The Evaluation of Ordered Features for SMS Spam Filtering

José M. Bande Serrano, José Hernández Palancar[1] and René Cumplido[2]

[1] Advanced Technologies Application Center, 7ma A ♯ 21406, e/ 214 y 216, Siboney,
Playa, C.P. 12200, Havana, Cuba
jbande@cenatav.co.cu, jpalancar@cenatav.co.cu
http://www.cenatav.co.cu

[2] Instituto Nacional de Astrofísica Optica y Electrónica, Luis E. Erro 1, Sta. Ma.
Tonanzintla, Puebla, 72840, México
rcumplido@inaoep.mx

Abstract. In this work we propose a method to capture the writing style of spams and non-spam messages by preserving the sequentiality of the text in the feature space. To be more specific, we propose to build the feature vector considering the features apparition order in the text. We extract features from messages by applying three techniques: Extrinsic Information, Sequential Labeling Extraction and Term Clustering. In doing so, the method presents low dimensional feature space that shows competitive classification accuracy for the tested classifiers.

Keywords: Short Text Messages, SMS, Spam Filtering, Sequential Labeling Extraction, Extrinsic Information, Term Clustering.

1 Introduction

SMS spam filtering is a concern in much of the world. Still, spam-sms filtering has not receive much of the attention. The reason is maybe, the common believe that the problem can be solve by simply applying current email filtering methods, but the truth, is that there two core differences between both domains. The First is related with the promptness of the SMS transference system. In few words, different from email, SMS are expected to be delivered and read just after sent. Any SMS-spam filter running in the SMS central must deal with millions of SMS messages per day that cannot afford long analysis time. Therefore, the efficiency is a requirement for classification algorithms and for data representation.

The second concern is the lack of information. As indicated by the name, SMS are short texts, so there are not much data from where to extract discriminative features. The lack of discriminative or strong features has a negative influence in the classification accuracy. An strategy to overcome this problem is the use of complementary or external source of information. This is called Extrinsic Information (Cormack *et. al*) [3]. In this work we make use of Extrinsic Information to improve the features quality. Most of the feature extraction methods used in SMS classification assumes statistic independency between features, and they

E. Bayro-Corrochano and E. Hancock (Eds.): CIARP 2014, LNCS 8827, pp. 383–390, 2014.

typically record the feature frequency in the text. Typically, a feature vector is built such that each attribute records the frequency or the presence (absence) of some feature in the message. In the case of SMS, to obtain distinctive features based on words frequency is difficult because of the little amount of words. But a more important problem is the fact that the transformation of the message, which is an ordered sequence of words, into a feature vector, which is an unordered set of assumed unrelated features, leads to a loss of valuable information, any time that a human readable message is defined by a sequence of words, not by a set of them. A change in the words order may change the message. Therefore the sequentiality information is an important feature we need to extract.

By simple observation, it is easy to take into account that spammers follows a particular style. From the spammer point of view, it is difficult masquerade its style without expressing their real intention, which is, to offer you something (by cheating or not), in order that you do something back. Our hypotheses, is that writing style is composed by a set of specific word sequences. For example, it is common to find in spams SMSs variations of the sentence: you have won (something), call (number) to claim. In our opinion, extracting the message style is fundamental since we consider it as the most capable feature for discerning between spam and non-spam message. In this process, not only the words are important but also the order they appear. We then propose a way of capturing the features and its order of apparition in text as a writing style meta-feature. The features used in our method are extracted by three techniques, Extrinsic Information, Term Clustering [10] and Sequential Labeling Extraction [11].

The article continues as follows. The section two is dedicated to previous works in sms-spam filtering field. In the section three the feature extraction method is explained in details. The section four is dedicated to experimentations and to analyze the results. Section five concludes the article and section six thanks the all the help and the support given to this work.

2 Related Work

Zelikovitz *et. al* [2] use Extrinsic Information by combining labeled and unlabeled instances in the training phase. The unlabeled instances are introduced depending of the similarity with the labeled ones. We make a different use of extrinsic information, instead of using it to reinforce the training data we use it to improve the extracted features.

In general text categorization several kind of features and feature extraction method exits. From those employed in SMS spam filtering we can count, Bag-of-Words (BoW), Part-of-Speech (PoS), Word bigrams and trigrams, and Orthogonal Sparse Bigrams (OSB). Bag of words [6], creates a bag with the most common words appearing in a class. Part-of-speech trigrams [5], use a predefined set of codes to tag syntactic information. Word bigrams and trigrams features are combination of two or three characters respectively found in text [7]. Special characters sequences, also called texties, are combination of characters to shortly express ideas and feelings. Some examples are: ":)", which means happy "LoL" which means "Laughs out Loud", "XXX" related with sex or porn, and others. Orthogonal Sparse Bigrams [1], define a window and then word pair with a common word are extracted. For example, "I fell lucky today", generates the following OSBs, (lucky today), (fell today), (I today).

Sohn *et. al* [7] consider to capture the style by proposing stylistic features. Stylistic features are combinations of the mentioned above features, plus phrasal categories, like noun phrases, verb phrases, which are extracted using tree parsers. They also perform a very useful set of experimentation to rank the performance of each kind of features. Gormack et al. [3] [4] evaluate the performance of several spams filters over SMS data set. In their experiments the SVM-based filter was the one with better performance.

3 Feature Extraction Method

In Sequential Labeling Extraction method a sequence of labels is outputted from the text by assigning tags to terms according to some criteria [11]. For example, if the sequence "John paints the wall" is labeled according to the words grammatical function, the tags sequence would be "NOUN VERB ART NOUN". Term Clustering on the other hand, is about grouping words with a high degree of pairwise semantic relatedness into clusters, so that clusters (or their centroid) describe dimensions of the feature space [10].

Our method begins by generating two bag of words (BoW), one for Spam class and another for Ham class. Each BoW contains those terms with high probability of appearing in the corresponding class and with low probability of appearing in the other class. To select those words, for every term w and for every message m in the training set, we compute two conditional probabilities, $P(w \in m|Spam = LABEL(m))$ and $P(w \in m|NonSpam = LABEL(m))$. Then we compute two coordinates, $x = P(w \in m|Spam = LABEL(m)) - P(w \in m|NonSpam = LABEL(m))$ and $y = P(w \in m|Spam = LABEL(m)) + P(w \in m|NonSpam = LABEL(m)$. With these coordinates each word is represented by a point in a two-dimensional space as depicted in Figure 1. It is straightforward to see that the more representative words in each class are those far from the center ($x << 0, x >> 0$), and close to the top ($y = 2$). That is to say words with high intraclass probability and low interclass probability. Therefore, as the representative words we take those out of the gauss bell represented in the Figure 1. Finally, the representative words with $x < 0$ conforms the spam BoW, while the words with $x > 0$ conforms the non-spam BoW.

Up to this point the words inside the two bags are only related by its frequency of occurrence, nothing can be said about the semantic relationship between them. Here is when the Extrinsic Information and Term clustering play its role. As external source of information we use the well-known MySpell dictionary [12]. In this, each word has assigned a set of senses. The senses denote different meanings depending on the context. Each word sense in the dictionary is associated with a set of synonyms and antonyms for the word. Once the spam and the non-spam BoW are obtained Term Clustering is applied separately on each bag. To

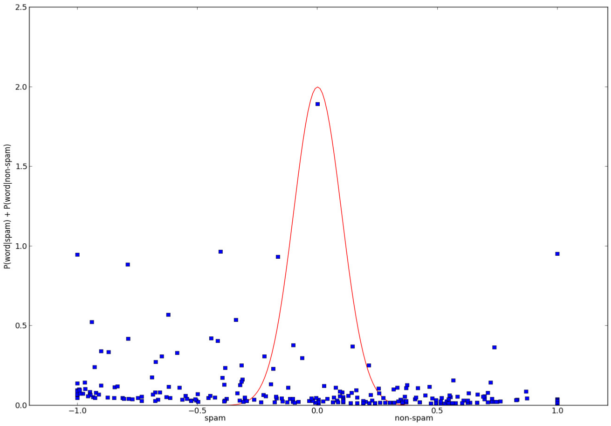

Fig. 1. Graphic of term intraclass vs term interclass probability for spam and non-spam classes

clustering the terms in a bag of words, we use a similarity function that express the closeness of two terms according the synonyms and antonyms they share in all the senses. Then two similarity matrix are computed one for spams terms and the other for non-spam terms. A hierarchical clustering algorithms is executed over these two matrices. The final result is the set of spams and non-spams term clusters. The number of clusters for a data set of 2500 SMS was of 75 and the average number of terms per cluster was four. Each cluster contains words somehow related. For example, the words ”day, night, hour, time” appears in a cluster that obviously group words related with the concept of time.

In traditional term clustering technique, each cluster is related to a feature space dimension. Each dimension records some term-frequency based measure of those terms included in the related cluster. We propose a different approach. In this, each cluster is tagged with a unique label having set of labels from the set of generated terms clusters. The labels are then used as features that can be extracted from the message by applying Sequential Labeling Extraction. Recall, that this technique assigns labels to terms following some criteria. In our case, the criteria is that each word in the message is translated to the cluster label it belongs to. If the word is not member of any cluster, then this is ignored and the next word is analyzed. The final result, is the transformation of the message in a sequence of labels that we assume as our ordered feature vector.

An ordered feature vector impose an ordered feature space. In this space the manning we give to the dimensions is different from traditional approach. According to this, dimensions are ordered such that, dimension n captures the feature appearing in the message just after the feature captured by dimension $n-1$. The dimensions take values form the set of cluster labels and the number of dimensions is pre-established. Since each dimension precedes the other, we call this feature space, Precedence Features Space. One interesting property of this space is that the value in dimension n is conditioned by the value in dimension

$n-1$ as a word in a text is conditioned by the preceding words. The idea behind this representation is to capture the sequentiality of the text as a metafeature which is recorded by the combination of features in the feature vector.

Since we assume the sequences of labels extracted from messages as feature vectors, the vectors may have different lengths. To normalize, we fix the vector length to the greater one obtained from the training set. A natural concern is this length can be too big. Obviously, the size of the vector depends on the size of the text and the number of representative words, i. e. word that appears in clusters. Fortunately, the size of the SMS messages are short by nature, in our experiments the maximum number dimensions we needed was forty, so this is our vector size. Furthermore, when we apply significance algorithm to evaluate the attributes, the number of significative attributes is reduced to less than 35. Another consequence of the translation of label sequences in vectors is the existence of missing values. This is because sequences may be shorter that vectors, so in that case the rest of the vector is occupied with a missing value tag.

Finally, we will like to make emphasis in the sequentiality the vectors as a way of capturing information regarding to the interdependency of words in the text. In other words, the attribute n of the vector records the feature value found in the text after the value of the attribute $n-1$. If a decision tree classifier is used for example, the branches elements will have order, therefore they can be treated as state machines that match not only the words, but also the sequentiality information. Anyhow, in the next section not only the rule based and tree based classifiers are tested, but in fact, the best overall performance is achieved by this kind of classifiers.

4 Experiments and Results

For experimentation we use the public online SMS data set *SMS Spam Collection v1.0* [9] which has also been used in [3][4]. This data collection contains 5288 SMS labeled as spams (spam), and non-spam (ham). In the data set there are 4677 ham instances and 611 spam instances. We use WEKA framework to evaluate several classifiers and classifiers combinations, with the proposed feature extraction method. All the classifiers were tested using 10-fold crossvalidation.

A generally accepted metric to evaluate the spams filters is the Area under the ROC curve (ROCA). The ROC curve measures the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR). The optimal case is a ROCA value equal to the unity, which means that the classifier makes no mistakes in discerning between spam and non-spam.

The table 1. shows the evaluated classifiers ranked by the ROCA. The best ROCA is achieved by the Decision Table Naive Bayes, this is a rule based classifier, and the worst ROCA is for SimpleCart algorithm which is a tree based classifier. Although ROCA is considered a good overall metric, this works well when the misclassification rates of both classes are equally important [8].

In real applications the Ham misclassification rate is more significant that the spam misclassification rate. From the business point of view, an undelivered fair

Table 1. Classifiers Performance

Single Classifiers						
					Num. Classified as	
Classifers	**ROCA**	**Class**	**TPR**	**FPR**	Ham	Spam
DTNB	**0.970**	Ham	0.986	**0.170**	4612	65
		Spam	0.830	**0.014**	104	507
Lazy	0.969	Ham	0.997	0.314	4664	13
KStar		Spam	0.686	0.003	192	419
Naive Bayes	0.968	Ham	0.978	0.180	4576	101
		Spam	0.869	0.022	80	531
NBTree	0.966	Ham	0.978	0.142	4582	95
		Spam	0.858	0.020	87	524
Bayes Net	0.964	Ham	0.979	0.180	4579	98
		Spam	0.820	0.021	110	501
Random	0.958	Ham	0.999	0.558	4674	3
Forest		Spam	0.442	**0.001**	341	270
SVM	0.910	Ham	0.990	0.170	4628	49
		Spam	0.830	0.010	104	507
Random.	0.876	Ham	0.933	0.550	4646	31
Tree		Spam	0.450	0.007	336	275
Simple	0.860	Ham	0.977	0.272	4570	107
Cart		Spam	0.728	0.023	166	445
Combined Classifiers						
Random Forest	0.969	Ham	0.985	**0.131**	4607	70
+ NBTree		Spam	0.969	**0.015**	80	531
Lazy KStar +	0.952	Ham	0.998	0.257	4666	11
DTNB		Spam	0.740	0.002	157	454

message is unacceptable since it directly affects the main source of profit. Therefore to offer a more realistic view of the results, Table 1 also shows some other additional information. Next to the ROCA column is the class name column, which helps to offer per class metrics. It follows the True Positive Rate, and the False Positive Rate columns. If for example, the class with major interest is the Ham class (Ham rows), then TPR represents the portion classified as Ham which are truly Ham, while the FPR represents the portion of instances classified as Ham that are actually Spam. At last, The Ham and Spam rows at each classifier together with the last two columns in the table, conform the Contingence Matrix for the experiment.

As mentioned before, the ROC Area is good overall measure but is not enough to make a decision about the SMS spam filters. For example, in Table 1. DTNB presents the highest ROCA, while its Ham misclassification rate is of 1,4% and its spam misclassification rate is of 17%. However, SVM classifier, presents a lower ROCA but achieves 1% of Ham misclassification and equally let 17% of Spam messages to pass as ham messages. In that sense, the algorithms presented range from more restrictive to more permissive regarding to the spam class. Therefore, a natural strategy is to combine them together. As consequence we also evaluate some classifiers combinations which are shown separately in the bottom part of the table 1.

In general all the tested classifiers presents a ham misclassification rate lower that 2,5%, while six of them presents a spam misclassification lower than 20%. Random Forest classifier presents a ham misclassification rate of 0.01% which is

Table 2. Comparison with other types of features

Features	(1 - ROCA)%
OSB	5.57
Stylistic	3.833
Stylistic and OSB	2.069
Proposed	3.0

extremely good, but with a spam misclassification over the fifty percent, manning that fifty percent of spam messages are misclassified as ham. When Random Forest is combined with NBTree, the spam misclassification rate of the combined classifier felt to 15% while spam misclassification remains acceptable in 1,4%. A similar improvement occurs in the combination of the lazy KStar classifier and DTNB. Thence, we consider the in our tests the classifiers combination presents the better performance.

Another general metric to compare the spam filters performance is one minus Area Under the ROC Curve as a percent, i. e. $(1 - ROCA)\%$. Although in our work we did not test others feature extraction methods. We take as reference the serious work made by Sohn et al in [7]. In this work, several feature methods were tested over the same public data set we use. A comparison with those works is shown in table 2. Our feature extraction method outperforms OSB and the stylistic features separately. The combination of OSB and Stylistic features outperforms our method but, considering the little amount of features we need to obtain similar results we may conclude that our method is still competitive.

5 Conclusions

As main conclusion we can say that the proposed feature representation has proved to be expressive enough to discern between spam and non-spam SMS messages. Although we employ a frequentist approach as starting point of our feature extraction method, our feature vector does not record features frequency. Instead of that, this is converted into a sequence of features.

The idea is to capture the writing style as a metafeature that improve the classification accuracy. The features are built using Term clustering technique over the most representative words in each. A dictionary as a source of external information is used in order to fulfill the lack of information on which elaborate good grouping criteria. This Extrinsic Information source is a common dictionary that helps to relates words according to de semantic, or the context they are used.

There are still room for improvement in the proposed method. The similarity function employed to group the terms can take into account others characteristics like, the grammatical function of the words. Also, more sophisticated sources of external information can be used in order to refine the meanings captured in the clusters, making the features more descriptive. Finally, it is also a good strategy to combine different approaches, so would be interesting to evaluate this features combined with others like the stylistic features for example.

Acknowledgments. This work has been partially funded by CONACyT grant SEP-2010-158135. Specially thanks to Dr. Rene Cumplido Parra, professor at *Instituto Nacional de Astrofísica Optica y Electrónica* (INAOE), Mexico, Puebla.

References

1. Siefkes, C., Assis, F., Chhabra, S., Yerazunis, W.S.: 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, pp. 410–421. Springer-Verlag New York, Inc. (2004)
2. Zelikovitz, S., Hirsh, H.: Improving short text classification using unlabeled background knowledge to assess document similarity. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 1183–1190 (2000)
3. Cormack, G.V., Gmez Hidalgo, J.M., Snz, E.P.: Spam Filtering for Short Messages. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 313–320. ACM (2007)
4. Cormack, G.V., Hidalgo, J.M.G., Snz, E.P.: Feature Engineering for Mobile (SMS) Spam Filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 871–872. ACM (2007)
5. Santini, M.: A shallow approach to syntactic feature extraction for genre classification. In: Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics, pp. 6–7 (2004)
6. Li, Y.H., Jain, A.K.: Classification of text documents. The Computer Journal, Br. Computer Soc 41, 537–546 (1998)
7. Sohn, D.-N., Lee, J.-T., Han, K.-S., Rim, H.-C.: Content-based mobile spam classification using stylistically motivated features. Pattern Recognition Letters 33, 364–369 (2012)
8. Assis, Fidelis. OSBF-Lua-A Text Classification Module for Lua: The Importance of the Training Method. En TREC (2006)
9. SMS Spam Collection v1.0, `http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/`
10. Fabrizio, S.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34(1), 1–47 (2002)
11. Prado, D., Antonio, H., Ferneda, E.: Emerging technologies of text mining: techniques and applications. Information Science Reference (2008)
12. MySpell Dictionary, `http://www.openoffice.org/lingucomponent/dictionary.html`