Towards SMS Spam Filtering: Results under a New Dataset

Tiago A. Almeida*, José María Gómez Hidalgo**, Tiago P. Silva*

*Department of Computer Science, Federal University of São Carlos – UFSCar.
Sorocaba, São Paulo, Brazil.

**R&D Department, Optenet. Las Rozas, Madrid, Spain.

e-mail: talmeida@ufscar.br, jgomez@optenet.com, tiago.pasqualini@gmail.com

Abstract—The growth of mobile phone users has lead to a dramatic increasing of SMS spam messages. Recent reports clearly indicate that the volume of mobile phone spam is dramatically increasing year by year. In practice, fighting such plague is difficult by several factors, including the lower rate of SMS that has allowed many users and service providers to ignore the issue, and the limited availability of mobile phone spam-filtering software. Probably, one of the major concerns in academic settings is the scarcity of public SMS spam datasets, that are sorely needed for validation and comparison of different classifiers. Moreover, traditional content-based filters may have their performance seriously degraded since SMS messages are fairly short and their text is generally rife with idioms and abbreviations. In this paper, we present details about a new real, public and non-encoded SMS spam collection that is the largest one as far as we know. Moreover, we offer a comprehensive analysis of such dataset in order to ensure that there are no duplicated messages coming from previously existing datasets, since it may ease the task of learning SMS spam classifiers and could compromise the evaluation of methods. Additionally, we compare the performance achieved by several established machine learning techniques. In summary, the results indicate that the procedure followed to build the collection does not lead to near-duplicates and, regarding the classifiers, the Support Vector Machines outperforms other evaluated techniques and, hence, it can be used as a good baseline for further comparison.

Keywords—Mobile phone spam; SMS spam; spam filtering; text categorization; classification.

1. Introduction

Short Message Service (SMS) is the text communication service component of phone, web or mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between fixed line or mobile phone devices. They are commonly used between cell phone users, as a substitute for voice calls in situations where voice communication is impossible or undesirable. Such way of communication is also very popular because in some places text messages are significantly cheaper than placing a phone call to another mobile phone.

SMS has become a massive commercial industry since messaging still dominates mobile market non-voice revenues worldwide. According to Portio

Research[1], the worldwide mobile messaging market was worth USD 179.2 billion in 2010, has passed USD 200 billion in 2011, and probably will reach USD 300 billion in 2014. The same study indicates that annual worldwide SMS traffic volumes rose to over 6.9 trillion at end-2010 to break 8 trillion by end-2011.

The increasing popularity of SMS has led to messaging charges dropping below US$ 0.001 in markets like China, and even free of charge in others. Furthermore, with the explosive growth in text messaging along with unlimited texting plans it barely costs anything for the attackers to send malicious messages. This combined with the trust users inherently have in their mobile devices makes it an environment rife for attack. As a consequence, mobile phones are becoming the latest target of electronic junk mail, with a growing number of marketers using text messages to target subscribers. SMS spam (sometimes also called mobile phone spam) is any junk message delivered to a mobile phone as text messaging. Although this practice is rare in North America, it has been very common in some parts of Asia.

According to a Cloudmark report[2], the amount of mobile phone spam varies widely from region to region. For instance, in North America, much less than 1% of SMS messages were spam in 2010, while in parts of Asia up to 30% of messages were represented by spam. The same report reveals that financial fraud and spam via text messages is now growing at a rate of over 300 percent year over year. In fact, in a more recent report by the same firm[3], it is stated that about 30 million smishing (SMS Phishing) messages are sent to cell phone users across North America, Europe, and the U.K. Smishing is part of the much larger SMS spam problem. In the U.S. alone, there has been an almost 400 percent increase in unique SMS spam campaigns in the first half of the year 2012.

Besides being annoying, SMS spam can also be expensive since some people pay to receive messages. Moreover, there is a limited availability of mobile phone spam-filtering software and other concern is that important legitimate messages as of emergency nature could be blocked. Nonetheless, many providers offer their subscribers means for mitigating unsolicited SMS messages.

In the same way that carriers are facing many problems in dealing with SMS spam, academic researchers in this field are also experiencing difficulties. Probably, one of the major concern corresponds to the lack of large, real and public databases. So, although there has been significant effort to generate public benchmark datasets for anti-spam filtering, unlike email spam, which has available a large variety of datasets, the mobile spam filtering still has very few corpora usually of small size. Other concern is that established email spam filters may have their performance seriously degraded when directly employed to dealing with mobile spam, since the standard SMS messaging is limited to 140 bytes, which translates to 160 characters of the English alphabet. Moreover, their text is rife with idioms and abbreviations.

To fill these important gaps, we have recently proposed the new SMS Spam Collection [1], which is a real, public, non-encoded, and to the best of our knowledge it is the largest SMS spam corpus available. In this paper we have presented a lot details about the proposed dataset along with a comprehensive analysis to ensure that there are not duplicates coming from other former databases, since the added messages may contain previously

1. http://www.portioresearch.com/MMF11-15.html
2. http://www.cloudmark.com/en/article/mobile-operators-brace-for-global-surge-in-mobile-messaging-abuse
3. http://news.cnet.com/8301-1009_3-57494194-83/protect-yourself-from-smishing-video/

existing messages in the original collection, as it may ease the task of learning SMS spam classifiers. Moreover, we compare the performance achieved by several established machine learning methods in order to provide good baseline results for further comparison.

Separated pieces of this work were presented at ACM DOCENG 2011 [1] and IEEE ICMLA 2012 [2]. Here, we have connected all ideas in a very consistent way. We have also offered a lot more details about each study and extended the performance evaluation.

The remainder of this paper is organized as follows. Section 3 offers details about the newly-created SMS Spam Collection. A comprehensive near-duplicate analysis of the new SMS Spam Collection is presented in Section 4. In Section 5, we present a comprehensive performance evaluation for comparing several established machine learning approaches. Finally, Section 6 presents the main conclusions and outlines for future works.

2. Relevant works in SMS spam filtering

Unlike the growing and large number of papers about email spam classifiers (e.g. [3], [4], [5], [6], [7], [8], [9], [10], [11]), there are still few studies about SMS spam filtering available in the literature. Bellow, we present the most relevant works related to this topic.

Gómez Hidalgo *et. al.* [12] evaluated several Bayesian based classifiers to detect mobile phone spam. In this work, the authors proposed the first two well-known SMS spam datasets: the Spanish (199 spam and 1,157 ham) and English (82 spam and 1,119 ham) test databases. They have tested on them a number of messages representation techniques and machine learning algorithms, in terms of effectiveness. The results indicate that Bayesian filtering techniques can be effectively employed to classify SMS spam.

Cormack *et. al.* [13] have claimed that email filtering techniques require some adaptation to reach good levels of performance on SMS spam, especially regarding message representation. Thus, to support their assumption, they have performed experiments on SMS filtering using top performing email spam filters (*e.g.* Bogofilter, Dynamic Markov Compression, Logistic Regression, SVM, and OSBF) on mobile spam messages using a suitable feature representation. However, after analyzing the results, it was concluded that the differences among all the evaluated filters were not clear, so more experiments with a larger dataset would be required.

Cormack *et. al.* [14] have studied the problem of content-based spam filtering for short text messages that arise in three different contexts: SMS, blog comments, and email summary information such as might be displayed by a low-bandwidth client. Their main conclusions are that short messages contain an insufficient number of words to properly support bag of words or word bigram based spam classifiers and, in consequence, the filter's performance were improved markedly by expanding the set of features to include orthogonal sparse word bigrams and also to include character bigrams and trigrams. Among all analyzed approaches, the technique based on Dynamic Markov Compression achieved the best results on short messages and message fragments.

Liu and Wang [15] have proposed an index-based online text classification method, investigated two index models, and compared the performances of several index granularities for English and Chinese SMS messages. According to the results from the English dataset, the relevant feature among words can increase the classification confidence and the trigram co-occurrence feature of words is an appro-

priate relevant feature. On the other hand, the results from Chinese collection show that the performance of classifier applying word-level index model is better than the one applying document-level index model. According to the authors, the trigram segment outperforms the exact segment in indexing, so it is not necessary to segment Chinese text exactly when indexing by their proposed method.

Lee and Hsieh [16] proposed an interactive SMS confirmation mechanism using CAPTCHA and secret sharing. According to the authors, the found results indicate that it takes small computation costs to complete the authentication including the identity verification and the check of user-participation. So, they conclude that the proposed method is suitable for mobile environment.

A new large real, public and non-encoded SMS spam collection was proposed in Almeida *et. al.* [1]. Furthermore, the authors have lead an evaluation between several established machine learning methods and the results clearly indicate that SVM achieved the best performance, which can be used as a good baseline for further comparison.

Vallés and Rosso [17] have evaluated the performance achieved by plagiarism detection tools when employed as filters for SMS spam messages. They have carried out experiments on the SMS Spam Collection [1] and compared the results with the ones achieved by the well-known CLUTO framework. Their main conclusion is that plagiarism detection tools have detected a good number of near-duplicate SMS spam messages and outperformed the CLUTO clustering tool.

Delany *et. al.* [18] have reviewed recent developments in SMS spam filtering and also discussed important issues with data collection and availability for furthering research, beyond being analyzed a large corpus of SMS spam. They have built a new dataset with ham messages extracted from GrumbleText and WhoCallsMe websites and spam messages from the SMS Spam Collection. They analyzed different types of spam using content-based clustering and identified ten clearly-defined clusters. According to the authors, such result may reflect the extent of near-repetition in data due to the similarity between different spam attacks and the breadth of obfuscation used by spammers.

Nuruzzaman *et. al.* [19] evaluated the performance of filtering SMS spam on independent mobile phones using Text Classification techniques. The training, filtering, and updating processes were performed on an independent mobile phone. Their found results show that the proposed model was able to filter SMS spam with reasonable accuracy, minimum storage consumption, and acceptable processing time without support from a computer or using a large amount of SMS data for training.

Coskun and Giura [20] presented a network-based online detection method to identify SMS spamming campaign by detecting an unusual number of similar messages sent in a network over a short period of time. The proposed scheme uses counting Bloom filters to maintain approximate count of message content occurrences. According to the authors, the method achieved a detection rate close to 100% with a counting Bloom filter of size larger than 500,000 bins for detecting as few as 10 similar spam messages that differ by at most 20 characters within 10,000 regular SMS messages. The authors claim that their method uses a fast online algorithm which can be deployed in large carrier networks to detect spam activities before too many spam messages are delivered. It does not store SMS message contents, therefore it does not compromise the privacy of mobile subscribers.

Qian *et. al.* [21] proposed a service-side solution that uses graph data mining to distinguish likely spammers from normal senders. In fact, they

investigate ways to detect spam on the basis of features that include temporal and graph-topology information but exclude content, thus addressing user privacy issues. More specifically, the authors focused on identifying professional spammers on the basis of overall message-sending patterns. In their performance evaluation, they carried out experiments on another real-world dataset that has been used to detect spammers in online video social networks and compared the results with SVM and k-NN classifiers. According to the authors, the SVM classifier has a stronger ability to detect spammers in online video social networks compared to the k-NN classifier. However, they showed that temporal and network features can be incorporated into conventional static features to achieve better performance when detecting spammers.

3. The SMS Spam Collection

Reliable data are essential in any scientific research. The processes of evaluation and comparison of methods can be seriously impacted by the lack of representative data. Consequently, areas of more recent studies generally suffer with the absence of public available data.

Studies of mobile spam filtering is one of these affected areas. Although there are a few databases of legitimate SMS messages available on the Internet, finding real samples of mobile phone spam is not a simple task. Due to these reasons, to create the SMS Spam Collection we used data derived from several sources.

In order to get legitimate samples, we have inserted 450 SMS ham messages collected from Caroline Tag's PhD Thesis, available at http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf.

We have also included a subset of 3,375 SMS ham messages randomly chosen from the NUS SMS Corpus, which is a dataset of about 10,000 legitimate messages collected for research at the Department of Computer Science at the National University of Singapore. These messages were collected from volunteers, mostly Singaporeans and students attending the University, who were made aware that their contributions were going to be made publicly available. The NUS SMS Corpus is available at: http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/.

Then, we added a collection of 425 SMS spam messages manually extracted from the Grumbletext Web site. This is a UK forum in which cell phone users make public claims about SMS spam messages, most of them without reporting the actual spam message received. The identification of the text of spam messages in the claims is a very hard and time-consuming task, and it involved carefully reading through hundreds of web pages. The Grumbletext Web site is: http://www.grumbletext.co.uk/.

Finally, we incorporated the SMS Spam Corpus v.0.1 Big. This collection has 1,002 SMS ham messages and 322 spam messages and it is public available at: http://www.esp.uem.es/jmgomez/smsspamcorpus/. This corpus has been used in the following academic research efforts: [13], [14], and [12]. The sources used in this corpus are also the Grumbletext Web site and the NUS SMS Corpus.

The created corpus is composed by just one text file, where each line has the correct class followed by the raw message. We offer some examples in Table 1.

The SMS Spam Collection is public available at http://www.dt.fee.unicamp.br/~tiago/smsspamcollection.

In the following we present some statistics of the dataset. In summary, the new collection is composed by 4,827 legitimate messages and 747 mobile spam

TABLE 1: Examples of messages present in the SMS Spam Collection.

ham	What you doing?how are you?
ham	Ok lar... Joking wif u oni...
ham	dun say so early hor... U c already then say...
ham	MY NO. IN LUTON 0125698789 RING ME IF UR AROUND! H*
ham	Siva is in hostel aha:-.
ham	Cos i was out shopping wif darren jus now n i called him 2 ask wat present he wan lor. Then he started guessing who i was wif n he finally guessed darren lor.
spam	FreeMsg: Txt: CALL to No: 86888 & claim your reward of 3 hours talk time to use from your phone now! ubscribe6GBP/ mnth inc 3hrs 16 stop?txtStop
spam	URGENT! Your Mobile No 07808726822 was awarded a £2,000 Bonus Caller Prize on 02/09/03! This is our 2nd attempt to contact YOU! Call 0871-872-9758 BOX95QU

messages, a total of 5,574 short messages. To the best of our knowledge, it is the largest available SMS spam corpus that currently exists. Table 2 shows the basic statistics of the created database.

TABLE 2: Basic statistics

Msg	Amount	%
Hams	4,827	86.60
Spams	747	13.40
Total	5,574	100.00

Table 3 presents the statistics related to the tokens extracted from the corpus. Note that, the proposed dataset has a total of 81,175 tokens and mobile phone spam has in average ten tokens more than legitimate messages.

We have also performed a study regarding the occurrence frequency of tokens in each class. Tables 4 and 5 show the twenty tokens that most have appeared in ham and spam messages, respectively.

To complement the study regarding token frequency among each class, we also evaluated the degree of importance of each token over the full corpus. For this, we sorted all the tokens according to the information gain score (IG) [22] and present the first twenty ones in Table 6.

TABLE 3: Token statistics

Hams	63,632
Spams	17,543
Total	81,175
Avg per Msg	14.56
Avg in Hams	13.18
Avg in Spams	23.48

4. Duplicate analysis of the SMS Spam Collection

To ensure that the way the SMS Spam Collection has built, by reusing the same message sources, does not lead to invalid SMS spam filtering results, it is needed to study the potential overlap between the sub-collections that have been used when building it. The hypothesis is that the messages added to

TABLE 4: The twenty tokens that most appeared in ham messages

Token	Number of Hams Msg	% of Hams
i	1619	33.54
you	1264	26.19
to	1219	25.25
a	880	18.23
the	867	17.96
in	737	15.27
and	685	14.19
u	678	14.05
me	639	13.24
is	603	12.49
my	600	12.43
it	464	9.61
of	454	9.41
for	443	9.18
that	421	8.72
im	414	8.58
but	411	8.51
so	403	8.35
have	401	8.31
not	384	7.96

TABLE 5: The twenty tokens that most appeared in spam messages

Token	Number of spam Msg	% of Spams
to	467	62.52
call	329	44.04
a	294	39.36
your	227	30.39
you	218	29.18
for	177	23.69
or	177	23.69
the	167	22.36
free	157	21.02
txt	145	19.41
2	142	19.01
is	140	18.74
have	127	17.00
from	124	16.60
on	119	15.93
u	118	15.80
ur	114	15.26
now	112	14.99
and	108	14.46
claim	108	14.46

the original SMS collection, even extracted from the same sources (the Grumbletext site, the NUS SMS Corpus), do not add duplicates to those previously existing messages, except for those previously existing in the original collection or the messages sources themselves. In this way, if there are duplicates in the final collection, the only causes can be:

- Spammers do use templates when writing their spam messages.
- Legitimate users do make use of message templates existing in their mobile phones.
- Legitimate users do re-send chain letters (*e.g.* jokes, Christmas messages, etc.).

So, if the task of SMS spam filtering is eased because of these duplicate messages, the reason for this is the actual behavior of SMS messaging by spammers and legitimate users, and not the way the collection used for testing was built.

TABLE 6: The twenty tokens with highest IG score over the full corpus

Rank	IG	Token	Rank	IG	Token
1	0.099	call	11	0.036	won
2	0.066	txt	12	0.033	or
3	0.057	claim	13	0.033	now
4	0.057	free	14	0.033	&
5	0.057	to	15	0.032	stop
6	0.043	mobile	16	0.029	reply
7	0.043	www	17	0.028	win
8	0.041	i	18	0.028	text
9	0.037	prize	19	0.026	cash
10	0.036	your	20	0.025	co

In consequence, we have built three SMS sub-collections described below (original, added and all messages), and we have studied the most frequent duplicates in all the sub-collections. The hypothesis gets confirmed if:

1 The existing duplicates in the original sub-collection keep the same frequency statistics

in the final collection, and
2 the existing duplicates in the added messages keep the same frequency statistics in the final collection as well.

In the next sections, we describe the three sub-collections used in the study, along with the approach we have used to detect message duplicates, or more properly, near-duplicates. We detail the results of the analysis, which confirm our hypothesis.

4.1. Text collections

In order to evaluate the potential overlap between the datasets which were used to build the proposed SMS Spam Collection, we have searched for near-duplicates within three sub-collections:

- The previously existing SMS Spam Corpus v.0.1 Big (**INIT**).
- The SMS collection that includes the additional messages from Grumbletext, the NUS SMS Corpus, and the Tag's PhD Thesis (**ADD**).
- The released SMS Spam Collection (**FINAL**).

The **INIT** dataset has a total of 1,324 text messages where 1,002 are ham and 322 are spam. The **ADD** sub-collection is composed by 3,825 legitimate messages and 425 mobile spam messages, for a total of 4,250 text messages. The percentages of ham and spam are shown in Table 7.

TABLE 7: How the sub-collections are composed.

	INIT		**ADD**	
Class	Amount	Pct	Amount	Pct
Ham	1,002	75.68	3,825	90.00
Spam	322	24.32	425	10.00
Total	1,324	100.00	4,250	100.00

It is worth noticing that the previously existing SMS Spam Corpus v.0.1 Big, which corresponds to the **INIT** sub-collection, poses a simpler problem to machine learning content based spam filters, as the collection is more balanced than the new SMS Spam Collection. On the other side, the new collection is much bigger, and more data often implies better learning generalization.

In Table 8 we present the main statistics related to the tokens extracted from the **INIT** and **ADD** sub-collections.

TABLE 8: Basic statistics related to the tokens extracted from the sub-collections.

	INIT	**ADD**
Ham	12,192	51,419
Spam	7,682	9,861
Total	19,874	61,280
Avg per Msg	15.01	14.42
Avg in Ham	12.17	13.44
Avg in Spam	23.86	23.20

Note that, for both sub-collections, mobile phone spams are in average ten tokens larger than legitimate messages. Also note that the average tokens per message is quite similar in both sub-collections.

4.2. Near-duplicate analysis overview

Two texts are considered near-duplicates when, although they are not exactly the same, they are strikingly similar [23]. Finding near-duplicates has many applications, including plagiarism detection [24], Web searching and information retrieval improvements [23], or duplicate record detection in databases [25]. Depending on the application, researchers have made use of different techniques for near-duplicate detection. Moreover, even the definition of a near-duplicate can be application dependent, as the concept of "strikinglyness" is itself subjective [17]. In any case, given two text fragments, the goal is to compute a distance or similarity between them in order to decide if they

are near-duplicates. Of course, distance and similarity are opposite, and the idea is the smaller the distance, or the bigger the similarity between two texts, the more likely is they are near-duplicates. For simplicity, we will speak about **near-duplicate metrics**, considering both distances and similarities. Thus, metrics for near-duplicates detection can be organized in two main groups:

- **Syntactic metrics** make reference to those computations in which the actual order of text components (strings, tokens) is taken into account.
- **Semantic metrics** try to better capture the semantics of the text by using Vector Space Model (VSM)-like text representations and similarity computations [26].

It is worth mentioning that syntactic methods are most often called grammar-based in the literature plagiarism detection [24]. The most basic syntactic metrics are character sequence distances, like the Edit Distance, the Jaro Distance and many others [25], typically applied in the duplicate record problem. Thus, two text fields for different records in a database can be considered near-duplicates if the *e.g.* Edit Distance among them is below a threshold. Alternatively, two fields match if the longest common character sequence is longer that a predefined threshold.

In the areas of plagiarism detection and information retrieval, syntactic methods many often involve N-gram matching detection [23], [24]. An N-gram is an ordered sequence of tokens or words present in a text, in which N is the number of tokens. Text tokenization may involve punctuation removal, white space normalization, and other simplifications of the original text, in order to ensure that little manual changes do not hide plagiarism. Typical N sizes are 5 and 6, and obviously, the longer the N, the less probability of a false positive but the less effectiveness.

A significant example of a syntactic metric is the "String-of-Text" method, implemented by the WCopyfind[4] tool, and which involves scanning suspect texts for approximately matching character sequences. In order to avoid little manual modifications, this approximation can involve transformations like case changing, separators variation (*e.g.* addressing those users including more white spaces between words), etc.

Semantic methods are quite popular in these areas as well. The most popular technique by far is using the VSM [26]: representing texts as term-weight vectors, in which terms are typically stemmed words, and computing the cosine similarity between the target texts. A similarity very close to one between two texts represents a potential near-duplicate. This approach can be improved by using really semantic information as WordNet concepts, like in [27]. It is possible to combine both syntactic and semantic metric, like e.g. in [28].

4.3. Near-duplicate detection approach

For the particular needs of this study, and given the short nature of SMS messages, we consider the "String-of-Text" method as a reasonable baseline for the purpose of detecting near-duplicated messages in our collection. With this goal in mind, texts can be compared searching for N-grams for relatively big sizes (*e.g.* N=6), with additional parameters (length of match in number of characters, etc.). This approach is implemented in WCopyfind, but we have simplified it to N-gram matches after text normalization involving:

- Replacing all token separators by white spaces.
- Lowercasing all characters.
- Replacing digits by the character 'N' (to preserve phone numbers structure).

4. See: `http://plagiarism.phys.virginia.edu`

For instance, the 6-gram "`stop to NNNNN customer services NNNNNNNNNNN`" corresponds to a match between the next two messages within the **ADD** sub-collection:

```
Thank you, winner notified by sms. Good
Luck! No future marketing reply STOP
to 84122 customer services 08450542832
```

and

```
Your unique user ID is 1172. For removal
send STOP to 87239 customer services
08708034412
```

As it can be seen, both messages are not near-duplicates; instead, they share a common pattern in messages reported by users as SMS spam in the Grumbletext site, which is the matching 6-gram. In particular, both messages correspond to two different SMS advertising campaigns in which the users have actually not subscribed the service. In consequence, this near-duplicate approach, especially with relatively short N-Grams, can lead to many false positives. As a result, the statistics collected during our analysis represent an upper bound of the potential near-duplicates that occur in the final collection. In our opinion, this is safer than finding a lower bound, because in this way no near-duplicates will be missing, and the conclusions of the study are sound.

In order to find matching N-grams and message near-duplicates within a given sub-collection, we have followed the next procedure:

1 All messages within the sub-collection are taken as a sorted list.
2 Each N-gram for a message is built from left to right.
3 A match or hit is registered when an N-gram present in a message i is found in a message j, with $i < j$.
4 If a hit for messages i and j is registered, no other matches between those messages are stored.
5 All N-grams occurring in two or more messages are stored, along with the number of messages in which they occur.

Thus, if a particular N-gram is present in messages i, j and k with $i < j < k$, only the hits for i and j, and for j and k are counted. It must be noted that it is possible that there is a match between messages i and j, and another match between j and k, but not between i and k because both previous matching N-grams are different (although they may have some overlap). In consequence, the way we compare SMS messages is not symmetric.

It is worth noting that it may be the case that two messages have several N-grams in common. In fact, that would be the case for full long duplicate messages. In this situation, only the first left N-gram is reported, and then other co-occurring N-grams may be missing counts for yet other messages.

4.4. Results and analysis

The goal of this process is to check if merging the first two sub-collections adds many near-duplicates to the final database, in order to assess the overlap between both collections. Within each sub-collection, we have compared each pair of messages, stored all N size matches (N-grams with N = 5, 6, and 10), and sorted the N-grams according to their frequency, examining in detail the top ten ones per N. According to the literature, N = 6 is a typical number for detecting near-duplicate paragraphs, and we have tested N = 5 because some messages were exactly this long, but there are not nearly shorter messages. Moreover, while N = 5 or N = 6 can lead to many false positives, these hits can be refined

with the longer matches required with N = 10, which in turn is quite close to the actual message length average.

4.4..1 Frequency results

We show the overall N-gram occurrence statistics for N = 5, 6 and 10 in the **INIT**, **ADD** and **FINAL** sub-collections in Table 9. In the third column, we list the number of unique N-grams with 2 or more occurrences for a given size in each sub-collection. As it can be expected, we can view that the numbers increase with the the number of messages in each sub-collection.

TABLE 9: N-gram occurrence statistics for different sizes in the studied sub-collections.

N	sub	#uniq	sum	avg	std
5	**INIT**	186	573	3.08	1.56
	ADD	484	1292	2.67	2.02
	FINAL	718 (+48)	2175	3.03	2.24
6	**INIT**	140	420	3.00	1.37
	ADD	361	923	2.56	1.20
	FINAL	548 (+47)	1619	2.95	1.71
10	**INIT**	92	243	2.64	0.99
	ADD	192	489	2.55	1.33
	FINAL	354 (+70)	964	2.72	1.41

We can notice as well that, typically, the number of unique N-grams for the **FINAL** sub-collection is bigger than the sum of N-grams in the **INIT** and **ADD** sub-collections. The exact number of new N-grams that is added to the **FINAL** collection is presented in parenthesis. The difference of unique new N-grams between 5- and 6-grams is small and, as expected, there are less new 6-grams than 5-grams.

However, the number of new unique 10-grams is quite bigger than previous ones, what may be considered counter-intuitive. Moreover, and due to their length, 10-grams are much less likely to correspond to false positive near-duplicates. In consequence, we have examined those 10-grams in **FINAL** occurring exactly in a message in **INIT** and in a message in **ADD** (thus, with an exact frequency of 2). We have found that 52% of them do contain "N^+" strings, representing short and/or telephone numbers in spam messages, and in consequence, the matched messages belong to the same SMS spam campaign. It must be noted that SMS messages in the same spam campaign can use different short and/or telephone numbers. The remaining 10-grams with a frequency of 2 do correspond to:

- Other spam messages (*e.g.* "u are subscribed to the best mobile content service in").
- Chain letter messages extracted from the NUS SMS Corpus (*e.g.* "the xmas story is peace the xmas msg is love").
- Actual duplicates contributed to the NUS SMS Corpus (*e.g.* "i have been late in paying rent for the past").

Regarding the rest of figures in Table 9, the fourth, fifth and sixth columns report the total and the average number of hits per N-gram, plus the standard deviation, for each N-gram size and sub-collection, respectively. Only N-grams occurring in two or more messages are reported, because the N-grams considered are those that can correspond to near-duplicates. For instance, there are 573 hits of the 186 unique 5-grams with frequency of two or more messages for the **INIT** sub-collection, and each 5-gram occurs on an average of 3.08 ± 1.56 messages.

As it can be expected, the longer the N-grams, the less total number and average of matching messages, because the probability of getting a longer match between two randomly chosen messages is smaller. In general, the figures for **INIT** messages are bigger than for **ADD**, what makes sense because

the proportion of spam in the first collection is three times the proportion in the second collection, and most of the N-gram matches correspond to SMS spam messages. This explains as well that the average number of matches in the **FINAL** sub-collection is closer to the **INIT** average than to the **ADD** average, as the total counts of spam messages is 322 and 425 for these latter sub-collections, respectively. As previously discussed, most matches come from spam messages, that make for the near-duplicates because of the intrinsic similarity between spam campaigns patterns, and **ADD** spam messages sum up on previously existing campaigns and patterns in the **INIT** sub-collection. In other words, the spam class messages are typically more similar among them, than the ham class, for any of the sub-collections.

4.4..2 Top scoring N-grams

In order to compare the actual matches between messages in the studied sub-collection, we report the top frequent N-grams and their frequencies for each N in the next tables. We show the ten top frequent 5 and 6-grams in Tables 10 and 11, respectively.

First of all, it must be noted that, given an N-gram with counts i, j and k in the **INIT**, **ADD** and **FINAL** collections respectively, we must not expect that $i + j = k$. This is because some counts are missing as a previous N-gram match between two messages may have been reported, and only N-gram matches corresponding to the left most match between two messages are summed up.

As it can be seen regarding 5-grams:

- 5-grams already present in the **INIT** and the **ADD** sub-collections do not collapse to greatly increase their frequency. For instance, the 5-grams "sorry i ll call later" and "i cant pick the phone" do not change its frequency from **ADD** to **FINAL**. These 5-grams correspond to templates often present in cell phones, and used in legitimate messages. Actually, both are complete messages themselves.
- The behavior of the rest of 5-grams, which all actually nearly only occur in spam messages, is a bit different. Most of them are fuzzy duplicates that result in small frequency increases, like in "we are trying to contact" from **INIT** (10 messages) to **FINAL** (14 messages). This means that the messages in **ADD** may be duplicates of the messages in **INIT**. However, as it can be seen, the patterns of spam 5-grams within each sub-collection are very regular and even overlapping, so this is not significant. In other words, these 4 messages are not repeated, but new instances of spam probably sent by the same organization. Other messages just disappear from the top, as they keep their frequencies.

Regarding 6-grams (the standard value used in tools like WCopyfind), shown in Table 11, we can see that the behavior is quite similar to the case of 5-grams. There are slightly different results because of two reasons:

- The fact that longer N-grams must obviously lead to lower frequencies. Actually, there is not a significant drop in the number of matches per 6-gram, as it can be seen in *e.g.* "private your NNNN account statement for", which includes the 5-gram "private your NNNN account statement" as a prefix.
- The most frequent 6-grams keep on belonging to spam messages. The 5-grams that frequently occurred on the legitimate messages have disappeared because the detected templates are, in fact, complete 5-length messages.

In 6-gram results, we can see again that there are not significant near-duplicates except for those

TABLE 10: Ten top 5-grams and their frequencies in the studied sub-collections.

INIT		ADD		FINAL	
5-gram	#f	5-gram	#f	5-gram	#f
we are trying to contact	10	sorry i ll call later	37	sorry i ll call later	37
this is the Nnd attempt	9	private your NNNN account statement	15	private your NNNN account statement	16
urgent we are trying to	9	i cant pick the phone	12	we are trying to contact	14
prize guaranteed call NNNNNNNNNN from	8	hope you are having a	9	prize guaranteed call NNNNNNNNNN from	13
bonus caller prize on NN	7	text me when you re	9	you have won a guaranteed	13
draw txt music to NNNNN	7	£ NNNN cash or a	8	a NNNN prize guaranteed call	12
prize N claim is easy	7	NNN anytime any network mins	8	draw shows that you have	12
you have won a guaranteed	7	a £ NNNN prize guaranteed	7	i cant pick the phone	12
a N NNN bonus caller	6	have a secret admirer who	7	urgent we are trying to	11
are selected to receive a	6	u have a secret admirer	7	call NNNNNNNNNN from land line	10

TABLE 11: Ten top 6-grams and their frequencies in the studied sub-collections.

INIT		ADD		FINAL	
6-gram	#f	6-gram	#f	6-gram	#f
this is the Nnd attempt to	9	private your NNNN account statement for	15	private your NNNN account statement for	16
urgent we are trying to contact	9	i cant pick the phone right	12	a NNNN prize guaranteed call NNNNNNNNNN	12
prize guaranteed call NNNNNNNNNN from land	7	a £ NNNN prize guaranteed call	7	draw shows that you have won	12
a N NNN bonus caller prize	6	have a secret admirer who is	7	i cant pick the phone right	12
bonus caller prize on NN NN	6	i am on the way to	6	prize guaranteed call NNNNNNNNNN from land	12
cash await collection sae t cs	6	pls convey my birthday wishes to	6	urgent we are trying to contact	11
tone N ur mob every week	6	u have a secret admirer who	6	call our customer service representative on	10
you have won a guaranteed NNNN	6	£ NNN cash every wk txt	5	this is the Nnd attempt to	9
a NNNN prize guaranteed call NNNNNNNNNN	5	as i entered my cabin my	5	tone N ur mob every week	9
call NNNNNNNNNN now only NNp per	5	goodmorning today i am late for	5	we are trying to contact u	9

already present in each sub-collection. Moreover, the results of 10-grams are very similar to these previous ones with 6-grams. In consequence, we believe it is safe to say that merging the sub-

collections, although they have roughly the same sources, does not lead to near-duplicates that may ease the task of detecting SMS spam.

5. Experiments

As mobile phone messages often have a lot of abbreviations and idioms that may affect the filters accuracy, established email spam filters may have their performance seriously impacted when employed to classify this kind of messages. In this way, we have tested several well-known machine learning methods in the task of automatic spam filtering using the SMS Spam Collection in order to provide good baseline results for further comparison.

5.1. Tokenizers

Tokenization is the first stage in the classification pipeline. It involves breaking the text stream into tokens ("words"), usually by means of a regular expression. In this work, two different tokenizers were used:

1 tok1: tokens start with a printable character, followed by any number of alphanumeric characters, excluding dots, commas and colons from the middle of the pattern. With this pattern, domain names and mail addresses will be split at dots, so the classifier can recognize a domain even if subdomains vary [29].
2 tok2: any sequence of characters separated by blanks, tabs, returns, dots, commas, colons and dashes are considered as tokens. This simple tokenizer intends to preserve other symbols that may help to separate spam and legitimate messages.

In addition, we did not perform language-specific preprocessing techniques such as stop word removal or word stemming, since other researchers found that such techniques tend to hurt spam-filtering accuracy [5], [4].

5.2. Classifiers

The list of all evaluated classifiers are presented in Table 12[5].

TABLE 12: Evaluated classifiers

Basic Naïve Bayes (NB) – Basic NB [10]
Multinomial term frequency NB – MN TF NB [10]
Multinomial Boolean NB – MN Bool NB [10]
Multivariate Bernoulli NB – Bern NB [10]
Boolean NB – Bool NB [10]
Multivariate Gauss NB – Gauss NB [10]
Flexible Bayes – Flex NB [10]
Boosted NB [30]
Logistic Regression [31], [32]
Multilayer Perceptron [33]
Linear Support Vector Machine – SVM [34], [3]
Sequential Minimal Optimization – SMO [35]
Minimum Description Length – MDL [7]
K-Nearest Neighbors – KNN [36], [12] (K = 1, 3 or 5)
C4.5 [37], [12]
Boosted C4.5 [12]
PART [38], [12]
Random Forest [39], [40]

5.3. Baselines

Since the collection is highly biased to the legitimate class, a simple baseline is the trivial rejector (TR) for the spam class.

Given that the spam class has most of the tokens with the highest Information Gain score, it is sensible to expect that messages may get automatically grouped into two classes on the basis of those tokens. In consequence, we provide an additional baseline in the form of the results of

5. Some of the implementations of the described classifiers are provided by the Machine Learning library WEKA, available at http://www.cs.waikato.ac.nz/ml/weka/. The algorithms have been used with their default parameters except when otherwise is specified.

the Expectation-Maximization (EM) clustering algorithm [41], over a vector representation based on the tokenizer tok2. EM is an iterative soft clusterer that estimates cluster densities. Basically, cluster membership is a hidden latent variable that the maximum likelihood EM method estimates.

EM clustering works in the following way. Initially the instances are randomly assigned to the clusters. Distributions for each cluster are learned from this starting point, and then the E and M step of the algorithm are executed in subsequent iterations. The E step estimates the cluster membership of each instance given the current model – this is a soft, probabilistic membership where the predicted density/probability distribution is used to weight each instance. Then the M step re-estimates the parameters of the normal and discrete distributions for each cluster using the weights computed by the E step. Iteration stops when the likelihood of the training data with respect to the model does not increase enough from one iteration to the next, or the maximum number of iterations have been performed.

In our experiments, we have limited the maximum number of iterations to 20 and used the rest of the default values for EM parameters in WEKA.

5.4. Protocol

We carried out this experiments using the following protocol. We divided the corpus in two parts: the first 30% of the messages were separated for training the methods (1,674 messages) and the remainder ones for testing (3,900 messages). Since all messages are fairly short, we did not use any kind of method to reduce the dimensionality of the training space, *e.g.*, terms selection techniques.

To compare the results achieved by the filters we employed the following well-known performance measures:

- Spam Caught (%) – SC;
- Blocked Hams (%) – BH;
- Accuracy (%) – Acc;
- Matthews Correlation Coefficient – MCC [6].

MCC is used in machine learning as a measure of the quality of binary classifications. It returns a real value between -1 and $+1$. A coefficient equals to $+1$ indicates a perfect prediction; 0, an average random prediction; and -1, an inverse prediction [7].

$$MCC = \frac{(tp \times tn) - (fp \times fn)}{\sqrt{(tp + fp) \times (tp + fn) \times (tn + fp) \times (tn + fn)}},$$

where tp corresponds to the amount of true positives, tn is the number of true negatives, fp is the amount of false positives, and fn is the number of false negatives.

5.5. Results

Table 13 presents the best results achieved by each evaluated classifier and tokenizer. Note that the results are sorted in descending order of MCC.

Although the Logistic Regression scored a slightly better MCC and caught more spam than SVM, it has blocked more than 2% of legitimate messages, against only 0.18% from the SVM. Consequently, as in spam filtering, a false positive is an error worse than a false negative, we can safe conclude that SVM outperformed the other evaluated methods and accomplished a remarkable performance considering the EM and TR baselines and the high difficulty of classifying mobile phone messages. However, the results also indicate that the best five algorithms achieved similar performance with no statistical difference. All of them accomplished an accuracy rate superior than 97%, that can be considered as a very good baseline in a such context.

TABLE 13: The best results achieved by combinations of classifiers + tokenizers and the baselines Expectation-Maximization (EM) and trivial rejection (TR)

Classifier	$SC\%$	$BH\%$	$Acc\%$	MCC
Logistic Reg. + tok2	95.48	2.09	97.59	0.899
SVM + tok1	83.10	0.18	97.64	0.893
Boosted NB + tok2	84.48	0.53	97.50	0.887
SMO + tok2	82.91	0.29	97.50	0.887
Boosted C4.5 + tok2	81.53	0.62	97.05	0.865
MDL + tok1	75.44	0.35	96.26	0.826
PART + tok2	78.00	1.45	95.87	0.810
Random Forest + tok2	65.23	0.12	95.36	0.782
C4.5 + tok2	75.25	2.03	95.00	0.770
Bern NB + tok1	54.03	0.00	94.00	0.711
MN TF NB + tok1	52.06	0.00	93.74	0.697
MN Bool NB + tok1	51.87	0.00	93.72	0.695
1NN + tok2	43.81	0.00	92.70	0.636
Basic NB + tok1	48.53	1.42	92.05	0.600
Gauss NB + tok1	47.54	1.39	91.95	0.594
Flex NB + tok1	47.35	2.77	90.72	0.536
Boolean NB + tok1	98.04	26.01	77.13	0.507
3NN + tok2	23.77	0.00	90.10	0.462
EM + tok2	17.09	4.18	85.54	0.185
TR	0.00	0.00	86.95	–

It is important to point out that Logistic Regression, Boosted NB, SMO, and Boosted C4.5 also achieved good results since they found a good balance between false and true positive rates. On the other hand, the remainder evaluated approaches had an unsatisfying performance. Note that, although the most of them have obtained accuracy rate superior than 90%, they have correctly filtered about only 50% of spams or even less.

Therefore, based on the achieved results, we can certainly conclude that the linear SVM offers the best baseline performance for further comparison.

6. Conclusions

The task of automatic SMS spam filtering is still a real challenge nowadays. Three main issues difficult the development of algorithms for this specific field of research: the absence of public and real datasets, the low number of features that can be extracted per message, and the fact that the messages are filled with idioms and abbreviations.

In order to fill some of those gaps, this paper presented a lot details about the SMS Spam Collection, that is the largest one as far as we know. Besides being large, it is also publicly available and composed by only non-encoded and real messages. Furthermore, this paper also offered statistics related to this dataset, such as tokens frequencies and the most relevant words in terms of information gain scores.

We have also performed a careful analysis of the SMS Spam Collection, since its corpus is composed by subsets of messages extracted from the same sources. This analysis was built in order to promote the experimentation with machine learning SMS spam classifiers. As this collection has been developed by enriching a previously existing SMS corpus using the same data sources, the added messages may contain previously existing messages in the original collection. Thus, it is required to ensure that this does not happen, as it may ease the task of learning SMS spam classifiers. In this sense, an analysis of potential near-duplicates was performed. We used a standard "String-to-text" method, on three sub-collections: the original one (**INIT**), the added messages (**ADD**), and the final collection (**FINAL**). The near-duplicate detection method consists of finding N-gram matches between messages, for N = 5, 6 and 10 within each collection, in order to verify that there is not a significant number of near-duplicates in the **FINAL** sub-collection, apart from those previously existing in the **INIT** and the **ADD** sub-collections.

We found that 5-grams already presented in the **INIT** and the **ADD** sub-collections do not collapse

to greatly increase their frequencies, and they typically correspond to templates often presented in cell phones, and used in legitimate messages (*e.g.* "sorry i ll call later"). The 5-grams that co-occur in **INIT** and **ADD**, so they get their frequencies increased in **FINAL**, are new instances of spam most likely sent by the same organization. In 6-grams results, we found that there are not significant near-duplicates except for those already presented in each sub-collection. Moreover, the results achieved with 10-grams are very similar to the 5- and 6-grams ones. In consequence, we believe it is safe to say that merging the sub-collections, although they have roughly the same sources, does not lead to near-duplicates that may ease the task of detecting SMS spam.

Finally, we compared the performance achieved by several established machine learning methods and the found results indicate that Support Vector Machine outperforms other evaluated classifiers and, hence, it can be used as a good baseline for further comparison.

Future work should consider to use different strategies to increase the dimensionality of the feature space. Well-known techniques, such as orthogonal sparse bigrams (OSB), 2-grams, 3-grams, among others could be employed with the standard tokenizers to produce a larger number of tokens and patterns which can assist the classifier to separate ham messages from spam. Additionally, we plan to perform throughout experiments with machine learning content based classifiers in order to confirm and improve previous work by we and others ([13], [14], and [12]) on the much smaller SMS Spam Corpus.

Acknowledgments

The authors would like to thank the financial support of Brazilian agencies FAPESP and CNPq.

References

[1] T. Almeida, J. Gómez Hidalgo, and A. Yamakami, "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in *Proceedings of the 2011 ACM Symposium on Document Engineering*, Mountain View, CA, USA, 2011, pp. 259–262.

[2] J. M. Gómez Hidalgo, T. A. Almeida, and A. Yamakami, "On the Validity of a New SMS Spam Collection," in *Proceedings of the 2012 IEEE International Conference on Machine Learning and Applications*, Boca Raton, FL, USA, 2012, pp. 240–245.

[3] J. M. Gómez Hidalgo, "Evaluating Cost-Sensitive Unsolicited Bulk Email Categorization," in *Proceedings of the 17th ACM Symposium on Applied Computing*, Madrid, Spain, 2002, pp. 615–620.

[4] L. Zhang, J. Zhu, and T. Yao, "An Evaluation of Statistical Spam Filtering Techniques," *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 4, pp. 243–269, 2004.

[5] G. Cormack, "Email Spam Filtering: A Systematic Review," *Foundations and Trends in Information Retrieval*, vol. 1, no. 4, pp. 335–455, 2008.

[6] T. A. Almeida, A. Yamakami, and J. Almeida, "Evaluation of Approaches for Dimensionality Reduction Applied with Naive Bayes Anti-Spam Filters," in *Proceedings of the 8th IEEE International Conference on Machine Learning and Applications*, Miami, FL, USA, 2009, pp. 517–522.

[7] ——, "Filtering Spams using the Minimum Description Length Principle," in *Proceedings of the 25th ACM Symposium On Applied Computing*, Sierre, Switzerland, 2010, pp. 1856–1860.

[8] ——, "Probabilistic Anti-Spam Filtering with Dimensionality Reduction," in *Proceedings of the 25th ACM Symposium On Applied Computing*, Sierre, Switzerland, 2010, pp. 1804–1808.

[9] T. A. Almeida and A. Yamakami, "Content-Based Spam Filtering," in *Proceedings of the 23rd IEEE International Joint Conference on Neural Networks*, Barcelona, Spain, 2010, pp. 1–7.

[10] T. A. Almeida, J. Almeida, and A. Yamakami, "Spam Filtering: How the Dimensionality Reduction Affects the Accuracy of Naive Bayes Classifiers," *Journal of Internet Services and Applications*, vol. 1, no. 3, pp. 183–200, 2011.

[11] T. A. Almeida and A. Yamakami, "Facing the Spammers: A Very Effective Approach to Avoid Junk E-mails," *Expert Systems with Applications*, vol. 39, pp. 6557–6561, 2012.

[12] J. M. Gómez Hidalgo, G. Cajigas Bringas, E. Puertas Sanz, and F. Carrero García, "Content Based SMS Spam Filtering," in *Proceedings of the 2006 ACM Symposium on Document Engineering*, Amsterdam, The Netherlands, 2006, pp. 107–114.

[13] G. V. Cormack, J. M. Gómez Hidalgo, and E. Puertas Sanz, "Feature Engineering for Mobile (SMS) Spam Filtering," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 2007, pp. 871–872.

[14] ——, "Spam Filtering for Short Messages," in *Proceedings of the 16th ACM Conference on Conference on information and Knowledge Management*, Lisbon, Portugal, 2007, pp. 313–320.

[15] W. Liu and T. Wang, "Index-based Online Text Classification for SMS Spam Filtering," *Journal of Computers*, vol. 5, no. 6, pp. 844–851, 2010.

[16] J. Lee and M. Hsieh, "An Interactive Mobile SMS Confirmation Method Using Secret Sharing Technique," *Computers and Security*, vol. 30, no. 8, pp. 830–839, 2011.

[17] E. Vallés and P. Rosso, "Detection of Near-duplicate User Generated Contents: The SMS Spam Collection," in *Proceedings of the 3rd International CIKM Workshop on Search and Mining User-Generated Contents*, 2011, pp. 27–33.

[18] S. J. Delany, M. Buckley, and D. Greene, "Sms spam filtering: Methods and data," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9899–9908, 2012.

[19] M. Taufiq Nuruzzaman, C. Lee, M. F. A. b. Abdullah, and D. Choi, "Simple sms spam filtering on independent mobile phone," *Security and Communication Networks*, vol. 5, no. 10, pp. 1209–1220, 2012.

[20] B. Coskun and P. Giura, "Mitigating sms spam by online detection of repetitive near-duplicate messages," in *2012 IEEE International Conference on Communications*, 2012, pp. 999 –1004.

[21] Q. Xu, E. Xiang, Q. Yang, J. Du, and J. Zhong, "Sms spam detection using noncontent features," *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 44–51, 2012.

[22] Y. Yang and J. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in *Proceedings of the 14th International Conference on Machine Learning*, Nashville, TN, USA, 1997, pp. 412–420.

[23] J. P. Kumar and P. Govindarajulu, "Duplicate and near duplicate documents detection: A review," *European Journal of Scientific Research*, vol. 32, pp. 514–527, 2009.

[24] A. M. El Tahir Ali, H. M. Dahwa Abdulla, and V. Snasel, "Survey of Plagiarism Detection Methods," in *Proceedings of the 5th Asia Modelling Symposium*, Manila, Philippines, 2011, pp. 39–42.

[25] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, pp. 1–16, January 2007.

[26] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.

[27] N. O. Kang, A. Gelbukh, and S. Y. Han, "Ppchecker: Plagiarism pattern checker in document copy detection," *Lecture Notes in Computer Science*, vol. 4188, pp. 661–667, 2006.

[28] A. Z. Broder, "On the resemblance and containment of documents," in *Compression and Complexity of Sequences*. Salerno, Italy: IEEE Computer Society Press, June 1997, pp. 21–29.

[29] C. Siefkes, F. Assis, S. Chhabra, and W. Yerazunis, "Combining Winnow and Orthogonal Sparse Bigrams for Incremental Spam Filtering," in *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Pisa, Italy, 2004, pp. 410–421.

[30] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Thirteenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 1996, pp. 148–156.

[31] S. J. Press and S. Wilson, "Choosing between logistic regression and discriminant analysis," *Journal of the American Statistical Association*, vol. 73, no. 364, pp. 699–705, 1978.

[32] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," pp. 841–848, 2002.

[33] S. S. Haykin, *Neural Networks and Learning Machines*. Prentice Hall, 2009.

[34] G. Forman, M. Scholz, and S. Rajaram, "Feature Shaping for Linear SVM Classifiers," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 2009, pp. 299–308.

[35] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Research, Tech. Rep. MSR-TR-98-14, 1998. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=69644

[36] D. Aha and D. Kibler, "Instance-based learning algorithms," *Machine Learning*, vol. 6, pp. 37–66, 1991.

[37] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[38] E. Frank and I. H. Witten, "Generating Accurate Rule Sets Without Global Optimization," in *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, USA, 1998, pp. 144–151.

[39] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[40] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, pp. 1–39, 2010.

[41] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.