SMS Spam Detection

Barthélémy MARSAULT, Florian GIGOT, Gaétan JAGOREL

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{barthelemy.marsault,florian.gigot, gaetan.jagorel}@fer.hr

Abstract

Nowadays, a phone number is almost part of the identity of his owner, a lot of legal and commercial forms ask for it as a mandatory information. So, the more a phone number become known, the more the owner become likely to receive spam. Furthermore, almost no phone provides a good and integrated spam filtering software.To fight this, some models already exist and in this paper, we present our specific implementations and concept that we tried to increase the precision of our filtering. It also goes through they way we tested and trained our work and some problems that we faced.

1. Introduction

SPAM or spamming are unwanted messages that you receive by email or SMS, having an advertising or a fraudulent goal and for which you didn't ask. Nowadays, communication via technologies as internet, email or mobile phone takes a large space in our daily life, so as this importance grows, the spamming grows too. Some say that the volume of spam is today up to between 60 % and 90 % of the daily traffic of email over the internet and spam has become a huge business for all kind of companies .

The problem with the spamming is that it's often annoying and aims to try to scam the recipient, it could be a simple advertising trying to force to buy something by an interesting ad, but it could also be phishing or trying to cause the download of a virus. So, now, a lot of personal informations are managed on internet and accessible through a computer or a smartphone, we could consider the spamming detection as a cyber security threat. In fact, 64% of organizations have experienced a phishing attack in 2018 and the majority of this attacks were due to spams.

Regarding to the dangerousness that the spamming can involve whether for companies or private individuals, every respectful email boxes include a spam section and try to be as accurate as possible to differentiate spam from common messages. However, filtering options for mobile phone are not offering the same performances and are raising a lot of concern about the fact that an important message could be blocked. And even if the popularity of the SMS communication tends to decrease, the SMS spamming still appears to be quite popular in some region of the globe, like in china for example. Furthermore, now, a smartphone often store bank, work or any other private datas that can be stolen with simple hacking programs downloaded from a link in a spam SMS.

Knowing the possible issues link to the spamming, researchers are trying to solve the problem, but there are some difficulties in this field of research, for example, there is the lack of real public database of SMS. Besides, most of spam SMS come from unknown or changing phone number, which made them really hard to identify on this criteria. That only left the content of the message, which in most SMS, is short and fill with smileys and abbreviations.

To try to make a working spam detection system, we tried different existing models in order to compare possible results and get the best of them.

The following of the article will include in second part the description of the dataset, then the process part talking about the lexical analysis, features and implementation. To end the paper, a brief part will explain the training of our system and the results that come out of it.

2. Description of the dataset

The dataset we used is ”The SMS Spam Collection v.1” created by Tiago A. Almeida and José María Gómez Hidalgo (Almeida et al., 2011). This dataset is composed of 5574 real SMS messages in english, each tagged with either the Spam or Ham label. In this latter dataset, 86.6% of the messages are Ham and 13.4% are Spam (figure 1). We used this dataset as a base to train and test our model. Moreover, it served as a reference for our statistical analysis of the differentiation features between spam and ham.

This dataset is the largest available, however it remains a small dataset for human language processing. Having a large dataset is very important to have enough data to train and thus increase the performance of the model. Moreover, a large dataset allows us to better test our model because we can have a larger test set (the data used for training and testing must be different).

3. Analysis of the lexical field

In this part, we will study the language used in the two types of messages: spam and ham. In order to know if a distinct lexical field differentiates the two classes.

First, a word frequency study is carried out for each type of message. If we look at the list of the twenty most frequent words (table 1 and 2), we find several similarities between the two lists. For example, the words ”call” and ”u” are found in both lists. However, the majority of the words are different, thus this path is promising. To explore this path, now let's look at the frequency of bigrams. A bigram is a sequence of two words like “I am”.The list of the twenty most frequent bigrams of spam messages is very different from the list of ham messages (table 3 and 4). Moreover in these two lists, two bigrams stand out by taking the first

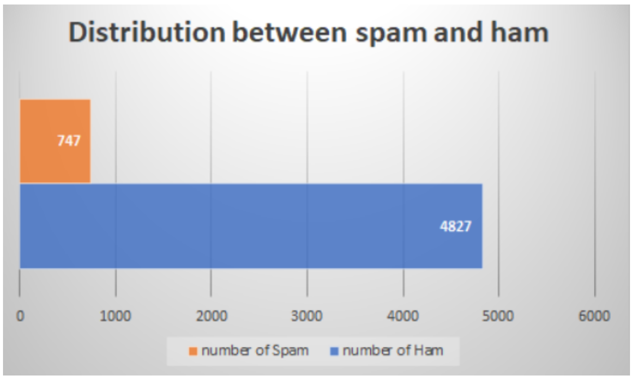

Figure 1: Shows the repartition between spam and ham messages in the dataset (747 spam vs 4827 ham).

Table 1: top 20 words in spam messages.

word	frequency
call	346
free	217
txt	156
u	144
ur	144
mobile	123
text	121
stop	114
claim	113
reply	104
prize	92
get	84
new	69
send	68
nokia	65
cash	62
urgent	62
win	60
contact	56
service	55

Table 2: top 20 words in ham messages.

word	frequency
u	973
gt	318
lt	316
get	301
go	246
got	242
ur	237
ok	235
know	234
like	231
call	230
good	227
come	225
time	195
love	180
day	176
going	168
one	167
want	163
home	158

Table 3: top 20 bigrams in spam messages.

bigram	frequency
(please, call)	45
(po, box)	24
(guaranteed, call)	23
(prize, guaranteed)	22
(call, landline)	22
(selected, receive)	19
(contact, u)	19
(send, stop)	19
(every, week)	19
(await, collection)	19
(call, claim)	18
(urgent, mobile)	18
(call, land)	18
(land, line)	18
(customer, service)	17
(chance, win)	17
(free, entry)	16
(claim, call)	16
(private, account)	16
(account, statement)	16

place, ”please call” for spam and ”lt gt” for ham. Interesting fact, in both cases their frequency of appearance is much higher than the bigrams taking the second place, with a frequency respectively +184.5% and +475.9% higher.

To conclude, using the notion of bigram in our implementation could be interesting to increase the accuracy of our model.

4. Feature Engineering

In this part, we look for features to differentiate between spam and ham. These features can be indicated to the model and thus support it in finding the correct solution.We studied the following features: the size of the message, the presence of a url link, the presence of a currency symbol and the presence of a phone number. In this list of features, only two are features that can assist the system to differentiate the type of a message. These two features are the presence of a phone number and/or the presence of a currency symbol. Indeed, in our study only spam contains currency symbol and/or phone number. Thus, in our implementation we have applied pre-processing on the messages in order to clearly indicate to the model if these two features are present. To do this, we replace the number string

Table 4: top 20 bigrams in ham messages.

bigram	frequency
(lt, gt)	276
(gon, na)	58
(call, later)	50
(let, know)	39
(sorry, call)	38
(r, u)	37
(u, r)	35
(good, morning)	30
(take, care)	29
(u, wan)	29
(wan, na)	28
(lt, decimal)	23
(decimal, gt)	23
(new, year)	23
(u, get)	23
(pls, send)	22
(ok, lor)	22
(gt, lt)	21
(u, still)	19
(good, night)	19

representing the phone number with “phone-number” and the currency symbols with “money-symbol”.

5. Implementation

To implement the system we used python with the following libraries: pandas to get data set analysis management tools and sklearn to easily introduce learning machines.

The structure of the system implementation is very simple (figure 2).

First we load a message, on this one we imply a first pre-processing to highlight the phone number and currency symbols if present. Then we imply a second pre-processing to enrich the message with the following information: are there any bigrams feature of spam or ham in the message? Finally, the message is analyzed by a Multinomial Naive Bayes classifier (Shirani-Mehr, 2018) to determine whether the message is spam or ham.

6. Training and Evaluating the model

To train our model we chose to use the first 4800 messages of our dataset, that is to say 86% of our dataset. The remaining 14% are used to test our model. Our objective being to have a maximum of data for training, while having enough for testing.

To evaluate our model, we used a total of 772 SMS which contains 671 ham messages (86.9%) and 101 spams (13.1%) . We based our evaluation on four criteria, the accuracy, the precision score, the recall score and the F1 score. We can see the results on Table 6.

Our system performed really well on the training dataset with an accuracy of 99.09% which is quite impressive. In comparison to other models, our model outperform the one created by Tiago A. Almeida, José María Gómez and Akebo Yamakami (Almeida et al., 2013) which was based on the same dataset and had an accuracy of 97.5%. However they used only the first 1674 messages for training their system while we used the first 4800 messages.

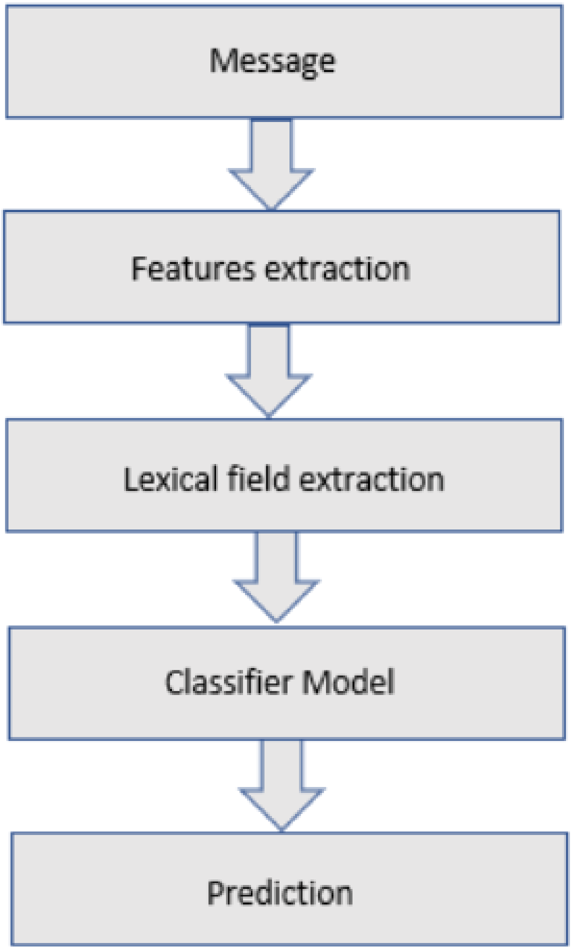

Figure 2: Procedure for implementation.

In our results, only 7 SMS were mispredicted as we can see in Table 5. We analyzed these messages in order to understand why the system failed. For example, the system predicted the message ”Nokia phone is lovly..” as a spam message while it is a ham. This kind of messages is really hard to predict because it is really short and even for a human it is not easy to be sure if it is either a ham or spam. Another example of misprediction is ”Hi this is Amy, we will be sending you a free phone number in a couple of days, which will give you an access to all the adult parties...”, our system detects this message as a ham but it is a spam. For this kind of messages, again it is hard to predict because there is no email address or phone number. Furthermore this SMS is written in a way that it looks like a normal conversation.

Finally, even if our results are very good, our system needs to be evaluated on a larger number of SMS to have better results.

Table 5: Detailed result of the prediction on the test set.

		predicted	
		ham	spam
actual	ham	688	3
	spam	4	97

Table 6: Predictions on the test set.

Accuracy of the model	0.9909326424870466
Precision score of the model	0.9820238095238095
Recall score of the model	0.977962550353396
F1 score of the model	0.9799809589431843

7. Conclusion

Automate the spam filtering is quite a hard task when it comes to SMS messages, during our work, we faced some important problems. The first one is the small number of representative SMS databases which made our system hard to test and train. The second one is the recurrent use of certain words on spam messages that often lead to a false positive detection on a ham. The third one is the fact that a lot of SMS use abbreviations and made them sometimes almost impossible to understand, which can lead to a misclassification.

As the result part shown, our results are correct, our work slightly improves the result of the original model we used and we can estimate that such an implementation could be helpful to avoid big stature scams. However, with the small number of spam examples that our system is trained on, an elaborate spamming technique could probably surpass our filtering.

Any future study on this domain should so consider implementing bigram techniques to better work with the classifier.(Hidalgo et al., 2012)

References

Tiago A. Almeida, José María Gómez Hidalgo, and Akebo Yamakami. 2011. The sms spam collection.

Tiago A. Almeida, Tiago A. Almeida, and Akebo Yamakami. 2013. Contributions to the study of sms spam filtering: New collection and results.

José María Gómez Hidalgo, Tiago A. Almeida, and Akebo Yamakami. 2012. On the validity of a new sms spam collection.

Houshmand Shirani-Mehr. 2018. Sms spam detection using machine learning approach.