Classification of SMS for the Identification of Relationships

Team Size: 2
Sarah Lee (sl896)
Johanni B. Thunstrom (jbt72)

Abstract

Recognition of text categories can be used for intelligence analysis. We present a scalable approach to text recognition, given a set of messages. In this paper, we propose an optimal method for the automatic classification of text categories, and the subsequent problem of detecting the type of relationship between the sender and the receiver. We focused on using a pipeline to extract data from text messages. We were able to achieve excellent results classifying text messages for its appropriate relationship types. For the purpose of this study, we recognize two text message classes: friends or intimate. The techniques we have outlined may serve as a building block for identifying texting trends and other characteristics for different texting types across a larger data set.

I. Introduction

SMS or short message service has always been a very popular form of communication, especially for the younger generation. With recent emergence of advanced smartphones, phone communication has seen an even greater increase in demand and usage. Especially with devices that can support images and other media types, many use texting and messaging as a convenient first line of communication. Few companies, however, are using automated techniques to analyze these communication messages to identify customer behaviors and trends. We are interested in creating a system that automatically identifies type of text messages for the purpose of identifying relationships between phone users. The development of such a system would make it feasible for companies to construct a database of currently trending text message usages, or get statistics about customer demographics and relationship statuses. This would significantly improve the quality of current phone companies' consumer analytics, and thus the effectiveness of their marketing strategies and forecasts. For example, if a company notices that its target market is on the whole being very intimate, perhaps they could market certain products tailored for intimate occasions. Moreover, sms recognition can be a very effective tool to enforce productivity in the work place. For example, in very professional work places such as federal agencies and banks, they enforce a policy to limit the use of company phones and other communication devices for work-related uses. By classifying SMS messages, companies can prevent possible scandals and increase employee productivity by monitoring the usage of company resources.

Although this problem attracts increasing research interests, an sms recognition program remains challenging for two primary reasons. First, there are many subtle nuances to human language. Jargons, "slang", and idiomatic expressions are often hard to detect correctly by machine learning algorithms. Second, different types of messages can essentially have similar meaning depending on the formalness and style of a person's texting voice. Therefore, the question of whether it is even possible to objectively discern a certain SMS type purely through visual data arises. Recognizing these issues, SMS recognition, and indeed object recognition in general, is largely an unsolved problem and an active area of research. Many text classifiers have been proposed in the literature using machine learning techniques, probabilistic models, generative learning, and more. They often differ in the approach adopted: decision trees, naıve-Bayes, rule induction, neural networks, nearest neighbors, and lately, support vector machines. Although many approaches have been proposed, automated text categorization is still a major area of research primarily because the effectiveness of current automated text classifiers is not faultless and still needs improvement. Despite this, there are still gaps in the literature surrounding basic classification across different SMS message types. With all the effort in this domain there is still room for improvement and a great deal of attention is paid to developing highly accurate classifiers.

II. APPLICATIONS:

There are many potential applications of text classification as briefly mentioned in the introduction. In this section we examine some of the text classification applications.

A. Information Retrieval:

News or media companies typically see hundreds and thousands of submissions every day. In order to efficiently handle such vast flow of information, there is a need of an automatic text classification system, which would categorize each document by topics so that they could be sent to the relevant recipient. Other companies that provide services such as advertisement data, customer profiles, and other marketing tools can utilize SMS classification to better target their desired demographics.

B. Spam Filtering Applications:

Receiving of vast quantities unsolicited junk email, i.e, spam is a big problem. A text classification system could, in the ideal case, categorize incoming messages into genuine and spam categories, rejecting these that it found to be spam.

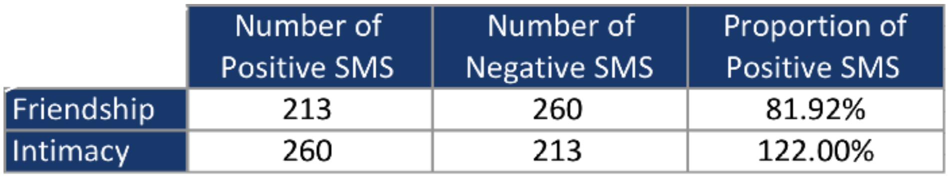
C. Identity Based Access & Reporting:

Filtering can be configured to create access policies based on groups, departments, levels in hierarchy or even the individual user. This allows enterprises to create different policies based on work profile to finance, marketing, HR, department or for educational institutions based on academic requirements for students, staff, administrator. This is useful for companies seeking to increase productivity to restrict only those who have the permission to do certain tasks perform them and prevent others that do not.

C. Filtering Pornography Content:

As the Internet has rapidly been expanded, we can find information quickly and easily. Minors have easy access to the material and monitoring them is not easy. The exponential increase of information in internet has raised the issue of information security. Pornography web content is one of the biggest problems for parents and school administrators who wish to monitor the activities of children and minors. Numerous parental control software has been developed but they are not always guaranteed in performance. Classification approaches have been proposed to avoiding these illicit web contents accessing by the children. Text classification controls search results from google, yahoo and other search engines. When used, web sites containing pornography and explicit sexual content can be blocked from google, yahoo and other search engines. SMS classification can prevent unsuspecting minors from reading and opening harmful messages sent from strangers, spammers, and advertisers from pornography industries.

III. Data Description



	Number of Positive SMS	Number of Negative SMS	Proportion of Positive SMS
Friendship	213	260	81.92%
Intimacy	260	213	122.00%

Table 1: Proportion of positive to negative instances used for training, validation, and test data in each class

A total of 473 messages were used in the process. The data was collected with a random sampling of students in different parts of campus willing to provide sample text messages from their SMS history. To ensure more accurate representation of the population, samples were collected from students at various locations and hours of the day. Whether a subject was asked to provide a friends text or an intimate text was generated randomly with a random number generator. To prevent a sampling bias, text messages were collected at all parts of campus that corresponded to the different schools.

Table 1 gives a summary of the proportion of the negative and positive results used for training, validation, and testing. Each data set was comprised of positive examples of its own class and negative examples taken from the other classes. With 213 friend texts and 260 intimate texts, friends composed 81.92% of the proportion of positive SMS messages and intimate composed 122.00% of positive SMS texts.

IV. Methodology

A. Support Vector Machines:

Support vector machines are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Formally, it is defined as: Given some training data $\mathcal{D}$, a Mercer kernel K, there is a set of hyperplanes that separate the data in the induced feature space F. It can be shown with the following formalization:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\, y_i \in \{-1, 1\}\}_{i=1}^{n}$$

where n represents points and y_i is either 1 or −1, indicating the class to which the point $\mathbf{x}_i$ belongs. Each is a *p*-dimensional real vector. We want to find the maximum-margin hyperplane that divides the points having $y_i = 1$ from those having $y_i = -1$. Any hyperplane can be written as the set of points $\mathbf{x}$ satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

where . denotes the dot product and $\mathbf{w}$ the (not necessarily normalized) normal vector to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector $\mathbf{w}$.

The version space, V is then defined as:

V = {f ∈ H | ∀i ∈ {1 . . . n} yi f (xi) > 0}.

Notice that since H is a set of hyperplanes, there is a bijection between unit vectors w and

hypotheses f in H. Thus we will redefine V as:

V = {w ∈ W | w = 1, yi (w · Φ(xi)) > 0, i = 1 . . . n}.

By definition, points in W correspond to hyperplanes in F. The intuition behind the converse is that observing a training instance xi in the feature space restricts the set of separating hyperplanes to ones that classify xi correctly. SVMs find the hyperplane that maximizes the margin in the feature space F. One way to pose this optimization task is as follows:

maximize w∈F mini {yi (w · Φ(xi))}

subject to: w =1

yi (w · Φ(xi)) > 0 i = 1 . . . n.

B. Naive Bayes Classifier:

A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes’ theorem states the following relationship:

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) P(x_1, \dots x_n \mid y)}{P(x_1, \dots, x_n)}$$

Using the naive independence assumption that

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

for all i, this relationship is simplified to

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Since $P(x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i \mid y)$; the former is then the relative frequency of class y in the training set.

C. Stochastic Gradient Descent

Stochastic gradient descent is a gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions. Both statistical estimation and machine learning consider the problem of minimizing an objective function that has the form of a sum:

$$Q(w) = \sum_{i=1}^{n} Q_i(w),$$

where the parameter w is to be estimated and where typically each summand function $Q_i()$ is associated with the i-th observation in the data set.

When used to minimize the above function, a standard gradient descent method would perform the following iterations :

$$w := w - \alpha \nabla Q(w) = w - \alpha \sum_{i=1}^{n} \nabla Q_i(w),$$

where α is a step size or the learning rate.

II.b) Process & Method

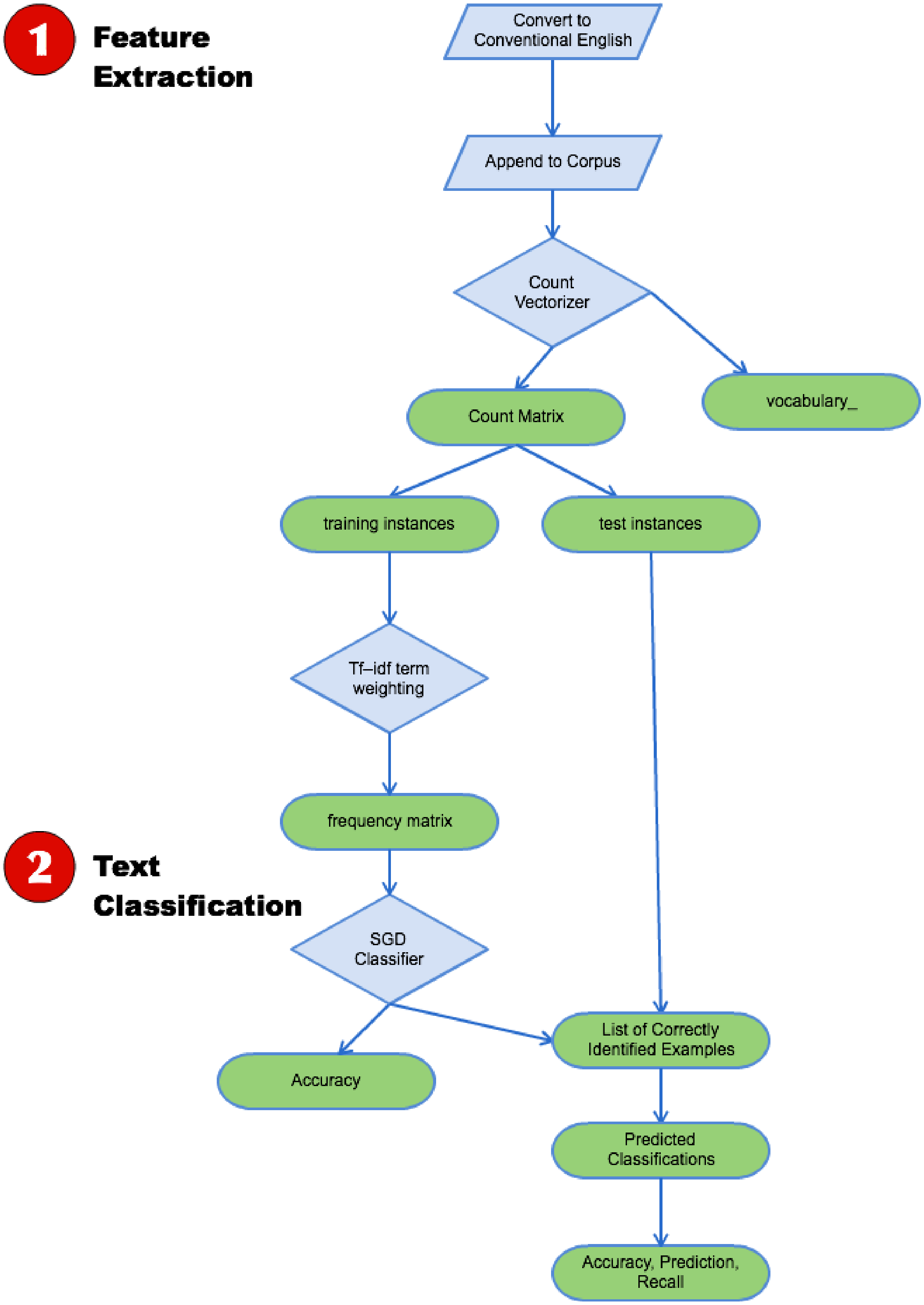
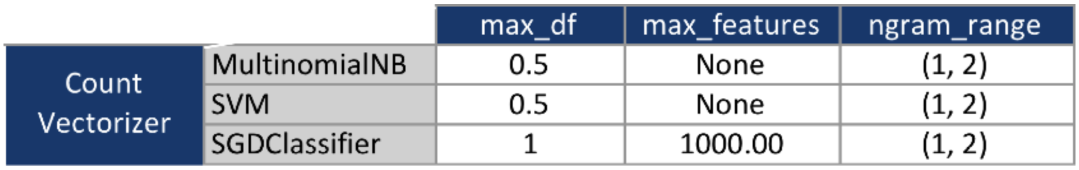
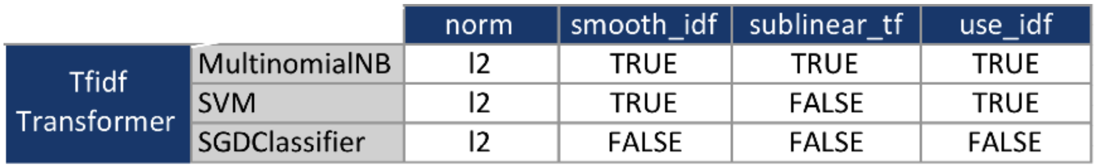


Figure 1: Method Diagram



		max_df	max_features	ngram_range
Count Vectorizer	MultinomialNB	0.5	None	(1, 2)
	SVM	0.5	None	(1, 2)
	SGDClassifier	1	1000.00	(1, 2)

Table 2: Optimal parameters for the Count Vectorizer given the learning method



		norm	smooth_idf	sublinear_tf	use_idf
Tfidf Transformer	MultinomialNB	l2	TRUE	TRUE	TRUE
	SVM	l2	TRUE	FALSE	TRUE
	SGDClassifier	l2	FALSE	FALSE	FALSE

Table 3: Optimal parameters for the Tfidf Transformer given the learning method

Step 1: Text Pre-processing:

We used a pipeline procedure to find the optimal parameters for each step in the process. The first step in our implementation of this system is converting the text messages to standard english. This helps reduce a level of error in frequency due to the same word being misrepresented either through misspellings or slang. The count matrix is normalized using a term frequency inverse document-frequency transformer. In addition, to normalizing the word count list, the transformer removes stop words, common words that are not useful for text classification. Examples of stop words that have been removed are “a”, “I”, “he”, “she”, “is”, “are”, and much more. Absolutely all one letter words are thrown out of the data, due to their significantly low probability of being an influential keyword. The output from the inverse document frequency transformer is used as a feature value for classification. These features are known as Unigram features, where each word is used as a feature. which also eliminates words that have little impact due to appearing in every instance such as the words “the” and “an.” Afterwards the process is split into the test set and training set, which is run through one of the three classifiers: Multinomial Naive Bayes Theorem, Support Vector Machines, and Stochastic Gradient Descent.

Step 2: Training

Text classification is the task of classifying documents by their content, often represented as a bag of words. We trained three classifiers using the below classification tools.

SVM: SVM was chosen because it is commonly used in computer vision tasks for recognizing various different object types. We believe that SVM would be effective given that we were training a limited number of classes, and we were interested in getting high accuracy without regard for response times. We initially started to use SVM classifier with a polynomial kernel, but switched to a linear kernel, which achieved higher results.

Multinomial Bayes Theorem: In contrast to SVM, we tried Multinomial Naive Bayes, because we were interested in what kind of accuracy numbers we would get for an algorithm that's supposed to be very fast. This also serves a simple baseline implementation. We accurately predicted a low accuracy for this due the algorithms naive assumption that word order is irrelevant in classification.

Stochastic Gradient Descent Classifier: Finally, we used Stochastic Gradient Descent algorithm, and achieved the best results. It is used to minimize the objective function which is in the form of a summation of differentiable functions. In our case, we want to minimize the sum of least squares in the prediction of our linear classifiers when fitting a regression line. Additionally, since we are working with sparse matrices, we believe that the use of a stochastic gradient descent model would provide multiple advantages, including efficiency and ease of implementation.

In classifying the training data, we used a pipeline to find the optimal parameters in order to achieve a higher accuracy. For the Multinomial Naive Bayes theorem, we found the best parameters to be an alpha of 0.3 and a fit prior of false. For SVM, the parameters tested was C-value, degree, and kernel. The best accuracy was achieved with a C value of 1.5, degree of 1, and a linear kernel. The best accuracy was achieved with the SGD Classifier, where we tested parameters alpha, penalty, and shuffle. Optimal accuracy was achieved with an alpha of 0, penalty of l1 and shuffling of data set to True.

Step 3: Classification

We applied the classifiers to new instances, the test set, in order to calculate accuracy, precision, and recall. Based off these results, we determined which algorithm was best fit for the classification of SMS messages into friends or intimate categories.

IV. Results and Discussion

In statistics, a paired t-test is a type of location test that is used when comparing two sets of measurements to assess whether their population means differ and offer the advantage of a reduced variance compared to an unpaired test. A paired difference test uses additional information about the sample that is not present in an ordinary unpaired testing situation, either to increase the statistical power, or to reduce the effects of confounders. For our data, we use a paired student-t test as the population deviation of difference is not known for our data.

In our paired difference analysis, we would first subtract the pre-treatment value from the post-treatment value for each subject, then compare these differences to zero:

If we only consider the means, the paired and unpaired approaches give the same result. To see this, let Y_{i1}, Y_{i2} be the observed data for the i^{th} pair, and let $D_i = Y_{i2} - Y_{i1}$. Also let D, Y_1, and Y_2 denote, respectively, the sample means of the D_i, the Y_{i1}, and the Y_{i2}. By rearranging terms we can see that

$$\bar{D} = \frac{1}{n}\sum_i (Y_{i2} - Y_{i1}) = \frac{1}{n}\sum_i Y_{i2} - \frac{1}{n}\sum_i Y_{i1} = \bar{Y}_2 - \bar{Y}_1,$$

where *n* is the number of pairs. Thus the mean difference between the groups does not depend on whether we organize the data as pairs.

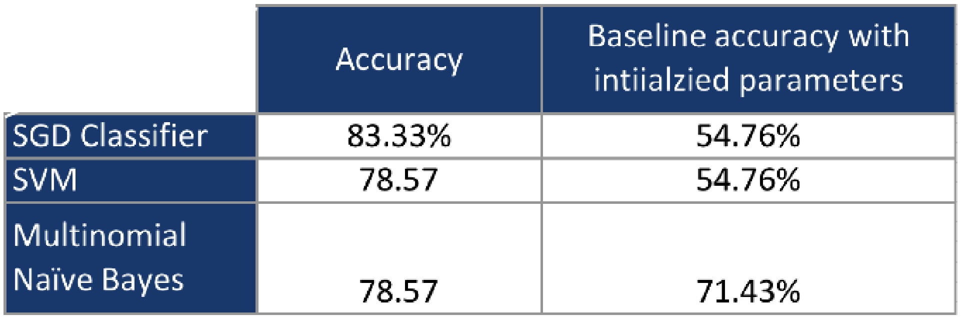
Although the mean difference is the same for the paired and unpaired statistics, their statistical significance levels can be very different, because it is easy to overstate the variance of the unpaired statistic. The variance of D is

$$\begin{aligned}\text{var}(\bar{D}) &= \text{var}(\bar{Y}_2 - \bar{Y}_1) \\ &= \text{var}(\bar{Y}_2) + \text{var}(\bar{Y}_1) - 2\text{cov}(\bar{Y}_1, \bar{Y}_2) \\ &= \sigma_1^2/n + \sigma_2^2/n - 2\sigma_1\sigma_2\text{corr}(Y_{i1}, Y_{i2})/n,\end{aligned}$$

where σ_1 and σ_2 are the population standard deviations of the Y_{i1} and Y_{i2} data, respectively. Thus the variance of D is lower if there is positive correlation within each pair.

For the experiment, we do a difference in the accuracy of SMS friends text calculated correctly versus that of intimate. As final calculation, t-test on \our data produces a p-value of p<0.021 and a t-value of -2.13. Thus, if we had a significance level of alpha=0.05, we conclude that our classifier better classifies intimate texts than friend texts.

According to our final results, the Stochastic Gradient Descent Classifier achieved the highest accuracy. This indicates a unique shaping of the feature vector space, similar to the image shown in Figure 2.



	Accuracy	Baseline accuracy with intiialzied parameters
SGD Classifier	83.33%	54.76%
SVM	78.57	54.76%
Multinomial Naïve Bayes	78.57	71.43%

Table 4: above table shows the accuracy level for the classifiers used in the project compared to the accuracy outputted without any changes to the initialized parameters

VI. Conclusion

Although our results are not detailed enough to detect SMS trends at this point, our high accuracy in detecting the basic messages suggests that a similar effort with more specific types of categories might be effective. Stochastic Gradient Descent Classification seems to be effective in distinguishing different types of text messages.

APPENDIX II: Resources

[1] A Study on Analysis of SMS Classification: Using Document Frequency Threshold
http://www.mecs-press.org/ijieeb/ijieeb-v4-n1/IJIEEB-V4-N1-6.pdf

[2] Lexical Normalisation for Social Media Text
http://ww2.cs.mu.oz.au/~tim/pubs/tist2013-lexnorm.pdf

[3] Text Messaging and Personality
http://cardinalscholar.bsu.edu/bitstream/123456789/194432/1/Paul_Korey_J-departmental_honors_thesis-Psychological_Science.pdf

[4] Mobile Phone, SMS, and Relationship
http://epublications.bond.edu.au/cgi/viewcontent.cgi?article=1076&context=hss_pubs

[5] A Large Scale Study on Text Messaging Use
http://www.battestini.net/blog/wp-content/uploads/fp217-battestini.pdf

[6] Creating Live, Public Short Message Service Corpus: The NUS Corpus
http://link.springer.com/article/10.1007%2Fs10579-012-9197-9

[7] Classifying Text Messages for the Haiti Earthquake
http://faculty.ist.psu.edu/wu/papers/emerse-iscram2011.pdf

[8] Text Categorization with Support Vector Machines: Learning with Many Relevant Features
http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf

[9] Experiments with SMS Translation and Stochastic Gradient Descent in Spanish Text Author Profiling
http://www.uni-weimar.de/medien/webis/research/events/pan-13/pan13-papers-final/pan13-author-profiling/caurceldiaz13-notebook.pdf

[10] Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent
http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf