$100,000 Prize Jackpot. Call Now! Identifying the Pertinent Features of SMS Spam

Henry Tan, Nazli Goharian, and Micah Sherr
Department of Computer Science, Georgetown University
Washington D.C.
ztan@cs.georgetown.edu, nazli@cs.georgetown.edu, msherr@cs.georgetown.edu

ABSTRACT

Mobile SMS spam is on the rise and is a prevalent problem. While recent work has shown that simple machine learning techniques can distinguish between ham and spam with high accuracy, this paper explores the individual contributions of various textual features in the classification process. *Our results reveal the surprising finding that simple is better*: using the largest spam corpus of which we are aware, we find that using simple textual features is sufficient to provide accuracy that is nearly identical to that achieved by the best known techniques, while achieving a twofold speedup.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

General Terms

Algorithms, Experimentation

Keywords

SMS, Spam, Short Text, Feature Selection, Messaging

1. INTRODUCTION

Spam detection has historically focused on email spam. More recently, with the ubiquity of mobile phones and the advent of Twitter and other social media, short text spam detection has garnered interest and importance. Our work focuses on SMS spam, which contains on the order of tens of characters. Various spam filtering methods have been examined in the SMS domain – for example, using challenge-response methods such as CAPTCHA [4] and a variety of content-based filters. Content-based filtering has shown significant promise and recent papers suggest that existing filters and methods for email spam are sufficiently accurate for identifying short text spam [1, 2, 5].

However, most of the work in this area operates with large, composite, and complex feature sets (e.g., part-of-speech, n-grams, etc.) without showing the contributions of any subset of the features.

In this paper, *we examine the individual contributions of different sets of text-based features to the classification process*. Our work provides insight into the characteristics of spam and what kinds of features would be good choices for SMS classification. Surprisingly, we find that using simple feature sets instead of those suggested by previous work provides comparable quality at increased speed.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
ACM 978-1-4503-1472-5/12/08.

2. METHODOLOGY

Cormack et al. showed that a variety of supervised learning methods paired with a set of features consisting of orthogonal sparse word grams, character n-grams, and words, works very well for short text spam detection [2]. They composed an English SMS corpus from multiple sources, generated the features mentioned above and showed that standard classifiers such as Naive Bayes and SVM performed extremely well. This dataset was used by Almeida et al. in a baseline study where they showed the performance of a large number of classifiers paired with two simple feature sets [1]. The feature sets used by Almeida et al. were based on simple tokenization of the text (e.g., splitting the text into tokens by punctuation and spaces and using each unique token as a feature).

In this paper, rather than picking a feature set and showing the performance of various classifiers, we examine the effects of various feature sets on the performance of a fixed SVM classifier.

Our work uses the publicly available corpus curated by Cormack et al., which allows us to use their work and the work done by Almeida et al. as a baseline. While the work done by Cormack et al. in creating, curating and hosting the spam corpus is admirable, its usage comes with a few caveats. An examination of the corpus showed a variety of issues which we try to mitigate with preprocessing (described in Section 3).

We generate standard sets of features (e.g., n-grams) suggested in previous work [1, 2]. We additionally experiment with novel "content matching" features (explained below). For all experiments, we measure the *classification quality* (accuracy, recall, precision, and F1 score) and the *computation cost* (the time required to compute the features and perform the classification).

3. EXPERIMENTAL FRAMEWORK

Normalization. We pre-process the corpus used by Cormack et al.[1] which contains 4827 ham messages and 747 spam messages, to change certain terms and characters to mitigate the effects of having spam and ham that were collected from independent geographic regions. Specifically, we converted '£' to '$', converted the words 'uk' and 'uks' (symbolizing the United Kingdom) to 'sg' (in order to standardize country references), and removed word and number-suffix instances of 'p' and 'ppm' (pence and pence per minute).

We then normalize the corpus, which was collected from multiple sources, to be consistent in its treatment of numbers. Certain sources had replaced the numbers with symbolic representations (presumably as a privacy-preserving measure). We replaced these representations, and all remaining numbers, with '⟨#⟩' and '⟨.⟩' to respectively denote numbers and decimal points.

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

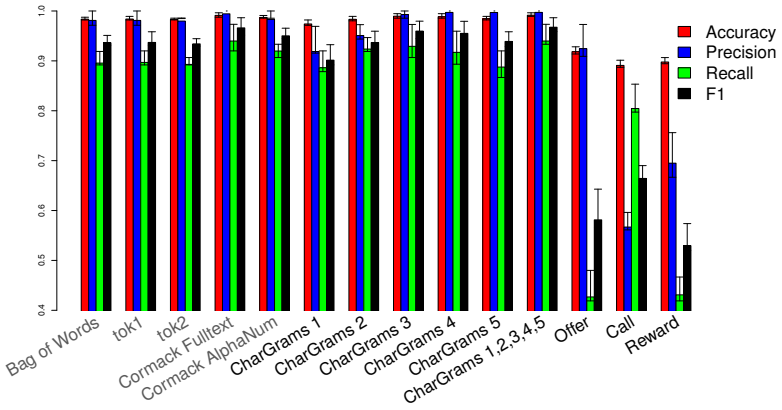

Figure 1: Average classification quality (accuracy, recall, precision, and F1 score) of the various feature sets. The first five feature sets (the baseline) have been suggested in previous work. Error bars indicate the interquartile range.

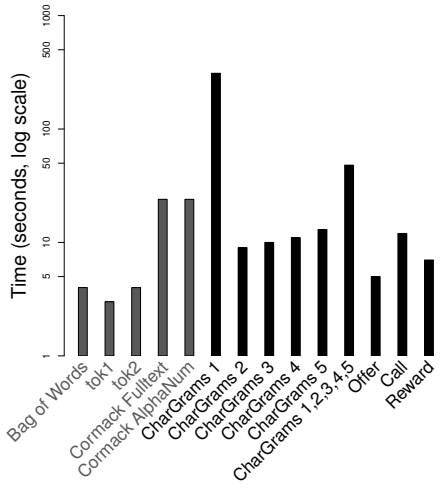

Figure 2: Computation time (in seconds) for computing the features and performing the classification using the various feature sets.

Feature Set Generation. We generate each of the following feature sets separately and in various combinations: word grams; character grams (ranging from $[1, 5]$-grams); and alphanumeric versions of the above n-grams (i.e., ignoring all non-alphanumeric characters). We additionally construct features for various regular expressions, which we hypothesized might be indicative of spam:

- **rate:** $(/|per)(\ |)$(year|month|hour|week|call)
- **reward:** free|award|prize|win|reward
- **website:** .co|.org|.net
- **call:** call|text|txt|msg|contact
- **offer:** **(call**$\vee$**website)**$\wedge$**(reward**$\vee$**rate)**

Note that **offer** is a logical combination of the previous four expressions. The **rate** expression can be interpreted as a percentage (either specified as a fraction or having the sequence `p-e-r`) optionally followed by a space and ending with a listed keyword.

Finally, we test a few statistical features: message length; the proportion of upper-case letters; and the proportion of punctuation.

The feature sets we generate are binary features (i.e., set to 1 iff the feature exists in the text) as recommended by Cormack et al. [2], except in cases in which the feature is inherently quantitative.

As a baseline, we also implement Cormack's feature selection method [2] and Almeida's *tok1* and *tok2* techniques [1].

Classification. We use *SVMLight* with stratified 10-fold cross validation and the recommended settings [3] to classify the corpus and record the resulting accuracy, precision, recall and F1 score.

4. RESULTS

Classification Quality. The classification quality achieved using the various feature sets is presented in Figure 1. Due to space constraints, we omit the features that result in poor classification.

We test two versions of the feature selection method used by Cormack et al. – one that ignores all non-alphanumeric characters and another that considers all characters – respectively labelled *Cormack AlphaNum* and *Cormack Fulltext* in the figure. Since the unstripped version produces slightly better results, we use that as our main baseline. Our results confirm that *Cormack Fulltext* is quite effective. We also note that none of *tok1* (F1: 93.698), *tok2* (F1: 93.409) or the standard bag-of-words model (F1: 93.625) produce classification quality on par with the highest performing feature sets.

Surprisingly, however, many of the simpler techniques achieve classification performance nearly identical to that of Cormack et al. [2]. In particular, using a feature set consisting of 1-, 2-, 3-, 4-, and 5-grams (*CharGrams 1,2,3,4,5*), we obtained an equivalent average F1 score to that of *Cormack Fulltext* (96.75 vs. 96.62). 3-grams (*CharGrams 3*) comes in third with an F1 score of 95.97.

The results also show that the regular expression and statistical features (e.g., proportion of upper case characters) that we tested are not good indicators of spam. The **offer** expression stands out with relatively good precision for a single feature. However, its recall is low due to the rule coverage, leading to an F1 of 58.132.

Computation Cost. Figure 2 plots the computation cost (in seconds) for computing the features and performing the classification across all 10 folds. As expected, the simpler techniques offer lower computation times. In particular, our results show that character 3-grams (*CharGrams 3*) produces comparable performance to *Cormack Fulltext* (F1: 95.97 vs. 96.62) at over twice the speed (9 vs. 24 seconds).

Mutual Information Study. We additionally examined the mutual information values of the various features. The string '$\langle\#\rangle$' offers the greatest mutual information, indicating that the presence of numbers is a very good indicator of spam. We also found that certain words show up in ham but not in spam; these consist mostly of spoken and text slang found in the geographic region of the ham dataset (Singapore). There are also words that show up in spam but not in ham (e.g., 'claim', 'URGENT', 'WON', 'YES' and 'prize'), as one would expect. We surmise that if one were able to obtain good coverage of such words to better formulate an **offer** expression, classification quality would further improve.

References

[1] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami. Contributions to the Study of SMS Spam Filtering: New Collection and Results. In *ACM DocEng*, 2011.

[2] G. V. Cormack, G. Hidalgo, E. P. Sánz, and J. María. Spam Filtering for Short Messages. In *CIKM*, 2007.

[3] D. Sculley and G. M. Wachman. Relaxed Online SVMs for Spam Filtering. In *SIGIR*, 2007.

[4] M. H. Shirali-Shahreza and M. Shirali-Shahreza. An Anti-SMS-Spam Using CAPTCHA. In *CCCM*, 2008.

[5] D. Sohn, J. Lee, and H. Rim. The Contribution of Stylistic Information to Content-based Mobile Spam Filtering. In *ACL-IJCNLP Conference*, 2009.