Optimizing Statistical Distance Measures in Multivariate SVM for Sentiment Quantification

Kevin Labille
University of Arkansas
Computer Science and Computer Engineering
Fayetteville, Arkansas, USA
Email: kclabill@uark.edu

Susan Gauch
University of Arkansas
Computer Science and Computer Engineering
Fayetteville, Arkansas, USA
Email: sgauch@uark.edu

***Abstract*—Twitter sentiment classification has been widely investigated in recent years and it is today possible to accurately determine the class label of a single tweet through various approaches. Although it could open new horizons for business or research, Twitter sentiment quantification, which aims to predict the prevalence of the positive class and the negative class within a set of tweets, has drawn much less attention. This paper presents our research on improving lexicon-based Twitter sentiment quantification. We first introduce a new approach to building a paired-score sentiment lexicon that is better suited for sentiment quantification. We then propose a novel feature vector representation for tweets that incorporates a collection of sentiment features. Finally, we investigate and compare several statistical distance kernels in multivariate Support Vector Machine for sentiment quantification. Results suggest that optimizing the Hellinger Distance with a multivariate SVM using our new sentiment lexicon outperforms current sentiment quantification approaches, including neural network approaches.**

***Keywords*—sentiment quantification, sentiment lexicon, multivariate SVM, statistical distances**

I. Introduction

The amount of user feedback available online has increased tremendously and it is now possible to read the opinions of millions of people all over the Internet on movies, restaurants, hotels, books, products, and professionals. This wealth of information allows researchers to study the ways in which individuals express opinions and to mine collections of opinions to identify trends and consensus. Two new research area have arisen from this phenomenon: sentiment analysis and sentiment quantification. Namely, sentiment analysis is the computational analysis of opinions in text; its goal is to identify the semantic orientation, or polarity, of such textual data. In contrast, sentiment quantification aims to estimate the distribution of documents that belong to each polarity class. Sentiment analysis is widely applicable in various areas, for example, politics and retail. For example, Wang et al. [1] applied real-time sentiment analysis to Twitter data to analyze public sentiment toward presidential candidates in the U.S elections of 2012. The most prominent and perhaps the most prevalent utilization of sentiment analysis and sentiment quantification, however, is in business intelligence, since customer's feedback directly reflects their opinion about a product or service. Sentiment analysis and sentiment quantification can be used as a concept testing tool when a new product, campaign, or logo is launched. It can be used to improve a company's own performances by analyzing competitor's sentiment data and gain competitive advantage. It can also be applied to gain insight from the opinions of customers to diagnose possible problems and make improvement. Additionally, sentiment analysis and sentiment quantification can be used to track customer sentiment over time. Although a lot of work has been conducted in sentiment analysis, we believe that the last application aforementioned can be further exploited through sentiment quantification to open new horizons for businesses and for research. In this paper, we focus on sentiment quantification over a set of tweets wherein the goal is to accurately predict the proportion of tweets that are positive and the proportion of tweets that are negative. This paper offers three contributions. In particular, (1) we propose a new statistical method for building sentiment lexicons from tweets that maps each word to a pair of **positive** and **negative** sentiment scores rather than the usual single sentiment score. (2) We investigate using this sentiment lexicon to derive sentiment features that capture tweets' positive aspects and negative aspects. These feature vectors include a combination of word sentiment features and additional features that summarize the positive and negative word distributions within the dataset. (3) Finally, through a multivariate Support Vector Machine (SVM) we optimize and compare numerous statistical distance kernels to evaluate which one performs best in a sentiment quantification task. Our results show that sentiment features derived from the pair of sentiment scores improve the performances of the quantifier. Finally, we show that a multivariate SVM that optimizes the Hellinger Distance outperform several other statistical distance measures such as the widely used Kullback-Leibler divergence (KLD), and therefore is a better approach for sentiment quantification. Our results outperform recent approaches to sentiment quantification, including neural network-based approaches. The paper is organized as follows, Section 2 will describe research works that are closely related to ours, Section 3 will detail the methodology of our approach to extract sentiment from text and to perform sentiment quantification, Section 4 will present our experimental evaluation, followed by the results in Section 5. Section 6 will end with a conclusion.

 ISBN: 978-1-61208-874-7

II. Related Work

With over 500 million tweets shared every day (6,000 tweets per second), Twitter has become the fastest growing source of information. Users generally tweet about their feelings or opinions about what's happening around the world. This makes Twitter a valuable source of data for sentiment analysis. However, tweets differ from regular text in many ways: the length of a tweets is restricted to 280 characters and, because they are often posted from cellphones, the language used contains many spelling mistakes, abbreviations, and slang words. These characteristics make traditional Natural Language Processing techniques, language models, and traditional sentiment analysis tasks trickier to apply. Despite these challenges, numerous projects have investigated Twitter sentiment classification ([2]–[6]. One of the pioneer works is that of Go et al. [7] wherein they compared a SVM classifier (with feature vectors composed of unigrams, bigrams, or unigrams+bigrams), a Maximum Entropy (MaxEnt) classifier, and a Naive Bayes classifier. Their results suggest that Part Of Speech (POS) tags are not useful in Twitter sentiment classification. They achieved their best accuracy with the MaxEnt classifier and the lowest accuracy with the Naive Bayes classifier. Mohammad et al. [8] and [9] tackled the same problem using a SVM classifier that uses sentiment lexicons as part of the feature vector. They showed that lexicons-related features were valuable features that improved the accuracy of the SVM classifier by more than 8.5%. More recently, a new research focus has emerged from automated classification: quantification. In contrast to classification that aims to estimate the class label of individual instances, the purpose of quantification is to evaluate the population or prevalence of the different classes within the dataset. Although the tasks are related, a method with a high accuracy on the individual level can be biased and achieve poor performance when estimating the proportion of the different classes, requiring new approaches. Esuli and Sebastiani [10] focused on text quantification using multivariate SVM, i.e., SVM_{perf}, that uses the Kullback-Leibler divergence as a loss function (KLD is a measure of the divergence between two probability distributions). They found that SVM(KLD) outperforms all other linear SVM approaches and other quantification methods and is therefore a more appropriate choice for text quantification. Gao and Sebastiani [11] apply the approach from [10] to Twitter data. They concluded that $SVM(KLD)$ outperforms the traditional $SVM(HL)$. In 2016 and 2017, the high-impact conference "International Workshop on Semantic Evaluation", i.e., SemEval, held a track on sentiment quantification. The best approach from SemEval 2016 was by Stojanovski et al. [12]. They used a combination of a Convolutional Neural Network (CNN) and a Gated Neural Network (GNN), which was then fed into a softmax layer. They concluded that the combination of the two neural network is well suited for quantification. In the 2017 edition Mathieu Cliche [13] achieved first place on the sentiment quantification task. He used a deep-learning method that uses both a Convolutional Neural Network (CNN) and a LSTM (Long Short-Term Memory) neural networks that uses word embedding. Our work is similar to [10] and [9] but differs from theirs in several key ways. In particular, we propose a new statistical method for building sentiment lexicons on tweets that maps each word to a pair of **positive** and **negative** sentiment score rather than the usual single sentiment score. Furthermore, although [9] also represents a tweet as a feature vector that uses a sentiment lexicon, we investigate using our newly built sentiment lexicon to derive sentiment features that reflect the words' positive distribution and negative distribution of the tweet. In addition, although [10] optimize the Kullback-Leibler Divergence (KLD) with the multivariate SVM, their choice of that particular statistical distance is not clearly motivated. We believe that mathematically stronger statistical distances could be a better choice in place of KLD. We therefore extend their approach and compare several other statistical distances measures and evaluate how they perform in the sentiment quantification task.

III. Quantifying Tweets

Our approach to quantify tweets is the following. We first represent a tweet in the Vector Space Model (VSM) through a feature vector that captures the sentiment of the tweet. The feature vectors use the Bag-Of-Words (BOW) representation augmented with sentiment features that we compute from a sentiment lexicon. The sentiment lexicon employs a new format and is built through a new statistical approach that we describe in this document. We then use a multivariate Support Vector Machine (SVM) to classify each tweet and count the number of positively classified instances as well as the number of negatively classified instances.

A. Paired-score Sentiment Lexicon

From a collection of tweets, we first build a sentiment lexicon, that is, a list of words with associated sentiment scores. The sentiment scores are calculated using a probabilistic approach. We define the positivity of a word w as $pos(w)$, and its negativity as $neg(w)$. While a single sentiment score gives us information about the polarity strength (its score) and the polarity orientation (its sign) of a word, it does not capture the word's distribution across positive and negative occurrences. Indeed, let's assume that we define the score of a word to be the difference between its positivity and its negativity. Then, if two words have the same sentiment scores does not necessarily mean that they have the same positivity and negativity. For instance, if two words have a score of -0.6, they could be the results of $0.11 - 0.71$ or $0.3 - 0.9$. In other words, we are losing information about the word's distribution across the dataset. We believe that using positivity and negativity values of words separately could improve the effectiveness of the feature vector in catching the sentiment of the tweet, since it embeds more information on the distribution of the words across the different classes. While single-score lexicons perform well for sentiment analysis, it is our intuition that paired-score lexicons are more suitable for sentiment quantification. In addition, such lexicons allows us to compute

 ISBN: 978-1-61208-874-7

distributional statistics such as the average negativity score or the average positivity score of a tweet, which could be useful information in sentiment quantification tasks. We therefore propose a new type of sentiment lexicon that maps each word to a pair of scores $< pos, neg >$, i.e., its positivity and its negativity.

The positivity of a word is calculated by dividing the positive document frequency of the word with the aggregated positive document frequency of every word. To account for potential unbalanced data, it is then normalized by the overall frequency of the word. The same calculation is done on the negative aspect as well. The positivity and negativity scores of a word are therefore calculated as follows:

$$\begin{aligned} Pos(w) &= \frac{pdf(w)}{N_{pos}} \times \frac{1}{df(w)} \\ Neg(w) &= \frac{ndf(w)}{N_{neg}} \times \frac{1}{df(w)} \end{aligned} \tag{1}$$

and:

$$pdf(w) = \sum_{t \in T_{pos}} x \begin{cases} x = \frac{1}{|tweet|} & \text{if } w \in t \\ x = 0 & \text{otherwise} \end{cases}$$

$$ndf(w) = \sum_{t \in T_{neg}} x \begin{cases} x = \frac{1}{|tweet|} & \text{if } w \in t \\ x = 0 & \text{otherwise} \end{cases}$$

$$df(w) = pdf(w) + ndf(w)$$

$$N_{pos} = \sum_{w \in vocab} pdf(w)$$
$$N_{neg} = \sum_{w \in vocab} ndf(w)$$

We first define three terms: $pdf(w)$, $ndf(w)$, and $df(w)$ where $pdf(w)$ is the positive document frequency of w, i.e., the number of time w occurs in positive tweets from the tweet collection T; $ndf(w)$ is the negative document frequency of w, i.e., the number of time w occurs in negative tweets from the tweets collection T; and $df(w)$ is the total number of occurrences of w in the tweet collection T. The positive document frequency . Furthermore, N_{pos} is the proportion of positive words in the collection of tweets, i.e., it is the sum of the the positive document frequency pdf of every word in the dictionary; Likewise, N_{neg} is the proportion of negative words in the collection of tweets, i.e., the sum of the negative document frequency $ndf(w)$ of every word in the dictionary. Pre-processing is performed similarly on all datasets, that is, URL, emojis, Tweet mentions, Tweet hashtags, and smileys are removed. Punctuations and number are further removed and the remaining is lower-cased. After pre-processing the tweets from our training dataset, each unique word is extracted from the remaining text in order to build a dictionary. Using the above formula, we compute each word's positive score and negative score as real values in the range [0, 1].

B. Building a Sentiment Feature Vector

A common way of representing documents in the Vector Space Model is using the Bag-Of-Words (BOW). In this approach, each document is represented by a vector wherein each feature is a word from the dictionary. Therefore, the size of the vector is equal to the size of the vocabulary. For the BOW features, we will use the tf-idf (term-frequency inverse document frequency) value of the word within the tweet. We do not take into consideration the Part-Of-Speech (POS) of the words based on results from Go et al. [7] that demonstrated that POS is not helpful in Twitter data. We further derive additional numerical features that catch several sentiment aspects of the tweet using each word's sentiment scores extracted from the paired-score lexicons. Our intuition is that adding sentiment features to the basic tf-idf BOW could improve the performance by providing crucial sentiment information. The sentiment features we consider are described below.

- *token found*: the number of words in the tweet that were found in the lexicon
- *token total*: the number of words in the tweet
- *max pos*: the maximum positive score in the tweet
- *min pos*: the minimum positive score in the tweet
- *max neg*: the maximum negative score in the tweet
- *min neg*: the minimum negative score in the tweet
- *ratio*: the ratio of *avg pos* over *avg neg*

C. Sentiment Quantifier

A traditional SVM optimizes an univariate loss function, it classifies each item one by one, independently of each other, i.e., an item does not impact how another item is classified. A traditional SVM machine can therefore be used to quantify by classifying each unlabeled documents and by then counting how many documents belong to the positive class and how many documents belong to the negative class.

However, even if the classifier correctly quantifies the positive and negative class proportions in the training set, there is no guarantee that the proportion of positive documents and negative documents will be the same in the test set. In fact, we are expecting a change in the proportion of the positive class and negative class ratios in the test set. Thus, such a quantifier will most likely suffer from statistical bias.

To overcome this problem, we will use a Support Vector Machine (SVM) for multivariate performance measures. The key here is that the multivariate SVM allows the optimization of multivariate performance measures, and particularly all that can be computed from a contingency table. It works by considering hypotheses $\bar{h}$ that maps a set of n feature vectors $\bar{x} = (x_1, x_2, ..., x_n) \in \bar{X}$ where $\bar{X} = X \times ... \times X$ to a set of n labels $\bar{y} \in \bar{Y}$ where $\bar{Y} \subseteq \{-1, +1\}^n$, i.e., $\bar{h} : \bar{X} \longrightarrow \bar{Y}$, as opposed to considering hypotheses h that maps one single feature vector $x \in X$ to one single label $y \in Y$, i.e., $h : X \longrightarrow Y$ [14]. Since statistical distance measures are used to evaluate how similar two probability distributions are, and since they are computable with the contingency table, it makes

perfect sense to optimize them through a multivariate SVM to perform quantification. Thorsten Joachims [14] developed such an SVM machine called SVM^{perf} which was originally developed to optimize the F1-Score, Prec/Rec Breakeven, Prec@k, Prec@k, and ROCArea metrics. We perform quantification using SVM^{perf} similarly to our baseline, that is, by classifying each unlabeled documents and then counting how many documents belong to the positive class and how many documents belong to the negative class. Specifically, we use SVM^{perf} with several statistical distance metrics and compare them to our baseline univariate SVM. Our results should confirm that of [10] wherein they concluded that a multivariate SVM outperforms a univariate SVM in the quantification task. The Kullback-Leibler Divergence (KLD) is a loss function that measures how one probability distribution p diverges from a second predicted distribution q. It is defined as follows:

$$KLD(p,q) = \sum_{c_i \in C} p(c_i) \cdot log \frac{p(c_i)}{q(c_i)}$$

SVM^{perf} was extended to optimize KLD in [10]. We compare it to our own approaches described below. The goal is to compare various measures of statistical distance that have different mathematical properties and compare their performance in a sentiment quantification task. Our first sentiment quantification machine is a SVM^{perf} that optimizes the Hellinger Distance instead of KLD. The Hellinger Distance is a statistical distance used to measure the similarity between two probability distributions. HD is defined as follows:

$$HD(p,q) = \frac{1}{\sqrt{(2)}} \cdot \| \sqrt{(p)} - \sqrt{q} \|_2 \tag{5}$$

The second SVM^{perf} optimizes the Bhattacharyya distance. The Bhattacharyya distance is another statistical distance that measures how similar two probability distributions are. It is defined as follows:

$$D_B(p,q) = -\ln(BC(p,q)) \tag{6}$$

where:

$$BC(p,q) = \sum_{x \in X} \sqrt{p(x)q(x)}$$

is the Bhattacharyya coefficient.

The third statistical distance that we optimize through SVM^{perf} is the Jensen Shannon Divergence. The Jensen Shannon Divergence is a smoothed and symmetrized version of the Kullback-Leibler Divergence. It is defined as follows:

$$JSD(p,q) = \frac{1}{2} KLD(p,m) + \frac{1}{2} KLD(q,m) \tag{7}$$

where:

$$m = \frac{1}{2}(p+q)$$

The next statistical distance that we use in the multivariate SVM machine is the Total Variation Distance. It is yet another metric used to measure the distance between two probability distributions and is defined as follows:

$$TVD(p,q) = \frac{1}{2} \sum_{x \in X} \|p(x) - q(x)\|$$

The last statistical distance that we use is another symmetrized version of KLD called Resistor-Average Distance introduced by Johnson and Sinanović [15]. It is equal to the harmonic mean of both Kullback-Leibler distances $KLD(p,q)$ and $KLD(q,p)$ and is formally defined as follows:

$$RAD(p,q) = \left[\frac{1}{KLD(p,q)} + \frac{1}{KLD(q,p)} \right]^{-1}$$

We call these SVM $SVM^{perf}(KLD)$, $SVM^{perf}(HD)$, $SVM^{perf}(BD)$, $SVM^{perf}(JSD)$, $SVM^{perf}(TVD)$, and $SVM^{perf}(RAD)$ respectively.

D. Notes on Statistical Distances

From a pure mathematical perspective, a function d in a space χ is said to be a **distance** if for any $x, y, z \in \chi$ the following three axioms are satisfied:

(i) $d(x,y) > 0$ when $x \neq y$ and $d(x,y) = 0$ if and only if $x = y$
(ii) $d(x,y) = d(y,x)$
(iii) $d(x,y) + d(x,z) \geq d(y,z)$

The axiom (i) implies that the distance must be non-negative and respect the identity of indiscernible, axiom (ii) implies that the distance must be symmetric, i.e., $d(x,y) = d(y,x)$, and axiom (iii) implies that the distance satisfies the triangular inequality, i.e., $d(x,y)+d(y,z) \geq d(x,z)$ (for any $x, y, z \in \chi$, the distance $d(x,z)$ is the shortest distance from x to z in the space) [16]. We now discuss a few properties of all 6 statistical measures mentioned in the previous section.

1) Kullback-Leibler Divergence:

The Kullback-Leibler Divergence does not satisfy axiom (ii) and (iii), KLD is therefore not a distance but a pseudo-distance or directed divergence measure. It is considered a measure of divergence because of its ratio $\frac{p(x)}{q(x)}$, that is, the difference in the probability distributions is large when the ratio is far from 1. $KLD(p,q)$ is undefined if there exists a $q(x) = 0$ for which $p(x) \neq 0$. $KLD(p,q)$ has no upper bound, that is, KLD's limit goes to $+\infty$ when $q(x)$ is infinitely small.

2) Hellinger Distance:

The Hellinger Distance satisfies all three axioms,and is therefore a true distance. $HD(p,q)$ is bounded, it has a lower bound of 0 and an upper bound of 1 (due to the $1/\sqrt{2}$ term in the formula). $HD(p,q)$ is also well defined.

3) Bhattacharrya Distance:

The Bhattacharyya Distance does not satisfy axiom (iii), BD is therefore not a distance in the proper sense but a non-directional divergence measure. As per its definition that employs the natural logarithm, BD is undefined is there exists a $q(x) = 0$ for which $p(x) = 0$. BD has an upper bound.

 ISBN: 978-1-61208-874-7

4) Jensen-Shannon Divergence:
The Jensen-Shannon Divergence does not satisfy axiom (iii), JSD is therefore not a distance in the proper sense but a non-directional divergence measure. JSD is a smoothed and symmetrized version of the Kullback-Leibler Divergence. $JSD(p,q)$ is bounded, it has a lower bound of 0 and an upper bound of $\ln(2)$. Because it uses the KLD, $JSD(p,q)$ is undefined if there exists a $p(x) = 0$ or $q(x) = 0$.

5) Total Variation Distance:
The Total Variation Distance satisfies all three axioms and is therefore a true distance. $TVD(p,q)$ is bounded, it has a lower bound of 0 and an upper bound of 1. Furthermore, $TVD(p,q)$ is well defined.

6) Resistor-Average Distance:
The Resistor-Average Distance does not satisfy axiom (iii), and is therefore not a distance in the proper sense but a non-directional divergence measure. RAD is another symmetrized version of KLD. It is equal to the harmonic mean of both $KLD(p,q)$ and $KLD(q,p)$. $RAD(p,q)$ is not defined when either $KLD(p,q)$ or $KLD(q,p)$ is equal to 0.
Although there exists numerous statistical distance measures and divergence measures available, for no apparent reason, the Kullback-Leibler Divergence has become the *de facto* standard measure for evaluating the distance between two statistical distributions. Our intuition is that statistical distances that are mathematically stronger might be a better choice in place of KLD. Our work aims at comparing several statistical distance measures to see which is best for sentiment quantification.

IV. Experimental Evaluation

A. Datasets

We evaluate our approach over several widely used datasets. The datasets include collections of tweets annotated with a class label chosen from *positive, negative, neutral*. Because we are dealing with 2-class sentiment quantification, we ignore tweets that are labeled *neutral* for both training and testing. The datasets are publicly available on the Internet and the tweets contained in each of them were annotated manually to ensure accurate class labels:

- International Workshop on Semantic Evaluation Task 2 A: Sentiment Analysis in Twitter 2013 [17]
- International Workshop on Semantic Evaluation Task 9 A: Sentiment Analysis in Twitter 2014 [18]
- International Workshop on Semantic Evaluation Task 10 A: Sentiment Analysis in Twitter 2015 [19]
- International Workshop on Semantic Evaluation Task 4 D: Tweet quantification 2016 [20]
- International Workshop on Semantic Evaluation Task 4 D: Tweet quantification 2017 [21]
- Sentiment Strength Twitter (SST) dataset created by [22] and modified by [23] to have the *positive, negative*, or *neutral* classes
- Sanders

The SemEval2016 datasets is split into 4 subsets: train, dev, devtest, and test. We combine the train, dev, and devtest subsets for training and use the remaining test subset for testing. In addition, both the SemEval2016 and SemEval2017 datasest are composed of tweets that belong to a particular topic (the Twitter query). The topic is ignored during training, i.e., all tweets are combined and used for training. However, during testing, we use each topic from the test set as a separate test subset. Table I details the size and contents of each of these two datasets.

TABLE I
TWEET QUANTIFICATION DATASETS

Dataset	Topics	Pos	Neg	**Total**
SemEval2016-train	60	2,841	582	3,423
SemEval2016-dev	20	778	279	1,057
SemEval2016-devtest	20	893	216	1,109
SemEval2016-test	100	8,212	2,339	10,551
SemEval2017-train	100	8,212	2,339	10,551
SemEval2017-test	125	2,463	3,722	6,185

TABLE II
SENTIMENT ANALYSIS DATASETS

	Train(+dev)			Test		
Dataset	# pos	# neg	Total	# pos	# neg	Total
SemEval2013	4,215	1,798	6,013	1,475	559	2,034
SemEval2014	4,215	1,798	6,013	982	202	1,184
SemEval2015	4,215	1,798	6,013	1,038	365	1,403
SST	989	842	1,831	263	195	458
Sanders	418	54	872	101	118	219

Unlike the aforementioned datasets, the sentiment analysis datasets (Table II) are not split into topics. We therefore consider the whole dataset as a single topic. Furthermore, the SemEval-task_A (2013-2015) datasets are partitioned into three subsets (training, dev, test), while the Sanders and the SST datasets are not. We therefore split those into two subsets with 80% used for the training set and 20% reserved for the testing set. The training and dev subsets will be combined and used for training while the test sets will be used for testing.

B. Metrics

Commonly used metrics used to evaluate quantification will be used throughout our experiments: the Kullback-Leibler Divergence (KLD), the Mean Absolute Error (MAE), and the Relative Absolute Error (RAE). The Kullback-Leibler Divergence measures how one probability distribution p diverges from a second predicted distribution $\hat{p}$ and is defined as follows:

$$KLD(\hat{p},p) = \sum_{c_i \in C} p(c_i) \cdot log \frac{p(c_i)}{\hat{p}(c_i)}$$

The Mean Absolute Error and the Relative Absolute Error are the absolute error between the class prevalence of two quantities and are defined as follows:

$$MAE(\hat{p},p) = \frac{1}{|C|} \sum_{c \in C} |\hat{p}(c) - p(c)|$$

$$RAE(\hat{p},p) = \frac{1}{|C|} \sum_{c \in C} \frac{|\hat{p}(c) - p(c)|}{p(c)}$$

TABLE III
RESULTS OF THE QUANTIFICATION USING UNIVARIATE SVM VS MULTIVARIATE SVM

	Metrics	univariate SVM	SVM(perf)	SVM(KLD)	SVM(HD)	SVM(BD)	SVM(JSD)	SVM(TVD)	SVM(RAD)
SST	KLD	0.031	0.011	0.036	0.005	0.044	0.046	**0.000**	0.030
	AE	0.124	0.148	0.266	0.100	0.295	0.301	0.028	0.245
	RAE	0.254	0.149	0.268	0.101	0.296	0.303	0.029	0.246
Sanders	KLD	0.000	0.004	0.010	0.001	0.007	0.028	**0.000**	0.007
	AE	0.005	0.088	0.138	0.037	0.115	0.230	0.005	0.115
	RAE	0.010	0.088	0.138	0.037	0.115	0.231	0.005	0.115
SemEval 2013	KLD	0.003	0.046	0.019	**0.000**	0.018	0.024	0.003	0.019
	AE	0.032	0.275	0.194	0.006	0.191	0.219	0.080	0.194
	RAE	0.081	0.290	0.204	0.006	0.201	0.230	0.084	0.204
SemEval 2014	KLD	0.011	0.018	0.022	**0.001**	0.020	0.026	0.001	0.022
	AE	0.059	0.171	0.204	0.040	0.197	0.222	0.045	0.204
	RAE	0.208	0.191	0.228	0.045	0.221	0.249	0.050	0.228
SemEval 2015	KLD	0.017	0.041	0.032	**0.001**	0.030	0.036	0.002	0.031
	AE	0.085	0.260	0.251	0.047	0.244	0.267	0.058	0.248
	RAE	0.220	0.276	0.266	0.050	0.259	0.283	0.061	0.263
SemEval 2016	KLD	0.090	0.069	**0.010**	0.013	0.014	0.011	0.018	0.014
	AE	0.130	0.242	0.098	0.111	0.108	0.098	0.136	0.106
	RAE	1.378	0.266	0.111	0.125	0.121	0.111	0.156	0.119
SemEval 2017	KLD	0.138	0.254	**0.024**	0.028	0.034	0.025	0.031	0.033
	AE	0.188	0.577	0.171	0.182	0.207	0.176	0.192	0.203
	RAE	2.559	0.676	0.200	0.213	0.241	0.205	0.225	0.237
Average	KLD	0.041	0.063	0.022	**0.007**	0.024	0.028	0.008	0.022
	AE	0.089	0.252	0.189	**0.075**	0.194	0.216	0.078	0.188
	RAE	0.673	0.276	0.202	**0.082**	0.208	0.230	0.087	0.202

We calculate the macro average KLD, MAE, and RAE, which is the harmonic mean of each metric. For instance, the macro average KLD will be defined as follows:

$$\text{Macro Average KLD} = \frac{\sum_{i=1}^{n} KLD_i}{n}$$

where n is the number of instances, i.e., the number of datasets in our case.
KLD is not defined in some special cases, namely when the predicted prevalence $\hat{p}$ is zero. To circumvent this problem we smooth both prevalence p and $\hat{p}$ through additive smoothing similarly to [20], [21], that is, $p(c_i)$ becomes:

$$p^s(c_i) = \frac{p(c_i) + \epsilon}{1 + \epsilon * 2}$$

where ϵ is the smoothing factor and is defined as follows:

$$\epsilon = \frac{1}{2 * |dataset|}$$

$\hat{p}$ is be smoothed similarly. We use smoothed KLD throughout the rest of the paper. The metrics are computed for each run and then averaged to yield the final score. Similar to [20], [21], we report three metrics to evaluate the quantification machines but we mainly focus on the smoothed KLD.

C. Experimental Protocol

Quantification can be performed through univariate SVM by classifying each unlabeled documents and by then counting how many documents belong to the positive class and how many documents belong to the negative class. We use this approach as our baseline using a univariate SVM [24] with a linear kernel [8] since it is known to be effective on text classification.

D. Single score vs paired score lexicons

To support our intuition that paired score sentiment lexicon are better suited for sentiment quantification than single score lexicons, we compare their performances using the aforementioned baseline on the datasets described in Table I and report our results in Table IV. The (single) score of a word is defined as the difference between its positivity $Pos(w)$ and negativity $Neg(w)$ as calculated in Section III-A. We derive sentiment features that are similar so the features derived from the paired score lexicon :

- *token found*: the number of words in the tweet that were found in the lexicon
- *token total*: the number of words in the tweet
- *max*: the maximum score in the tweet
- *min*: the minimum score in the tweet
- *avg*: the average of the scores in the tweet
- *nb pos*: the number of positive words in the tweet
- *nb neg*: the number of negative words in the tweet

TABLE IV
SINGLE SCORE VS PAIRED SCORE LEXICONS ON SENTIMENT QUANTIFICATION USING UNIVARIATE SVM

	Metrics	single score lexicon	paired score lexicon
SemEval 2016	KLD	0.094	**0.090**
	AE	0.132	**0.130**
	RAE	**1.269**	1.378
SemEval 2017	KLD	0.174	**0.138**
	AE	0.216	**0.188**
	RAE	2.972	**2.559**
Average	KLD	0.134	**0.114**
	AE	0.174	**0.159**
	RAE	2.121	**1.969**

 ISBN: 978-1-61208-874-7

TABLE V
KLD OF OUR SVM VS OTHER APPROACHES

	SST	Sanders	SemEval 2013	SemEval 2014	SemEval 2015	SemEval 2016	SemEval 2017
SVM(HD)	**0.005**	**0.001**	**0.000**	**0.001**	**0.000**	0.013	0.028
SVM(KLD)	0.036	0.010	0.019	0.022	0.032	**0.010**	**0.024**
SVM(KLD) [11]	0.011	**0.001**	0.029	0.033	0.076	-	-
Stojanovski et al	-	-	-	-	-	0.034	-
Mathieu Cliche	-	-	-	-	-	-	0.036

E. Sentiment quantification

We train the various multivariate SVM using the training subsets and evaluate on the test subsets. Additionally, when using both SemEval2016 and SemEval2017- datasets we will train each quantifier on each individual topic (that is a total of 100 topics when combining train, dev, devtest) and evaluate each quantifier on each of the 100 topics for SemEval2016-test, and each of the 215 topics for SemEval2017-test. We compare the baseline univariate SVM to the various multivariate SVM approaches described in Section III-C, and report our findings in Table III.

V. RESULTS

A. Single score vs paired score lexicons

Table IV shows that the paired score lexicons outperform the single score lexicon on both datasets. Although the performances of the single score lexicons are close to that of the paired score lexicons on the SemEval 2016 dataset, the paired score lexicons achieve a much better KLD on the SemEval 2017 dataset, yielding an average KLD difference of 0.020. It demonstrates using both the negativity and the positivity of the words help to derive sentiment features that help to more accurately catch the distribution of the positive class and negative class in the dataset.

B. Sentiment quantification

Table III shows that all but one multivariate SVM outperforms the univariate SVM. Precisely, the multivariate SVM(perf) (which is the original multivariate SVM) did not perform better than our baseline univariate SVM. However, this is not surprising since SVM(perf) optimizes the error rate which is not a statistical distance and therefore may not be suitable to perform sentiment quantification. We further compare our best approach, e.g. SVM(HD), to other published results. Specifically, we compare our SVM(HD) with the results published by go et al. [11] whom are the originators of SVM(KLD). In addition, we compare our feature vectors to theirs in the SVM(KLD) setting. We also compare with the work of Stojanovski et al. [12] and Mathieu Cliche [13]. The approach from Stojanovski et al. ranked first on the SemEval2016 competition, while the approach of Mathieu Cliche ranked first on the SemEval2017 competition. We report our results in Table V, due to length constraints, we only report the KLD.

Our SVM(HD) outperform the SVM(KLD) from Go et al. [11] on all datasets but the Sanders dataset on which both approaches perform equivalently. In addition, the feature vector that we use with SVM(KLD) outperforms the feature vector used by Go et al. Likewise, our approach outperform both best approaches from SemEval 2016 and SemEval 2017 competitions. Our results suggest that (1) our feature vector that uses sentiment features derived from our paired-score lexicon is a strong model to represent Tweet sentiment in the VSM, and (2) the multivariate SVM machine that minimizes the Hellinger Distance is a very strong approach to the sentiment quantification problem.

Our experimental results show that the best performance is achieved with our SVM(HD), which outperforms all other multivariate SVM. We believe that the mathematical properties of the distances could explain why one would outperform another. Unlike KLD, BD, JSD, and RAD, both the HD and the TVD are true distance metrics, which means that they perfectly capture the notion of distance within a space and satisfy all three axioms. While property (i) is satisfied by all six metrics that we compared, property (ii) and (iii) are not always satisfied. JSD and RAD are both symmetrized version of KLD and yet provide no improvement upon KLD. Hence, we can not positively assert that the symmetrical property (axiom ii) plays a key role in our task. If the triangular inequality (property iii) is not met, then the distance measured by a function d between two points is not guaranteed to be the shortest in that space. This property is one key difference between a true distance metric. KLD, BD, JSD, and RAD do not satisfy axiom iii and are therefore pseudo-distance measure (or divergence measure), whilst both the HD and the TVD do and are therefore true distance metrics, meaning that they perfectly capture the notion of distance within a space. When optimizing a divergence measure through SVM we are minimizing a distance that is not guaranteed to be the minimal distance between both points. Our results indicate that HD and TVD both yield the best results. Therefore, we believe that a true distance metric is more effective for the sentiment quantification task.

VI. CONCLUSION

In this paper, we have presented a statistical approach to build a sentiment lexicon that computes and maps each word to a pair of score, i.e., its positive weight and its negative weight rather than a single sentiment score. Such sentiment values can then be used in a feature vector to represent tweets in the Vector Space Model. We further confirm previous results that showed that a Support Vector Machine for multivariate perfor-

 ISBN: 978-1-61208-874-7

mance measures performs better than a traditional univariate SVM when dealing with the sentiment quantification problem. Our experiments compare and optimize several statistical distance measures through a multivariate SVM machine and our results suggest that the choice of the statistical distance to employ when performing sentiment quantification is crucial and heavily impacts the accuracy. Our experiments show that the Hellinger Distance outperforms all other statistical distances that we explored. Finally, we argue that, since it is a true statistical distance measures, the Hellinger Distance is an ideal candidate for performing sentiment quantification with a multivariate SVM.

REFERENCES

[1] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle," in *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012, pp. 115–120.

[2] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining." in *LREc*, vol. 10, no. 2010, 2010, pp. 1320–1326.

[3] D. Davidov, O. Tsur, and A. Rappoport, "Enhanced sentiment learning using twitter hashtags and smileys," in *Proceedings of the 23rd international conference on computational linguistics: posters*. Association for Computational Linguistics, 2010, pp. 241–249.

[4] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd international conference on computational linguistics: posters*. Association for Computational Linguistics, 2010, pp. 36–44.

[5] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" in *Fifth International AAAI conference on weblogs and social media*, 2011.

[6] J. Zhao, L. Dong, J. Wu, and K. Xu, "Moodlens: an emoticon-based sentiment analysis system for chinese tweets," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1528–1531.

[7] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, p. 2009, 2009.

[8] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," *arXiv preprint arXiv:1308.6242*, 2013.

[9] S. Kiritchenko, X. Zhu, and S. M. Mohammad, "Sentiment analysis of short informal texts," *Journal of Artificial Intelligence Research*, vol. 50, pp. 723–762, 2014.

[10] A. Esuli and F. Sebastiani, "Optimizing text quantifiers for multivariate loss functions," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 4, p. 27, 2015.

[11] W. Gao and F. Sebastiani, "Tweet sentiment: From classification to quantification," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ACM, 2015, pp. 97–104.

[12] D. Stojanovski, G. Strezoski, G. Madjarov, and I. Dimitrovski, "Finki at semeval-2016 task 4: Deep learning architecture for twitter sentiment analysis," in *Proceedings of the 10th International workshop on semantic evaluation (SemEval-2016)*, 2016, pp. 149–154.

[13] M. Cliche, "Bb_twtr at semeval-2017 task 4: twitter sentiment analysis with cnns and lstms," *arXiv preprint arXiv:1704.06125*, 2017.

[14] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 377–384.

[15] D. Johnson and S. Sinanovic, "Symmetrizing the kullback-leibler distance," *IEEE Transactions on Information Theory*, 2001.

[16] A. Ullah, "Entropy, divergence and distance measures with econometric applications," *Journal of Statistical Planning and Inference*, vol. 49, no. 1, pp. 137–162, 1996.

[17] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson, "Semeval-2013 task 2: Sentiment analysis in twitter," 2019.

[18] S. Rosenthal, P. Nakov, A. Ritter, and V. Stoyanov, "Semeval-2014 task 9: Sentiment analysis in twitter," 2019.

[19] S. Rosenthal, S. M. Mohammad, P. Nakov, A. Ritter, S. Kiritchenko, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter," 2019.

[20] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, "Semeval-2016 task 4: Sentiment analysis in twitter," in *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, 2016, pp. 1–18.

[21] S. Rosenthal, N. Farra, and P. Nakov, "Semeval-2017 task 4: Sentiment analysis in twitter," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 502–518.

[22] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010.

[23] H. Saif, M. Fernandez, Y. He, and H. Alani, "Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold," 2013.

[24] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

 ISBN: 978-1-61208-874-7