Expert Systems with Applications xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Review

SMS spam filtering: Methods and data

Sarah Jane Delany [a,*], Mark Buckley [b], Derek Greene [c]

[a] School of Computing, Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland
[b] Digital Media Centre, Dublin Institute of Technology, Aungier Street, Dublin 2, Ireland
[c] School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland

ARTICLE INFO

Keywords:
SMS spam filtering
Text classification
SMS spam dataset

ABSTRACT

Mobile or SMS spam is a real and growing problem primarily due to the availability of very cheap bulk pre-pay SMS packages and the fact that SMS engenders higher response rates as it is a trusted and personal service. SMS spam filtering is a relatively new task which inherits many issues and solutions from email spam filtering. However it poses its own specific challenges. This paper motivates work on filtering SMS spam and reviews recent developments in SMS spam filtering. The paper also discusses the issues with data collection and availability for furthering research in this area, analyses a large corpus of SMS spam, and provides some initial benchmark results.

1. Introduction

Spam is unsolicited and unwanted messages sent electronically. Email spam is sent/received over the Internet while SMS spam is typically transmitted over a mobile network. Traditional email spammers are moving to the mobile networks as the return from the email channel is diminishing due to effective filtering, industry collaboration and user awareness. The Short Messaging Service (SMS) mobile communication system is attractive for criminal gangs for a number of reasons. It is becoming cost effective to target SMS because of the availability of unlimited pre-pay SMS packages in countries such as India, Pakistan, China, and increasingly the US. In addition SMS can result in higher response rates than email spam as SMS is a trusted service with subscribers comfortable with using it for confidential information exchange. According to the GSMA it is inevitable that mobile network operators across the globe will see a rise in the volume and sophistication of SMS attacks in 2011 (GSMA, 2011b).

SMS spam is an emerging problem in the Middle East and Asia, with SMS spam contributing to 20–30% of all SMS traffic in China and India (GSMA, 2011b). As an example of this Chinese mobile subscribers received 200 billion spam messages in one week in 2008.[1] While it is estimated that in North America the current level of mobile spam is currently only 0.1% of all messages per person per day (GSMA, 2011a), 44% of mobile device owners surveyed in the US reported receiving SMS spam.[2]

Apart from being a nuisance, mobile subscribers can suffer financial loss from SMS spam. By responding to an SMS spam subscribers can end up calling premium rate numbers or signing up to expensive subscription services. They can unknowingly access suspect websites and be at risk of phishing attacks or malware downloads. Mobile network operators are also suffering financially, experiencing higher network and operating costs and increased customer care costs in addition to damage to their brand and threat of regulation.

Current anti-spam measures in place in mobile operator networks include anti-spoofing and faking measures which can successfully identify SMS messages that have been manipulated to forge the originating details in order to avoid charges. With the rise in non spoofed or faked SMS spam messages the need for more sophisticated filtering techniques is increasing. Simple filtering methods use traffic analysis to identify high volumes of messages from individual subscribers.[3] A worrying dynamic is that spammers are using low volumes and advanced methods to avoid detection (GSMA, 2011b). They typically send small quantities of spam messages to observe how the operator's SMS infrastructure responds and then determine the volume limits policies. These directions indicate that content-based filtering is necessary to counteract the increasing threat of SMS spam.

This paper reviews the current state of the art in SMS spam filtering concentrating on the content-based technologies which are becoming more and more necessary in the battle against SMS spam. The rest of this paper is structured as follows: Section 2 discusses the similarities and differences between email and SMS spam filtering. Section 3 discusses the current research into content-based SMS spam filtering while Section 4 discusses the

* Corresponding author. Tel.: +353 1 402 4728; fax: +353 1 402 4985.
E-mail addresses: sarahjane.delany@dit.ie (S.J. Delany), mark.buckley@dit.ie (M. Buckley), derek.greene@ucd.ie (D. Greene).

[1] http://www.sophos.com/en-us/press-office/press-releases/2008/03/china_sms.aspx.

[2] http://www.cloudmark.com/en/resources/mobile-survey/.

[3] http://raiderstyle.com/SMS_SpamFraudPrevention.pdf.

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.053

issues with collecting SMS data for research into SMS spam filtering and analyses a large corpus of SMS spam messages. The paper then concludes in Section 5.

2. From email to SMS filtering

The (at least superficial) similarity of SMS spam filtering to email spam filtering suggests that proven techonologies in email spam filtering may be useful in combating SMS spam. The content-based technologies used in email spam filtering that are candidates for SMS spam filtering include both *direct content filtering* and *collaborative content filtering* techniques.

The direct content filtering technologies search or use the direct textual content of the message and vary from the simplistic keyword filtering to the more varied SpamAssassin-type rule sets, to the more complex automatic text classification approaches. Automatic text classification (Sebastiani, 2002) uses supervised machine learning algorithms to train a model on a set of examples of spam and legitimate messages which are labelled appropriately. This set is known as the training set and should be representative of typical spam and legitimate messages. The model learns from this training set how to distinguish spam from non spam and is used to predict whether new messages are spam or not.

Automatic text classification requires a representation of each message, typically an n-dimensional vector where each dimension represents a characteristic or feature that is predictive of the text classification problem. The features are identified by parsing and tokenisation of the textual content, a typical tokenisation being word tokenisation but n-gram character-based or word-based tokenisations are also popular. The value of each feature in the vector representation of a message is normally representative of the frequency of occurrence of that feature in the message.

Collaborative content filtering techniques allow a group of users to share information on spam messages. A successful approach is to generate a signature (sometimes known as a fingerprint) from the content of a known spam message and this is distributed and shared with a group of users. A signature is generated for all incoming messages and checked against the known spam signatures, and matches are labelled as spam messages. A well-known example is Vipul's Razor,[4] an un-disclosed variation of which is used by the email spam filtering company Cloudmark.[5] Collaborative filtering techniques rely heavily on the quality and amount of user-reporting of spam, which can be difficult in the mobile world as smart mobile devices and appropriate software technology are necessary to support user-reporting functionality.

The fact that many of the same issues apply across both filtering domains supports using proven email filtering technologies. Both domains have the technical issues of efficiency of filtering in real-time and have to decide between client-side and/or server-side filtering. More significantly, the characteristics of email spam filtering that make it a challenging filtering problem transfer also to the mobile space. The unequal and uncertain misclassification costs, in particular the requirement to exclude false positives (legitimate SMS messages that are incorrectly classified as spam by the filter) are as apparent in SMS spam filtering as in email spam filtering. In addition the issue of handling concept drift, the constant change in spam in order to bypass filters, is also a key challenge. There is already strong evidence of concept drift in current SMS spam with spammers using low volumes to avoid volume filters. As SMS spam becomes more prevalent and the filtering becomes more sophisticated in response, concept drift will become a significant problem in SMS spam filtering.

For SMS spam however a number of additional issues arise, firstly regarding the message itself. The maximum length of an SMS message is 160 characters which means there is little material for content-based filtering. Due to the short message length available, SMS subscribers use an idiosyncratic language subset with abbreviations, phonetic contractions, bad punctuation, emoticons, etc., which is different to the more traditional written language more typically used in emails (Kobus, Yvon, & Damnati, 2008; Ling, 2005). It has also been shown that email spam filtering can be improved by including contextual information found in the email headers (Rennie, 2000; Zhang, Zhu, & Yao, 2004; Lai, 2007) but SMS messages contain far less information in the headers, which offers less context to work with.

The mobile technology is also a factor. Client side solutions to spam filtering must operate on resource-constrained mobile devices. Despite the increasing use of smartphones, so-called "feature" phones, with only basic voice call and text functionality are still in the majority, especially in emerging markets where such phones continue to be launched and sold (Maina, 2010). Such devices also do not have the functionality to display a spam folder such as is common with email clients, so it is more difficult to tell users that messages have been blocked. Furthermore, mobile devices typically do not facilitate user reporting of spam messages, unless this service is offered by the network or by a third party, e.g. via a shortcode, which makes collaborative content filters, which rely on user feedback, difficult to implement.

Recently there has been research into applying the successful email spam filtering techniques to SMS spam filtering with some success. The next section will review the developments in SMS spam filtering which tend to focus on using the more popular supervised learning or text classification approaches but it will also discuss research into other types of classification approaches used including frequency analysis and social network analysis.

3. Content based SMS spam filtering

Early work proposing the application of automatic text classification techniques to SMS spam filtering includes work by Xiang, Chowdhury, and Ali (2004) who suggested that Support Vector Machines (SVMs) would be appropriate for the problem but did not evaluate their use, and work by Healy, Delany, and Zamolotskikh (2005) that considered using k-NN classifiers. Gómez Hidalgo, Bringas, Sánz, and García (2006) evaluated a number of classification algorithms on two SMS spam datasets and concluded that these techniques can be effectively transferred from email to SMS spam filtering, with SVMs being the most suitable. Work by Cai, Tang, and Hu (2008) on a Chinese spam dataset used the simpler and lesser used Winnow algorithm (Littlestone, 1988), a linear classifier that has shown good performance in high dimensional feature domains with irrelevant features.

Wu, Wu, and Chen (2008) used a Bayes learner to extract keywords for monitoring traffic centrally, allowing a spamminess score to be assigned, however this work was not evaluated. Jie, Bei, and Wenjing (2010) added a cost function to a Naive Bayes filter which assigned a high cost to false positives. This translates into a high spam classification threshold, and a higher threshold results in higher spam precision. Longzhen, An, and Longjun (2009) proposed using a k-nearest neighbour algorithm (k-NN) as part of a multi-filtering approach. After black- and white-listing, a message is first classified by a filter using rough sets, which provide approximate descriptions of concepts. If this filter classifies the message as spam, it is then passed to the k-NN classifier for final classification. An evaluation on a data set of 550 spam SMS and 200 non-spam SMS with $k = 12$ showed that this dual filtering method is faster and more accurate than using k-NN alone.

[4] http://razor.sourceforge.net/.

[5] http://www.cloudmark.com/.

In other recent work Liu and Wang (2010) proposed a simple index model which calculates the spaminess score of each SMS message as a function of the frequency of occurrence of features across the different categories of spam and non-spam in the training data. Their approach uses inverted indexes for speed of access and ease of update and they suggested that an ensemble of these index models, each based on a different feature set derived from the lexical analysis of the message content and the header information, provides good filtering performance.

Junaid and Farooq (2011) investigated the use of evolutionary classifiers for filtering SMS spam. They compare five supervised learning algorithms with four evolutionary classifiers. Results show comparable performance in general but the sUpervised Classifier System (UCS), a Michigan style rule-based learning classifier system, outperformed the others when over 3000 test messages were presented for filtering. The authors claim that this is due to the capability of UCS to evolve rules online. A limitation of such evolutionary classifiers is the performance at runtime. Junaid and Farooq's (2011) work reported that the average time to classify a message using the supervised learning algorithms is a fraction of a second, while most evolutionary classifiers require 3 to 4 s for classification although UCS is faster at 1.2 s.

Most recently Almeida, Gómez Hidalgo, and Yamakami (2011) have reported on a comparison of a number of supervised learning algorithms to provide baseline results for each. They use a corpus of 5574 messages in total which contains 747 spam and 4827 non-spam messages. Two methods of tokenisation are tested, a tokeniser which separates on any character other than alphanumeric and certain punctutation characters (comma, dash, dot and colon) and a variation which also tokenises domain names and mail addresses. The 13 classifiers used in the experiment include 8 variations of Naive Bayes, a linear SVM, a Minimum Description Length classifier, k-NN, the decision tree learner C4.5, and PART, a rule learner. They find that the linear SVM along with the alphanumeric tokenisation performs best, with an overall accuracy of 97.64%, a false positive rate of 0.18%, and a recall on the spam class of 83.1%. The next three top-ranked algorithms, namely boosted NB, boosted C4.5 and PART, were not significantly worse, each with an overall accuracy of 97.5%.

Hybrid approaches have also been proposed which combine content-based filtering with challenge-response, a technique which automatically sends a reply to a message sender which requires the sender to perform some action to ensure delivery of their message (Yoon, Kim, & Huh, 2010; He, Wen, & Zheng, 2008). Challenge-response systems have been put forward for email spam filtering but there is considerable anecdotal evidence discrediting them (Graham-Cumming, 2005).[6] Their limitations include increased network traffic, problems with receiving legitimate messages from valid automated online services such as mailing lists or online retailers and the fact that they are open to abuse. Yoon et al.'s (2010) proposal to address some of the limitations was to use challenge-response for a limited subset of SMS messages where the content-based classifier is uncertain of the classification. The filter is operated centrally and they identify a number of protocols of use involving an image CAPTCHA to ensure delivery. He et al.'s (2008) suggestion was that an image CAPTCHA should be generated for each message which is neither centrally black- or white-listed.

Much of the work already discussed offers server-based, centrally focused solutions to the SMS spam filtering problem. However, there are a number of researchers who have suggested solutions which are installed on the client mobile device or that are part of a distributed approach that incorporates central server processing with client side processing. Deng and Peng (2006) proposed a distributed spam filtering system using a Naive Bayes classifier on the client mobile device. User feedback on the client side is needed to confirm spam classifications and misclassifications and these are reported back to the SMS processing centre, via a shortcode, where the classifier is retrained and filter updates are downloaded to the mobile phone. The central processing also includes traffic analysis of sender information ranking senders based on their likelihood of sending spam. This likelihood value is calculated for each sender as a function of the sender's sending frequency, the consistency of the interval between sender's messages and the proportion of receivers to whom messages are sent by that sender.

Yadav, Kumaraguru, Goyal, Gupta, and Naik's (2011) approach is similar to Deng and Peng's (2006) in that they propose a client side Naive Bayes filter which uses the occurrence of keywords that appear in spam messages to determine a spam score. Messages that score above a certain threshold are labelled as spam. Their solution also requires user feedback to confirm and correct errors made by the classifier and therefore their filter can learn new spam keywords from client reports to a central server, which are in turn pushed out to other clients.

A different client side approach which considered a byte-level representation of the messages is proposed by Rafique and Farooq (2010). They trained individual first-order HMMs to model the probabilities of occurrence of particular byte sequences for both spam and legitimate messages. These probabilities were used to calculate a spam score for an unseen message which was classified as spam if the score exceeded a specific threshold. Their solution was deployed at the access control layer of a smart mobile device. The motivation behind this work was to provide a lightweight client-side solution that was deployable on resource-constrained mobile devices. A potential limitation of this approach is the difficulty in the re-training required to handle concept drift.

3.1. Feature engineering in SMS spam

The success of machine learning techniques depends greatly on the selection of an appropriate feature set for the problem in question. There has been work in feature engineering for mobile spam which attempts to identify the best features to use in the message representation. A feature set including words, normalised (i.e. lowercase) words, character bi- and tri-grams and word bi-grams suggested by Gómez Hidalgo et al. (2006) has provided a base feature set for much of the work in feature engineering. Cormack, Gómez Hidalgo, and Sánz (2007) found that a slight variation on this set including orthogonal word bigrams improved the performance of classification algorithms on SMS spam data. Sohn, Lee, and Rim (2009) expanded the base feature set by including features based on stylometry[7] suggested in author attribution studies. The stylistic features extracted using shallow linguistic analysis included the byte and the average byte length of messages, function word frequencies, part of speech n-grams and emoticon and special character frequencies which were extracted using manually constructed lexicons. These stylistic features, tested on a Korean SMS spam dataset, were shown to be potentially useful in improving the performance of a maximum entropy based spam filter, with the length features providing most benefit and part of speech features least benefit. There is some evidence across the existing work on SMS spam filtering that the length of the message is a strong predictive feature (Deng & Peng, 2006; Sohn et al., 2009; Liu & Wang, 2010; Yadav et al., 2011) although these studies typically use single language datasets and as such this result may not scale across multiple languages.

In their review of machine learning methods for email spam filtering Guzella and Caminhas (2009) report that bag of words

[6] http://www.pcmag.com/article2/0,2817,2386036,00.asp.

[7] Stylometry is the statistical analysis of linguistic style.

representation is the most widely used for email spam filtering. They caution that this leads to a bias in the problem due to the difficulty in updating the feature set to add new and remove existing words that become less predictive as the concept changes. This problem exists in SMS spam filtering also, however work by Junaid and Farooq (2011) attempts to get over this problem by using a representation which includes all possible octet bigram combinations (1521 features in total) in the feature set. They enhance the feature representation by including the frequency of each character appearing in the message. Due to encoding schemes used in the GSM standards there are limited character sets available and this results in an additional 256 characters or features in their feature set. They report a good performance of this feature set across a number of supervised and evolutionary learning algorithms although it was not directly compared with the more typical bag of words representation.

A recognised problem in text classification is the high dimensionality of the feature space. Most work in SMS spam filtering uses some sort of feature selection technique to reduce the large feature space, including Information Gain (Gómez Hidalgo et al., 2006; Sohn et al., 2009) and Mutual Information (Deng & Peng, 2006) which are widely accepted methods in text classification, but also including less commonly used methods such as Expected Cross Entropy (Cai et al., 2008)—interestingly Information Gain is also the most commonly used method for email spam filtering (Guzella & Caminhas, 2009). On the other hand, Almeida et al.'s (2011) comprehensive empirical study does not use any feature selection technique as they felt it was unnecessary due to the short length of SMS messages; their dataset had an average of 14 tokens per message. There is no indication in their work of the size of their feature set, although they extract over 81,000 tokens from their training set of 5574 messages. Due to the sparsity in textual representation it is likely that the feature space is very large.

A common language-independent pre-processing step performed by a number of researchers is feature abstraction, where features with infrequently re-occurring values such as URLs, phone numbers or currency amounts are replaced by a general feature representing the concept rather than the actual value (Cai et al., 2008; Deng & Peng, 2006).

3.2. Other machine learning approaches

Although supervised learning techniques feature in the majority of recent work in SMS spam filtering there have been other machine learning approaches investigated. An early centralised spam filtering solution was suggested by Dixit, Gupta, and Ravishankar (2005) where, rather than using the more standard text classification approaches, the SMS messages were represented as a character-based vector which was projected into a smaller normalised feature space and clustered to identify clusters of spam and non spam messages. New messages are classified based on their distance from the known spam and non spam clusters. This approach is motivated by the lack of keywords available for the normal classification algorithms due to the short length of the SMS messages, but the efficacy of this approach at classifying spam was not evaluated.

The behaviour of spam senders over time can be indicative of whether a given message is spam or not. Hu and Yan (2010) add a frequency analysis of SMS traffic to an existing spam filter with the goal of improving the central system's real-time processing speed. By considering the frequency of spam messages received during different time periods and at different locations, they focus filtering on specific time periods and locations. Their approach improves the throughput of the system greatly, but at the cost of a large decrease in spam detection and a significant rise to 2.5% in the false positive rate.

Non content-based technologies such as social network analysis have become popular in the email filtering area (Boykin & Roychowdhury, 2005; yu Lam & yan Yeung, 2007; Tseng & Chen, 2009). Network analysis approaches are address-based filtering approaches which aim to predict whether a sender is a potential spammer or not. This is different from the objective of the content and collaborative filtering techniques, which is to predict whether the message itself is spam or not. There is some evidence of the start of the use of these techniques for SMS filtering. Wang et al. (2010) presented an interesting solution for point-to-point SMS messages, those sent from one mobile device to another, which combines social network analysis with spectral analysis of message submission behaviour. They generate a directed graph from message logs and suggest two kinds of filters, an offline filter and an online filter. The offline filter uses features from a one-hop social network that models longer-term sender behaviour while the online filter focuses on how many receivers a sender has sent to in a given time period which is extracted from a two-hop social network and combined with temporal spectral analysis of submission behaviour. They suggest that their approaches can be combined with content-based approaches either serially, where results of independent filter systems can be combined, or sequentially where the behaviour-based filter can provide input to the content-based system or vice versa.

3.3. Spam filtering in other short text classification domains

There has been other relevant spam classification work recently in related short text message domains. There is significant evidence of spam in social networks including instant message spam (aka spim) and twitter spam. Typically, fake or bot accounts are used to automatically send messages or tweets that contain links that can be used to gather marketing information or for more malicious or phishing purposes. It has been recently reported that just 35% of the average Twitter users' followers are real people.[8]

Most of the published research into spim or Twitter spam filtering typically tries to identify the spammer or bot that is generating the messages rather than actually identify the message as a spam message. A variety of techniques are used, with most approaches combining two or more techniques, generally including blacklisting or blocking. The characteristics used to identify spam are usually based on the behaviour of the user because the expectation is that bots will behave in a significantly different manner to human users. Twitter offers additional challenges as a number of twitter users follow but do not tweet themselves, which can give them a behaviour profile like spammers[8]. Due to the emphasis on the identification of the bot user network analysis approaches are also popular in these domains where the node in the network represents the user in the social network and the edges represent friends or followers of the user (Yardi, Romero, Schoenebeck, & Boyd, 2010; Wang, 2010; Gao et al., 2010).

Supervised learning algorithms are used in these domains but there is very little work that uses the actual textual content of the message. User-based properties such as the number of social network friends or Twitter followers or followees (Yardi et al., 2010; Benevenuto, Magno, Rodriques, & Almeida, 2010; Gianvecchio, Xie, Wu, & Wang, 2008; Wang, 2010; Stringhini, Kruegel, & Vigna, 2010) or the posting behaviour (Chu, Gianvecchio, Wang, & Jajodia, 2010; Castillo, Mendoza, & Poblete, 2011) are common. Also popular are features based on the characteristic elements of such domains such as the number of hashtags, mentions (uses of other usernames – '@username') and urls in a message (Wang, 2010; Benevenuto et al., 2010; Stringhini et al., 2010), or message

[8] http://www.popularmechanics.com/technology/how-much-of-twitter-is-spam.

characteristics such as number of resends/retweets (Benevenuto et al., 2010).

Gianvecchio et al. (2008) do use the textual message content in a supervised learning component which is proposed as one element in an ensemble approach to spim filtering. This component uses a Naive Bayes classifier with orthogonal sparse bigrams of words as the message representation. This solution is also adopted by Chu et al. (2010) for Twitter spam filtering. Liu, Lin, Li, and Lee (2005) also investigate using a Naive Bayes classifier with standard word tokenisation for spim filtering but report this approach as not effective possibly due to the short length of the messages. Decision trees (Castillo et al., 2011; Maaroof, 2010), SVMs (Benevenuto et al., 2010) and random forests (Stringhini et al., 2010) have been shown to be effective in these short text domains but always using the more typical user or message-based properties as features rather than the actual textual content.

The datasets used are collected and labelled by the researchers individually and to date there are no benchmark results or public datasets available from this research.

4. SMS spam data

Any supervised machine learning approach such as those mentioned above is very dependent on the quality and quantity of the training data which is available to it. Good spam filtering using text classification methods needs to have representative, accurate, timely corpora of spam and non-spam messages with which to train the algorithms. In the email world, a number of different corpora are available, including the SpamAssassin corpus[9] or the TREC email corpora.[10] Table 2 in Guzella and Caminhas (2009) gives a good overview of the benchmark email spam datasets available and the number of works that have used them.

For SMS spam filtering however, there are few corpora available to-date which would allow independent corroboration of research results. This is understandable for a number of reasons. SMS, whether spam or non-spam, passes through proprietary networks run by private companies who are reluctant or unable to make their customers' data available for research purposes. SMS spam filtering is also in its relative infancy compared to email spam filtering, so many research projects may not have reached a point where they can make their data publicly available.

The main method to date of collecting SMS data is to ask mobile users to contribute text messages voluntarily. This method has been used primarily to collect legitimate SMS text messages for research into linguistic analysis, producing the ICT corpus[11] (Shortis, 2000) which is a collection of 202 messages in British English, the NUS corpus[12] (How & Kan, 2005), an ongoing collection of messages in Singaporean-influenced English which currently contains 28,268 items, and the SMS4Science corpus[13] (Beaufort, Roekhaut, Cougnon, & Fairon, 2010; Fairon & Paumier, 2006), the product of a continuing SMS collection project by universities in French-speaking regions whose current release contains 29,979 messages. A slightly different approach was used by Rafique and Farooq (2010) to collect legitimate SMS messages for spam research. They provided software to access the memory at the baseband processor of a mobile phone and redirect all messages in order to collect them.

There also have been more questionable examples of collecting legitimate messages such as the manual extraction of the text of SMS messages from a PhD thesis (Tagg, 2009) where the collected corpus for the PhD work had not been made directly available (Taufiq, bin Abdullah, Kang, & Choi, 2010).

SMS spam messages have also been collected by asking for contributions from mobile users. Researchers at the Indraprastha Institute of Information Technology in India have collected a dataset of SMS spam using crowdsourcing where students on campus were incentivised to forward unique SMS spam messages to a SMS server (Yadav et al., 2011). This proved successful, with 4000 spam messages (half of which were unique) being collected in 2 months. This is an interesting collection of SMS data as it contains cross-lingual examples, with a large proportion of the messages in the collection containing both Hindi and English words, and is due to be publicly released.

A quicker method of collecting SMS spam messages has been by scraping consumer complaint websites such as GrumbleText[14] which facilitate the reporting of unwanted or possibly fraudulent SMS text messages. This method of data collection has been popular with a number of researchers (Dixit et al., 2005; Gómez Hidalgo et al., 2006; Junaid & Farooq, 2011; Rafique & Farooq, 2010; Almeida et al., 2011).

Recently the *SMS Spam Collection* has been made publicly available[15] (Almeida et al., 2011), which is an extension of a corpus previously compiled by Gómez Hidalgo et al. (2006). It consists of 747 spam messages manually extracted from GrumbleText, 450 non-spam messages taken from Tagg's PhD thesis (Tagg, 2009), and finally 4,377 non-spam messages randomly sampled from the NUS corpus. This is the first benchmark dataset available, however whether it is a representative corpus of SMS data is somewhat questionable. While the spam data is in British English and is drawn from a single source, the non-spam is a combination of data from two very disparate sources. The NUS data is strongly influenced by Singaporean English, using particles such as "lor" or "lah" which do not occur in British English. Datasets of this nature are unlikely to occur naturally. In addition, the distribution of spam and non-spam in the corpus is totally arbitrary, with 13.4% spam. The actual distribution of spam can only be found by analysing a full stream of SMS traffic.

The authors perform a duplicate analysis using word n-grams of length 5 and 6 in order to detect repeated messages introduced by the extension of their original corpus with new data. However despite this analysis the 747 spam messages include 94 messages which are exact duplicates and a further 14 messages which are near duplicates, (i.e. exact matches after whitespace has been removed). These near duplicates will be, in effect, exact duplicates if standard bag of words tokenisation is used.

4.1. Collection of an SMS spam corpus

In order to investigate the nature of current SMS spam, we have collected a corpus of SMS spam messages[16] by scraping messages from two public consumer complaints websites: GrumbleText and WhoCallsMe.[17] GrumbleText has the advantage that users can report spams by forwarding them to a shortcode, which means that the original form of the message is preserved and no errors are introduced when messages are retyped. The website marks up the spam text explicitly, so we were able to extract each one automatically. Removing duplicates resulted in a corpus of 571 unique spam SMS messages.

Users at the WhoCallsMe website report unsolicited calls and text messages, which are then indexed by source phone number. There is no requirement to include the text of the SMS in the post, but many users do so. We scraped all entries within the UK mobile

[9] http://spamassassin.apache.org/publiccorpus/.

[10] http://plg.uwaterloo.ca/gvcormac/spam/.

[11] http://www.demo.inty.net/app6.html.

[12] http://wing.comp.nus.edu.sg:8080/SMSCorpus/.

[13] http://www.sms4science.org/.

[14] http://www.grumbletext.co.uk/.

[15] http://www.dt.fee.unicamp.br/tiago/smsspamcollection/.

[16] http://www.dit.ie/computing/research/resources/smsdata/.

[17] http://whocallsme.com/.

Please cite this article in press as: Delany, S. J., et al. SMS spam filtering: Methods and data. *Expert Systems with Applications* (2012), doi:10.1016/j.eswa.2012.02.053

prefix space, that is, 075- to 079-. From these entries we used a positive word list (“sms”, “text”, “txt”, “message”, “msg”) and a negative word list (“missed call”, “called me”, “voice”, “caller”) to find posts which most likely refer to SMS. We then extracted any strings which were between quotes as candidate SMS text, since users usually quoted their messages this way. We discarded any candidates of length less than 10 words (whitespace separated substrings). Finally we inspected the list of candidates to remove obviously erroneous entries, such as foreign language text, mismatched quotes, or user-added parentheses, resulting in a list of 737 items. Because many entries occur more than once we removed any duplicates based on string comparison after lowercasing and deletion of whitespace to mitigate user transcription errors, leaving a corpus of 436 messages overall.

We took the following steps to assemble a single corpus from the spam component of Almeida et al.’s (2011) SMS Spam Collection (hereafter SSC) together with our two sources. We first removed the 108 duplicate messages from SSC, leaving 639 unique messages. The union of our two sources contains 1007 messages, from which we removed three further duplicates. We then removed any message from our data which was already present in the SSC, again based on string comparison after whitespace removal. There were 290 such messages, all as expected in the GrumbleText part of our corpus. This left 714 messages, which were added to the 639 SSC messages, resulting in a data set of 1353 messages in total which contains no duplicates.

Each message extracted from GrumbleText and WhoCallsMe is stamped with the date it was reported on, and the corpus covers the period from late 2003 to the middle of 2010. It can therefore be considered an up-to-date corpus of SMS spam. In addition, all of the data occurred in the same linguistic region, since all messages had originally been received by UK mobile users.

Although we have removed duplicates from the data, many of the non-matching messages may still be close matches, since SMS spam, like email spam, is characterised by obfuscation. An example is the following message, which occurs 41 times in total in the corpus, each time differing only by phone number, claim number and possibly punctuation:

> URGENT! We are trying to contact U. Todays draw shows that you have won a £800 prize GUARANTEED. Call 09050003966 from land line. Claim S76. Valid 12hrs only.

Similarly the following attack occurs 97 times using slightly different phrasings, amounts of money and punctuation. For 17 of these the only difference is in the four characters which are appended to the end of the message text.

> You may be entitled to 6000 lb compensation for the Accident you had. To claim for free reply with YES to this msg. To opt out text stop. VLUJ.

This latter example provides evidence of concept drift in SMS spam, in particular an extension of what is known as *word salad* in email spam where random text can be added to the end of the message to make each spam message different in order to frustrate Bayesian and fingerprinting filters.

4.2. Analysis of an SMS spam corpus

With the goal of analysing and identifying different categories of SMS spam, we have performed a clustering experiment on the corpus presented in the previous section. The raw documents were parsed and processed according to the standard unigram-based text clustering practices. We employed a stop-list containing 499 entries to remove common functional words. For privacy purposes we also removed all references to phone numbers. We applied basic frequency-based term selection to remove terms occurring in less than three documents, and standard log-based TF-IDF to weight individual terms. This resulted in a vector space model representation of 1353 messages using 894 terms.

To cluster the messages, we attempted to divide the data into a flat, disjoint partition via spectral clustering methods (Ng, Jordan, & Weiss, 2001). We computed a normalised linear kernel matrix on the vector representation of the data (i.e. cosine similarity), and applied k-way spectral clustering with orthogonal initialisation which has previously been shown to be effective on a number of different types of data (Ng et al., 2001). This algorithm involves computing the truncated eigenvalue decomposition of the kernel matrix described above, and applying k-means in the reduced dimension space. To produce human-readable labels for clusters, we select the top-ranked terms in the centroid vectors of the clusters when projected back to the original space, as proposed by Dhillon and Modha (2001).

We experimented with a range of values for the number of clusters $k \in [5,15]$. Standard internal and stability-based validation methods did not suggest a specific “correct” value for k. Based on manually inspecting the cluster labels, we selected $k = 10$. Values of $k < 10$ tended to obscure more specific groups of messages, while values of $k > 10$ consistently produced clusters with highly-similar labels, suggesting “over-clustering”.

Table 1 shows the top terms for the clusters identified for $k = 10$, together with a set of manually-annotated cluster names based on the top terms. We can see that a number of distinct clusters are apparent, including groups of messages pertaining to potential “phishing” financial spam, mobile ringtones, and dating services.

Fig. 1 shows the fraction of messages assigned to each of the cluster sizes in the full set of 1353 messages.

We see from Fig. 1 that groups of SMS messages pertaining to mobile products (the “ringtones” and “services” clusters) and competitions (the “competitions” and “prizes” clusters) appear to be most prominent in the data. To investigate the relations between these clusters, we examine the similarities of all pairs of clusters based on the cosine similarity between their centroids.

Fig. 2a shows a heatmap view of the matrix of pairwise similarities of the clusters in the original sparse vector space – relatively little similarity is evident between the groups. When we examine their corresponding similarities in the low-dimensional embedded spectral space used for clustering (see Fig. 2b), as we might expect, we see a greater degree of similarity. This is particularly the case for the large “ringtones” and “competitions” clusters, though we also see a relatively high level similarity between the “ringtones” and “services” clusters, which is perhaps unsurprising. In Fig. 2b we can also observe a few distinct outlying clusters, such as “claims” and “voicemail”, containing messages that are considerably different from those in the other clusters.

Table 1
Ten clusters produced by applying spectral clustering to the SMS message dataset, with their associated top 8 terms and a putative annotation. Clusters are listed in descending order of size.

Annotation	Top terms
Ringtones	Send, ringtone, text, tone, free, sms, reply, mobile
Claims	Accident, entitled, records, pounds, claim, msg, compensation, opt
Competitions	Txt, win, uk, voucher, cash, 150p, send, entry
Prizes	prize, guaranteed, urgent, todays, valid, claim, draw, cash
Voicemail	Please, message, voicemail, waiting, call, delivery, immediately, urgent
Dating	Dating, service, contacted, find, guess, statement, points, private
Services	Mins, video, free, camera, orange, latest, phone, camcorder
Finance	Help, debt, credit, info, government, loans, solution, bills
Chat	Naughty, ring, alone, chat, xx, heard, luv, home
Miscellaneous	Find, secret, admirer, special, looking, r^* reveal, contact, call

Please cite this article in press as: Delany, S. J., et al. SMS spam filtering: Methods and data. *Expert Systems with Applications* (2012), doi:10.1016/j.eswa.2012.02.053

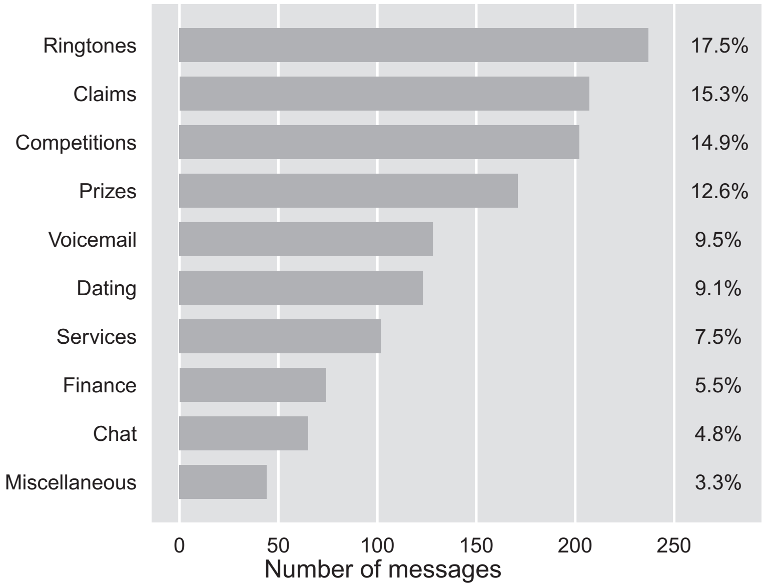


Fig. 1. Distribution of cluster sizes, for ten clusters produced by applying spectral clustering to the SMS message data.

When we compare these clusters to the types of spam identified by the GSMA (GSMA, 2011b), we find a close correspondence to the three main types which are described as,

(i) *SMS spam*, where unsolicited text messages are sent to subscribers for mass advertising and social engineering viral hoaxes;
(ii) *premium rate fraud* which is sending unsolicited text messages that trick subscribers into calling premium rate numbers or signing up for subscription services that are charged to their bill and
(iii) *phishing/smishing* which is sending unsolicited text messages asking subscribers to call certain numbers to extract confidential information, which is then used for other purposes.

The most common type in our dataset is premium rate fraud, which includes the clusters claims, prizes, voicemail, dating, and chat, and accounts for 43.9% of the messages. The ringtones and competitions clusters can be categorised as SMS spam and account for 32.4% of the messages whereas phishing attacks, which correspond to the services and finance clusters, account for 13.0% of the messages. Considering the reported rise in phishing and smishing attacks in recent times[18] this is a relatively low figure.

The data also includes instances of Value Added Service Provider (VASP) abuse which is unsolicited messages sent to subscribers from services providers for marketing purposes but not in sufficient volumes to be reflected in this clustering experiment. VASP abuse is also identified by the GSMA as a distinct type of SMS spam.

We have generated a number of baseline results by applying state-of-the-art techniques to our corpus. Using LibSVM (Chang & Lin, 2011) we have implemented the linear support vector machine and alphanumeric tokenisation which was found by Almeida et al. (2011) to achieve the best performance on their corpus. We intentionally do not include any non-spam data in our corpus because of the issues outlined above about spam distribution and linguistic regions. For the purposes of these benchmarks however we have combined our spam data with 652 unique non-spam messages—taken from Taufiq et al. (2010) and Shortis (2000)—to create a dataset of British English SMS. We took these two non-spam sources and combined them with ten random subsamples of the spam corpus, resulting in ten balanced datasets of 1304 messages. Ten-fold cross validation on this data resulted in a mean accuracy of 94.63% (sd = 0.6%), a spam recall of 93.31%, and a false positive rate of 4.05%. The resulting models contain on average 350.8 support vectors. We have carried out this SVM baseline experiment on a standard workstation (Linux, 3.16 GHz dual core processor, 4 GB RAM), and find that we can classify messages continuously in under 2 ms per message.

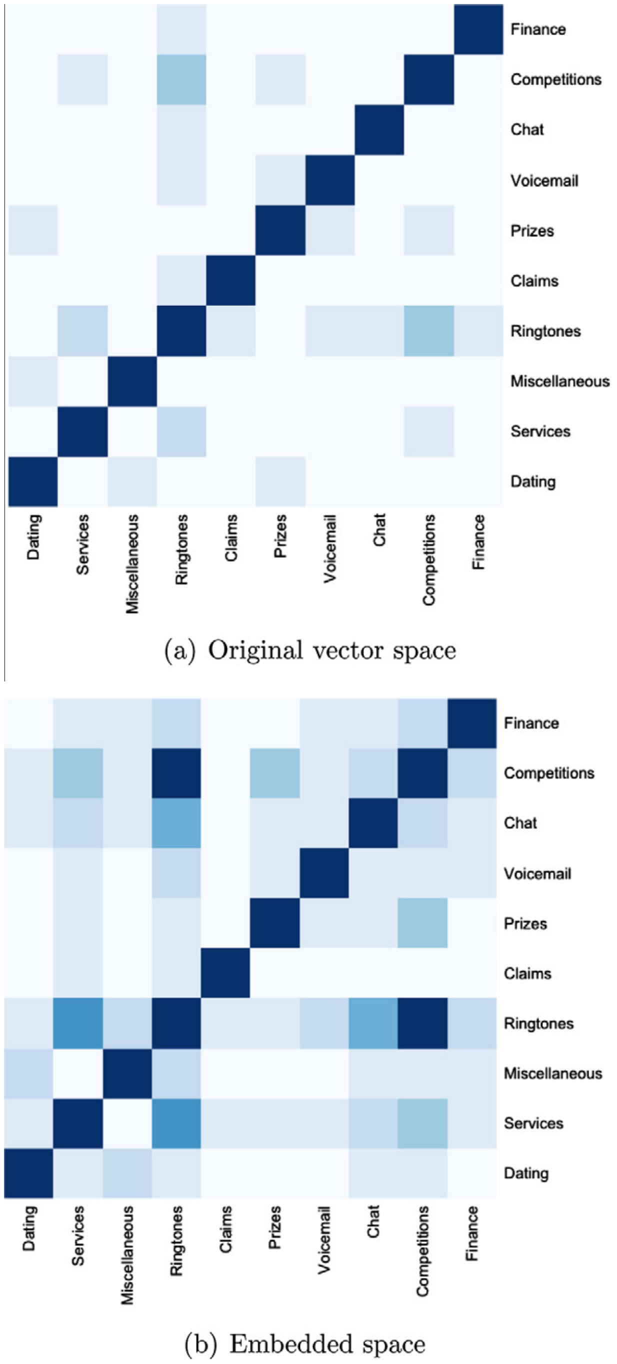
We also examined such a model's performance in terms of spam recall in each of the individual clusters. For this we constructed a data set containing all 1353 spam messages and the 652 non-spam messages as above. We performed a single run of 10-fold cross validation and partitioned the results according to cluster membership, which are summarised in Table 2.

The results show that in seven of the nine clusters other than miscellaneous, 96% or more of the messages are correctly categorised as spam, with a detection rate of as high as 99.4% for the prizes cluster. The two remaining clusters, ringtones and chat, have considerably lower spam recall but are the largest and smallest cluster, respectively, suggesting that the poorer performance is not just due to a smaller number of training instances. In Section 4.1 we discussed near-duplicate messages, citing frequently-occurring examples from the prize and claims clusters. Clusters which contain multiple occurrences of the same spam attack define a tighter, more homogeneous concept, and therefore a classifier will find these clusters easier to predict correctly. This is reflected in the results in Table 2, where claims and prizes are among the best-performing categories. The lower recall rate for the ringtones cluster may be due to it being a more diverse category.

As a final baseline result we implemented the collaborative filtering approach proposed by Liu and Fang (2008) for email and applied it to our SMS data. In this approach a unique fingerprint can be computed from the text of a message. The fingerprint of each new message is compared to the fingerprints of all known spam SMS, and if it is similar to any spam message fingerprint then the message is considered spam. Using ten randomised runs of 10-fold

[18] http://www.fbi.gov/news/stories/2010/November/cyber_112410/cyber_112410, http://www.rsa.com/solutions/consumer_authentication/intelreport/11244_Online_Fraud_report_0111.pdf.

Please cite this article in press as: Delany, S. J., et al. SMS spam filtering: Methods and data. *Expert Systems with Applications* (2012), doi:10.1016/j.eswa.2012.02.053



(a) Original vector space

(b) Embedded space

Fig. 2. Heatmap of pairwise cluster similarities calculated in (a) the original vector space, (b) the space constructed from a 10-dimensional spectral embedding of the original vector space. In both cases, a darker colour is indicative of a higher degree of similarity.

cross validation as described above, this method achieves on average 79.09% accuracy, 58.24% spam recall and 0.06% false positives.

5. Discussion

This paper has presented the state of the art in SMS spam filtering and has reviewed a number of different approaches to the problem which have been suggested and tested. Using different data sets, various researchers have shown that supervised learning algorithms can be effective for SMS spam classification, with reported accuracies of up to 97%. There is also some evidence of the use of non content-based approaches such as social network analysis and the identification of patterns of SMS submission.

Table 2
Size and spam recall results for each cluster.

Cluster name	Size	Spam recall (%)
Ringtones	237	91.1
Claims	207	97.5
Competitions	202	96.0
Prizes	171	99.4
Voicemail	128	96.9
Dating	123	97.6
Services	102	97.1
Finance	74	98.7
Chat	65	87.7
Miscellaneous	44	95.5

To date, many of the proposed approaches are centrally based but as an alternative to server-side classifiers, SMS filtering on the client device has the potential benefit of being independent of the network and the operator's spam policy, and can filter based on the user's personal concept of what is spam. It does however introduce additional technical restrictions such as available processing power and the need for a programmable device. Combinations of server-side and client-side filtering are also possible.

Like with any machine learning task, feature engineering is key, and this is especially true of SMS filtering because there is less content material in the source than in other text classification tasks. Feature selection has also been highlighted by many researchers due to the high dimensionality of the feature space.

We motivated the issue of SMS spam mainly in relation to its potential damaging impact on consumers and mobile networks. This raises the question of what the requirements of an SMS spam solution should be. The research reviewed here is evaluated using scientific metrics such as accuracy, F-score and false positive rate. In addition to these, any complete SMS spam solution must consider the requirements of an industrial-strength application. Speed of processing for instance is crucial, as SMS has become a *de facto* real-time, dependable service for applications such as online banking, and research into client-side filtering identifies processing time as an important constraint. Simple solutions such as blacklisting and spoofing/faking detection are currently being deployed, however these are by their nature brittle, do not take the content of the message into account, and require ongoing management. Content-based solutions such as those reviewed here have the potential to be more accurate and flexible.

The results of the work published to date indicate that there is as yet no consensus on what the best techniques are for SMS spam filtering. Overall, the techniques which have been used to date are quite straightforward, applying what has been used in text classification in general to SMS filtering, and not necessarily taking the specific characteristics of SMS into account. One reason for this is simply the relative infancy of the field. Only recently have the ubiquity of SMS and the falling cost of delivery attracted the interest of spammers, so there has not yet been much time for academic research to identify and define the problem. A more important reason for the lack of consensus is that it is hard to compare and contrast research done on radically different datasets. These datasets vary greatly within the work reviewed here: by language, where English and Chinese dominate, by size, from a few hundred instances to a few thousand, and by method of collection. The distribution of categories also varies, from even numbers of spam and non-spam to arbitrarily chosen small proportions of spam.

Two methods of data collection have provided the data sets for research on SMS spam filtering. Harvesting data from online, user-driven sources is fast, but the available data is limited. For instance

Please cite this article in press as: Delany, S. J., et al. SMS spam filtering: Methods and data. *Expert Systems with Applications* (2012), doi:10.1016/j.eswa.2012.02.053

a number of researchers have duplicated the work of scraping SMS data from the GrumbleText website. This method potentially results in an unrepresentative sample, as only some users feel motivated to report the spam they have received. Eliciting voluntary contributions from users is a more time- and resource-intensive method of data collection, but can result in larger data sets. Whether this method results in a more representative sample though is questionable. Consider, for example, the SMS4Science non-spam data collection in Switzerland, although 23,988 SMS were collected, 80 people donated more than 50 messages each, and one person even donated 413 messages. One notable success here is the work of Yadav et al. (2011), who have collected 2,000 unique spam SMS in the space of two months.

The difficulty and expense of collection spam collection means that the work reviewed here is based on data sets whose size is in the order of a few thousand instances at best. This compares poorly to other work in the field of text classification in general, where corpora containing millions of documents are available, for instance the Enron email data set of 1.5 m emails (Klimt & Yang, 2004) or the New York Times Annotated Corpus, which contains 1.8 m news articles (Sandhaus, 2008). It may not be possible to make SMS data freely available if it has been collected in collaboration with an industrial partner, as there are often strict privacy restrictions.

In own our work we have taken the first approach to data collection, and have assembled a corpus of 1,353 unique SMS spam messages from a number of online sources. It represents a larger sample of the same kinds of SMS as have been used in research on English language SMS spam which is reviewed here. In our analysis we examined the types of spam using content-based clustering, identifying ten clearly-defined clusters. This may reflect the extent of near-repetition in our data caused by the similarity between different spam attacks and the breadth of obfuscation used by spammers. In terms of the topics covered by the spam data, it shows that SMS is providing an additional channel for modern email-based attacks such as phishing, advertising and premium rate fraud.

In terms of sophistication however, SMS spam is comparable to early email spam. It is not personalised, obfuscation is limited and little effort is made to hide the true content of the message. This is of course in part due to the text-only nature of SMS.

Data may well be expensive to collect, but it seems that it is easier in regions where there is more spam in the wild. The largest data sets in the literature have been collected by researchers working in India and China, two countries where the incidence of spam is highest. This indicates that spam must reach a certain level before it can be reliably collected, and that this level may not yet have been reached in English-speaking regions.

Despite the undoubted initial progress already made on this difficult problem, we can identify a number of challenges and directions which are apparent now and which will become more important as the sophistication of spam increases and as the need for anti-spam solutions leads to more real-world deployments.

Multilingual environments: Mobile networks are language independent and, especially in multiple-language regions, they handle SMS in a mixture of languages. However, all of the research on SMS spam filtering up to now has used single-language data sets, and the use of word tokenisation introduces an intrinsic restriction to the language portability of filtering solutions. Due to the short length of the message content and the lack of clear identifying information in the headers it can be difficult to identify the language of a particular message. Robustness in multilingual environments will be a key requirement of deployed spam filters.

Shared data: Research will benefit greatly if there is a common spam SMS data set which is representative and sufficiently large. We have seen that collection of such a data set is possible for non-spam, and the publication of spam corpora will hopefully lead to alignment on a shared data resource. Such a development has already been beneficial for researchers on email spam filtering with the publication of the TREC corpora and the associated evalution toolkit, and it would make research results more readily interpretable and repeatable.

Hybrid solutions: With spam filtering there is no single solution that works. It is likely that some types of SMS spam can be better filtered by certain methods, so similar to the email domain, we see hybrid solutions as a promising avenue. Given that SMS filtering must happen under very strict processing time restrictions, content-based and collaborative filters could be usefully augmented with simple, less resource-intensive filtering methods such as blacklisting or traffic profiling.

Advanced address-based filtering: The move in recent times in email spam filtering has been towards advanced address-based filtering approaches including social network analysis and reputation-based filtering. These techniques should be considered in the mobile domain also but the lack of adequate data will hamper such efforts.

Scalability and real-world deployment: The work reviewed here represents research prototypes and solutions prepared under controlled laboratory conditions. Our benchmark implementation, which classifies messages in under 2 ms on average, is within the requirements for a system handling real-life SMS traffic volumes in terms of messages per second per node. This indicates that a filter based on support vector machines may be a feasible solution. For any real-world deployment however, the issues of scale and robustness become crucial, and high-speed databases, clustering, as well as efficient data structures and implementations will be required.

Industry collaboration: Progress in this field will have to be validated by real-world trial deployments, which only the network operators can facilitate. As volumes of spam increase, the promise of content-based filtering should make such collaboration attractive to industry.

We also see a number of positives in the current state of research, and much potential for further advances. Overall there are many candidate technologies and many competing solutions, and the best solution may well turn out to be a combination of these.

The state of mobile handset technology means that SMS spam filtering will continue to be needed. Smartphones represent an ever growing segment of the handset market, and in this segment it is likely that messaging technology will converge, unifying SMS, email and other message types. However non-smartphones (known as “feature” phones) are still the majority, and they continue to be launched in developing markets, for instance in Kenya (Maina, 2010). Feature phones do not have the functionality to support user interaction in the spam filtering process. They do not have separate spam inboxes and they can not run third-party spam filtering software. This means that centralised SMS spam filtering will continue to be in demand as the volume of mobile spam increases.

References

Almeida, T. A., Gómez Hidalgo, J. M., & Yamakami, A. (2011). In *Proceedings of the 11th ACM Symposium on document engineering DOCENG'11* (pp. 259-262). Mountain View, CA, USA: ACM.

Beaufort, R., Roekhaut, S., Cougnon, L. A., & Fairon, C. (2010). A hybrid rule/model-based finite-state framework for normalizing SMS messages. In *Proceedings of the 48th annual meeting of the association for computational linguistics ACL '10* (pp. 770–779). Stroudsburg, PA, USA: Association for Computational Linguistics.

Benevenuto, F., Magno, G., Rodriques, T., & Almeida, V. (2010). Detecting spammers on twitter. In *Procs of the 7th annual collaboration electronic messaging, anti-abuse and spam conference*.

Please cite this article in press as: Delany, S. J., et al. SMS spam filtering: Methods and data. *Expert Systems with Applications* (2012), doi:10.1016/j.eswa.2012.02.053

Boykin, P. O., & Roychowdhury, V. P. (2005). Leveraging social networks to fight spam. *IEEE Computer, 38*, 61–68.

Cai, J., Tang, Y., & Hu, R. (2008). Spam filter for short messages using winnow. In *Proceedings of the international conference on advanced language processing and web information technology* (pp. 454–459). IEEE.

Castillo, C., Mendoza, M., & Poblete, B. (2011). Information credibility on twitter. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, & R. Kumar (Eds.), WWW (pp. 675–684). ACM.

Chang, C. C., & Lin, C. J., 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2 (27), 1–27. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.

Chu, Z., Gianvecchio, S., Wang, H., & Jajodia, S. (2010). Who is tweeting on twitter: human, bot, or cyborg? In Gates, Franz, and McDermott, 2010.

Cormack, G. V., Gómez Hidalgo, J. M., & Sánz, E. P. (2007). Spam filtering for short messages. In *Proceedings of the 16th ACM conference on conference on information and knowledge management CIKM '07* (pp. 313–320). New York, NY, USA: ACM.

Deng, W.-W., & Peng, H., 2006. Research on a Naive Bayesian Based Short Message Filtering System. In Proceedings of the international conference on machine learning and cybernetics (pp. 1233–1237). IEEE.

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning, 42*, 143–175.

Dixit, S., Gupta, S., & Ravishankar, C. V. (2005). LOHIT: An online detection & control system for cellular SMS spam. In *Proceedings of the IASTED international conference on communication, network and information security* (pp. 48–54).

Fairon, C., & Paumier, S. (2006). A translated corpus of 30,000 French SMS. In *Proceedings of language resources and evaluation*.

Gao, H., Hu, J., Wilson, C., Li, Z., Chen, Y., & Zhao, B. Y. (2010). Detecting and characterizing social spam campaigns. In E. Al-Shaer, A. D. Keromytis, & V. Shmatikov (Eds.), *ACM conference on computer and communications security* (pp. 681–683). ACM.

Gates, C., Franz, M., & McDermott, J. P. (Eds.), 2010. *Twenty-sixth annual computer security applications conference*. ACSAC 2010, Austin, Texas, USA, 6–10 December 2010. ACM.

Gianvecchio, S., Xie, M., Wu, Z., & Wang, H. (2008). Measurement and classification of humans and bots in internet chat. In P. C. van Oorschot (Ed.), *USENIX security symposium* (pp. 155–170). USENIX Association.

Gómez Hidalgo, J. M., Bringas, G. C., Sánz, E. P., & García, F. C. (2006). Content based SMS spam filtering. In D. Bulterman, & D.F. Brailsford (Eds.), *Proceedings of the 2006 ACM symposium on document engineering DocEng '06* (pp. 107–114). New York, NY, USA: ACM.

Graham-Cumming, J. (2005). Why I hate challenge-response systems. In *JGC's spam and antispam newsletter*. <http://www.jgc.org/antispam/02282005-60dfea1d4f36a4071c21d1ba86f5e988.pdf> 28.02.05.

GSMA (2011a). Operator FAQs. *GSMA spam reporting service*.

GSMA (2011b). SMS spam and mobile messaging attacks - Introduction. Trends and examples. *GSMA spam reporting service*.

Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to spam filtering. *Expert Systems with Applications*, 10206–10222.

He, P., Wen, X., & Zheng, W. (2008). A novel method for filtering group sending short message spam. In *Proceedings of the international conference on convergence and hybrid information technology* (pp. 60 –65).

Healy, M., Delany, S., & Zamolotskikh, A. (2005). An assessment of case-based reasoning for short text message classification. In N. Creaney (Ed.), *Proceedings of 16th Irish conference on artificial intelligence and cognitive science, (AICS-05)* (pp. 257–266).

How, Y., & Kan, M.-Y. (2005). Optimizing predictive text entry for short message service on mobile phones. In M. J. Smith & G. Salvendy (Eds.), *Proceedings of human computer interfaces international (HCII 05)*. Lawrence Erlbaum Associates.

Hu, X., & Yan, F. (2010). Sampling of mass SMS filtering algorithm based on frequent time-domain area. In *Proceedings of the third international conference on knowledge discovery and data mining* (pp. 548 –551).

Jie, H., Bei, H., & Wenjing, P. (2010). A Bayesian approach for text filter on 3G network. In *Proceedings of the 6th international conference on wireless communications networking and mobile computing* (pp. 1–5).

Junaid, M. B., & Farooq, M. (2011). Using evolutionary learning classifiers to do mobile spam (SMS) filtering. In *Procs. of genetic and evolutionary computation conference (GECCO 2011)*.

Klimt, B., & Yang, Y. (2004). The enron corpus: A new dataset for email classification research. In J.-F. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Machine learning: ECML 2004. Lecture notes in computer science* (Vol. 3201, pp. 217–226). Berlin/ Heidelberg: Springer.

Kobus, C., Yvon, F., & Damnati, G. (2008). Normalizing SMS: Are two metaphors better than one? In *Proceedings of the 22nd international conference on computational linguistics* (Vol. 1, pp. 441–448). Association for Computational Linguistics.

Lai, C.-C. (2007). An empirical study of three machine learning methods for spam filtering. *Knowledge-Based Systems, 20*, 249–254.

yu Lam, H., & yan Yeung, D. (2007). A learning approach to spam detection based on social networks. In *Proceedings of fourth conference on email and antispam CEAS 2007*.

Ling, R. (2005). The socio-linguistics of SMS: An analysis of SMS use by a random sample of Norwegians. In R. Ling & P. Pedersen (Eds.), *Mobile communications: Renegotiation of the social sphere* (pp. 335–349). London: Springer.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning, 2*, 285–318.

Liu, W., & Fang, W. (2008). Adaptive spam filtering based on fingerprint vectors. *ISECS international colloquium on computing, communication, control, and management* (Vol. 1, pp. 384–388). Washington, DC, USA: IEEE Computer Society.

Liu, W., & Wang, T. (2010). Index-based online text classification for SMS spam filtering. *Journal of Computers, 5*, 844–851.

Liu, Z., Lin, W., Li, N., & Lee, D. (2005). Detecting and filtering instant messaging spam: A global and personalized approach. In *Proceedings of the first international conference on secure network protocols NPSEC'05* (pp. 19–24). Washington, DC, USA: IEEE Computer Society.

Longzhen, D., An, L., & Longjun, H. (2009). A new spam short message classification. In *Proceedings of the first international workshop on education technology and computer science* (Vol. 2, pp. 168 –171).

Maaroof, U. (2010). Analysis and detection of SPIM using message statistics. In *6th International conference on emerging technologies (ICET)* (pp. 246–249).

Maina, W. (2010). Vodafone unveils low-cost mobile handsets. Business Daily. <http://www.businessdailyafrica.com/Corporate+News/-/539550/863416/-/xsf06c/-/index.html> 17.02.10.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing, 14*, 849–856.

Rafique, M. Z., & Farooq, M. (2010). SMS SPAM detection by operating on byte-level distributions using hidden markov models (HMMs). In *Proceedings of the 20th virus bulletin international conference*.

Rennie, J. (2000). IFILE: An application of machine learning to E-mail filtering. In *Proceedings of the KDD-2000 workshop on text mining, sixth ACM SIGKDD international conference on knowledge discovery and data mining*.

Sandhaus, E. (2008). The New York times annotated corpus. *Linguistic data consortium*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*, 1–47.

Shortis, T. (2000). *The language of ICT*. London: Routledge.

Sohn, D. N., Lee, J. T., & Rim, H. C. (2009). The contribution of stylistic information to content-based mobile spam filtering. In *Proceedings of the ACL/AFNLP 2009 conference short papers* (pp. 321–324).

Stringhini, G., Kruegel, C., & Vigna, G. (2010). Detecting spammers on social networks. In Gates et al., 2010.

Tagg, C. (2009). *A corpus linguistics study of SMS text messaging*. Ph.D. thesis University of Birmingham.

Taufiq, M., bin Abdullah, M. F. A., Kang, K., & Choi, D. J. (2010). A survey of preventing, blocking and filtering short message services (SMSs) spam. In *Proceedings of international conference on computer and electrical engineering* (Vol. 1, pp. 462–466). IEEE.

Tseng, C.-Y., & Chen, M.-S. (2009). Incremental SVM model for spam detection on dynamic email social networks. *CSE* (Vol. 4, pp. 128–135). IEEE Computer Society.

Wang, A. H. (2010). Don't follow me – spam detection in twitter. In S.K. Katsikas, & P. Samarati (Eds.), *SECRYPT* (pp. 142–151). Sciteress.

Wang, C., Zhang, Y., Chen, X., Liu, Z., Shi, L., Chen, G., et al. (2010). A behavior-based SMS antispam system. *IBM Journal of Research and Development, 54*, 3:1–3:16.

Wu, N., Wu, M., & Chen, S. (2008). Real-time monitoring and filtering system for mobile SMS. In *Proceedings of 3rd IEEE conference on industrial electronics and applications* (pp. 1319–1324).

Xiang, Y., Chowdhury, M., & Ali, S. (2004). Filtering mobile spam by support vector machine. In N. Debnath (Ed.), *Proceedings of the third international conference on computer sciences, software engineering, information technology, E-business and applications* (pp. 1–4).

Yadav, K., Kumaraguru, P., Goyal, A., Gupta, A., & Naik, V. (2011). SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering. In *Proceedings of the 12th workshop on mobile computing systems and applications (HotMobile 2011)*.

Yardi, S., Romero, D. M., Schoenebeck, G., & Boyd, D. (2010). *Detecting spam in a twitter network*. First Monday, 15.

Yoon, J. W., Kim, H., & Huh, J. H. (2010). Hybrid spam filtering for mobile communication. *Computers & Security, 29*, 446–459.

Zhang, L., Zhu, J., & Yao, T. (2004). An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP), 3*, 243–269.