data (did we sample the same number of samples for each of the remaining classes or we followed the natural distribution of classes), the classifier potentially would not be able to see enough negative samples to be

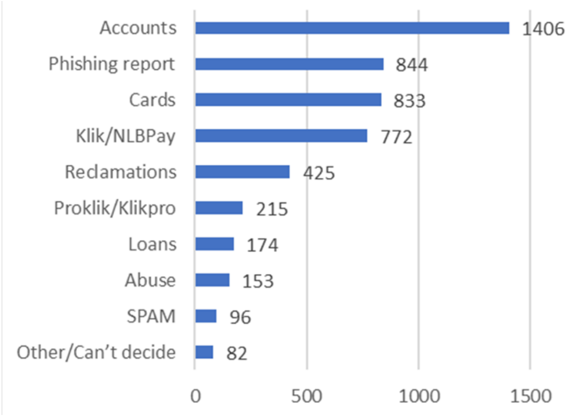

Figure 3: Final topic distribution

able to generalise well enough. As a result, the prediction would result in many false positives, because of the lack of negative instances seen during the training phase.

4 Dataset

It was first necessary to define the annotation schema and the data cleaning process. After taking into account the internal schema KC used over time when archiving finished email correspondences with clients into a database, and agreeing that the priority should directly follow from the topic, we settled on the 10 topics shown in Table 1: high priority Reclamations (1), Phishing Report (3), and Abuse (6); low priority SPAM (2), Other/Can't decide (5); medium priority Cards (4), Accounts (7), Klik/NLBPay (8), Loans (9), Proklik/Klikpro (10).[1] After manual annotation following the guidelines, the final dataset had the topic distribution shown in Figure 3.

Before using textual data for any machine-learning-related task, certain pre-processing steps should be performed. What has turned out to be a very beneficial in practice is to reduce the vocabulary of the text collection as much as possible, as long it does not affect the semantics of the content. Let us observe an email given in Table 2. Values X, Y, W and Z are displayed instead of personal names.

Only segments that were not struck out (subject and middle of the body) contain the client's query, while the rest is either some generic content (generated by the NLB's mail server or by the user's mobile email application). Similarly, personal information such as addresses, full names, mobile phone numbers, PIN, and account numbers represent sensitive information, yet do not in any way influence classification predictions.

On the email body concatenated to its subject, pre-processing procedure consists of the following stages: 1) removal of generic content; 2) tokenisation/the first anonymisation (masking personal names)/ lemmatisation; 3) the second anonymisation (masking email addresses, URLs) and final clean.

After step 1 there is no more generic content (e.g., *Sent from my iPhone*). In the 2^{nd} step, whenever possible, words are replaced by their lemmas. Simultaneously, personal names are replaced with a predefined token "Janez", and only words comprising of alphanumeric characters are kept. This was done using the CONLL-U format outputted by the classla Python library,[2]. For every sentence segmented from the input text provided to the classla's processor, a verticalised list of tokens is given. In each row, there are 10 columns. For our needs, we used the third column that contains lemma of the original token, and the last column that contains information about a recognised named entity, if that was the case. During this second step of the cleaning procedure, we kept only the non-punctuation tokens, simultaneously replacing every token with its lemma. Special case is when a token is recognized as a personal named entity, what we treat by replacing the original name with the common Slovenian name "Janez."

Finally, after the 3^{rd} step, all sequences of numbers were masked with the word "num", and using the scrubadub Python library[3] remaining sensitive information was also masked by corresponding tokens predefined by the library's configuration. The reduced vocabulary of the final, processed dataset of emails was 13,943.

5 Model training and evaluation

In this section we will describe the conducted experiments on the dataset of prepared emails, using the approaches proposed in Section 3.

5.1 BERT "all-in-one" approach

Idea was to fine-tune the existing BERT language model for Slovenian on the whole dataset of emails and teach it how to classify them. We first wanted

[1] To maintain client privacy and the bank's confidentiality, both the dataset and the code are not accessible to the public.

[2] classla Python library, https://pypi.org/project/classla/

[3] scrubadub, https://pypi.org/project/scrubadub/

C	Description	Examples
1	Re-payment required	Danes sem prijatelju nakazal denar preko flikaki pa letega ni prejel, meni pa je na računu trgalo denar 999,99 eur. Lepo prosim za informacijo kaj se v takem primeru zgodi.
2	Advertising, junk	XXX Vas poziva na intenzivni, jednodnevni edukacijski program.
3	Attempts of phishing	Dobil sem sporočilo na mail, da je moj spletni račun začasno zaklenjen zaradi nenavadne dejavnosti. Zanima me kaj je to?
4	Card-related matters	Zanima me koliko drazje je ce uzamem namesto mastercard, visa kartico?
5	General matters	V prilogi vam pošiljam svojo prijavo na prosto delovno mesto svetovalke kontaktnega centra.
6	Reporting abuse	Včeraj sem izgubila denarnico z mojo bančno kartico. Prosim blokirajte moj bančni račun od danes na dalje.
7	Account-related matters	Dobro jutro. Pošiljam zahtevo za ukinitev bančnega računa.
8	Klik/NLBPay apps-related	Prosim za podatke za ponovno aktivacijo klikina.
9	Loans-related matters	Prosim ce mi javite nov znesek obroka kredita po novi obrestni meri.
10	Proklik/Klikpro apps-related	Podjetje bi na Proklik za pregledovanje pooblastili zaposleno.

Table 1: Annotation guidelines

Vprašanje - NLB klik-in, zgubljen denar
~~External mail: Do not open links and attachments in case of unknown sender or suspicious content.~~ Sem X Y, Vaša uporabnica in imam eno kratko vprašanje. Namreč, sem poslala 22 evrov preko NLB klikin aplikacije gospe W Z. Danes mi je napisala, da plačila ni prejela. V upanju, da boste odgovorili ter pomagali v iskanju rešitve, LP, X Y ~~Sent from my iPhone~~

Table 2: Example email

to approximate the time needed for the fine-tuning on the whole dataset, hence we initially experimented only on a subset of annotated emails, whose distribution is shown in Figure 4.

With the learning rate of 1e-6, 10 epochs and batch size equal to 16, SLOBERTA was fine-tuned for the 10-class classification yielding TOPICSLOBERTA (around 6 hours on a local CPU machine). Simultaneously, topics were replaced with their priorities, resulting in 74 emails with low priority, 754 with medium and 490 with high priority and yielding PRIORSLOBERTA (around 4 hours). However, after experimenting with different parameters and adding more training samples from the other email batch, we realised that model's performance does slightly improve with the increase in the number of instances, so our conclusion was that we needed more training instances.

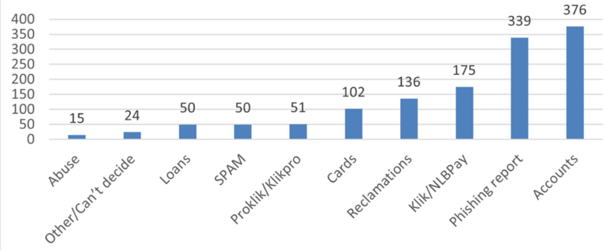

Figure 4: Sample topic distribution

However, the results were not encouraging, since the model performed successfully mostly only for the major classes, *Accounts* and *Phishing report*. Nevertheless, we continued to assess the other two proposed approaches, what we describe in detail in the next subsections.

5.2 Waterfall-1 approach

Core concept of the both WATERFALL techniques is to train separate classifiers and assembly them into one step-structure. Since we wanted lightweight classifiers with fast inference, and we knew that we should not have high grand expectations from the transformers-based BERT language model, we decided to continue experimentation with the more

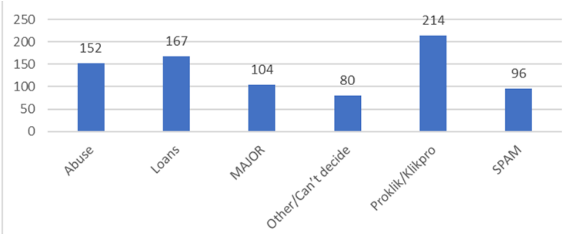

Figure 5: KC-DATASET-MINOR

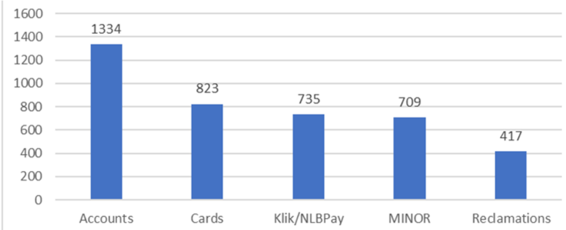

Figure 6: KC-DATASET-MAJOR

traditional machine learning models. In order to represent emails as vectors, we used the TF-IDF vectorisation technique. Among different classifiers with optimal parameters found by applying grid search technique, we trained a linear SVM (with C=10) on the first-annotation-round dataset and obtained more encouraging results. We concluded that for our type of dataset (in the means of length of instances and the overall size), traditional machine learning models are a better fit.

As described in Section 3, in the WATERFALL-1 approach we proposed to divide the final model into three sub-models, first phishing-report-classifier, and then one for the major and the other for the minor classes. Since the WATERFALL models require samples of all other classes present, unified into a single negative class, for the minor-classifier we uniformly sampled other classes and joined the samples into class *MAJOR*. This dataset dubbed as KC-DATASET-MINOR is shown in Figure 5.

The *Phishing report* category was separated for the WATERFALL models (comprised of 836 emails), since it represents a separate component in the inference process. Uniform sampling the same number of negative instances from the other classes yielded the KC-DATASET-PHI.

After adding minor-class representatives, sampled uniformly as in the previous step, we ended up with the KC-DATASET-MAJOR depicted in Figure 6.

The *Phishing report* category, and sampling uniformly the same number of instances of other classes for the purpose of having a balanced dataset for the KCPHI model we obtained the KC-DATASET-PHI.

After having the dataset prepared, we trained the SVM algorithm on these three datasets (8:2 ratio for the train/validation split). The resulting models were dubbed KCMINOR-SVM, KCMAJOR-SVM and KCPHI-SVM for the major, minor, and phishing components, respectively. The best parameters for the both KCMINOR-SVM and KCMAJOR-SVM were C=1000, gamma=0.001 with the RBF kernel, and for the KCPHI-SVM the optimal was linear SVM with C=1. The results on the validation test of all the three model components separately are shown in Table 3.

Class	P	R	F_1	Nr.
Abuse	.94	.86	.9	36
Other/Can't decide	.85	.65	.73	17
Loans	.93	.9	.91	41
SPAM	.84	.84	.84	19
Proklik/Klikpro	.74	.93	.82	30
MAJOR	.55	.55	.55	20
Cards	.91	.81	.86	171
Reclamations	.51	.53	.52	77
Klik/NLBPay	.86	.82	.84	144
MINOR	.79	.7	.74	149
Accounts	.75	.87	.81	263
Not	.90	.94	.92	164
Phishing report	.94	.89	.92	171

Table 3: WATERFALL-1 components on validation set

5.3 Waterfall-2 approach

The main idea of the novel WATERFALL-2 approach is to break the 10-class classification task into smaller tasks, as proposed in Figure 2. For the binary classifiers (the first four components) number of negative samples equal to number of positive samples, while in the case of the multi-class classifiers, number of negative samples for the *Non-Major* and *Non-Minor* represent the mean number of other class labels in the component, sampled from the natural distribution of the negative class labels. Each of these sub-datasets was divided into training and validation sets (9:1). Then we performed grid search for various classifiers on each of the sub-datasets, and finally trained and exported optimal models. We report our findings in Table 4 (LR represents Logistic Regression, while RF stands for Random Forest).

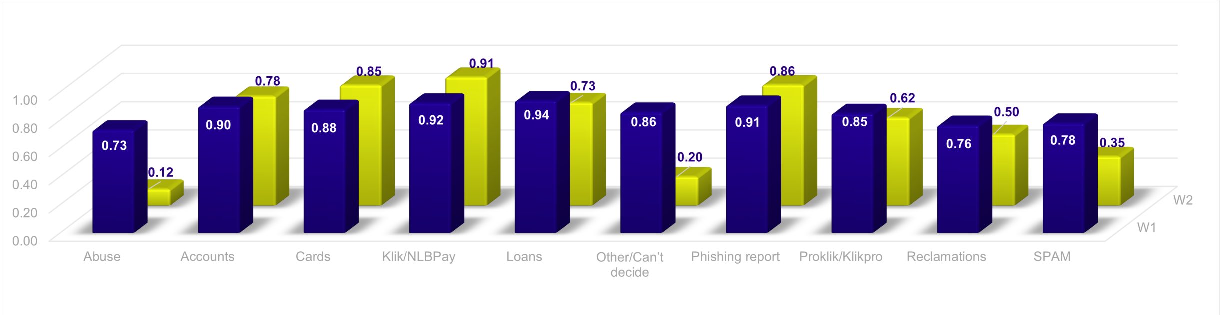

Figure 7: Waterfall-1 vs. Waterfall-2

	Nr.	A	F_1	Cls
SPAM	96	.9	.9	LR
Phishing	844	.91	.92	SVM
Abuse	153	.85	.86	RF
Reclamations	425	.78	.78	RF
Accounts	1406	.83	.84	LR
Cards	833		.81	
Klik/NLBPay	772		.85	
Non-Major	1003		.83	
Loans	174	.81	.79	SVM
Other/Can't decide	82		.5	
Proklik/Klikpro	215		.91	
Non-Minor	157		.7	

Table 4: Waterfall-2 components on validation set

5.4 Discussion

So far we have shown evaluation metrics only on separate components of the both approaches. After joining all components into two models, Figure 7 shows their F_1 scores on the whole 5,000-sample dataset. Worse performance of the Waterfall-2 could be interpreted as follows. Let us observe the SPAM component: there were 96 emails of that class in the dataset, and the same number of negative instances. The model has seen all cases of SPAM from our dataset during the training and recognises them perfectly. However, the model has seen only ninety-six examples that are not SPAM and mistakes frequently other classes for SPAM. In summary, it marked 456 emails as SPAM (therefore, the rate of false positives was extremely high) which is unacceptable for the final model.

The conclusion is that these smaller models work better separately, but assembled they are worse on our dataset. Each model has seen only a few samples from the negative pool. We could say that the BERT model-all-at-once approach and the Waterfall-2 approach represent opposite ends of the spectrum, whereas the Waterfall-1 approach strikes a balance in between. Therefore, for the first production model, we decided to use the Waterfall-1 approach.

We finally report Waterfall-1 performance on a separate, independent test set of emails in Table 5, comprising of 304 emails.

Class	P	R	F_1	Nr.
Accounts	.86	.8	.83	106
Cards	.76	.9	.82	27
Klik/NLBPay	.67	.75	.71	27
Loans	1	.9	.95	16
Non-Slovenian	.84	1	.91	3
Other/Can't decide	.71	.33	.46	16
Phishing report	.73	.95	.83	22
Proklik/Klikpro	.89	1	.95	20
Reclamations	.45	.45	.45	26
SPAM	1	.81	.9	41

Table 5: Waterfall-1 on a test set

6 Conclusions and Future Work

One way to further enhance the model would be to log the topic labels predicted by the model and see how many assigned topics were corrected by the person who received the email. This way the dataset would naturally grow, and we would get the feedback about number of cases the KC accepted the model's predictions, and in situations when that was not the case, what were the common mistakes and the reasons behind them.

With the enlarged dataset, it would be possible not only to improve existing Waterfall-1 model, but also to give another try to the other two approaches, since their bottlenecks in practice were lack of training samples.

References

Nashit Ali, Anum Fatima, Hureeza Shahzadi, Aman Ullah, and Kemal Polat. 2021. Feature Extraction Aligned Email Classification based on Imperative Sentence Selection through Deep Learning. *Journal of Artificial Intelligence and Systems*, 3(1):93–114.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Khalid Iqbal and Muhammad Shehrayar Khan. 2022. Email Classification Analysis using Machine Learning Techniques. *Applied Computing and Informatics*.

Maryam Shuaib, Oluwafemi Osho, Idris Ismaila, John K Alhassan, et al. 2018. Comparative Analysis of Classification Algorithms for Email Spam Detection. *International Journal of Computer Network and Information Security*, 12(1):60.

Akash Sundararaj and Gökhan Kul. 2021. Impact Analysis of Training Data Characteristics for Phishing Email Classification. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 12(2):85–98.

Matej Ulčar and Marko Robnik-Šikonja. 2021. SloBERTa: Slovene Monolingual Large Pretrained Masked Language Model. *Proceedings of SI-KDD within the Information Society 2021*, pages 17–20.