Tweet Sentiment: From Classification to Quantification

Wei Gao and Fabrizio Sebastiani
Qatar Computing Research Institute
Hamad bin Khalifa University
PO Box 5825, Doha, Qatar
Email: {wgao,fsebastiani}@qf.org.qa

***Abstract*—Sentiment classification has become a ubiquitous enabling technology in the Twittersphere, since classifying tweets according to the sentiment they convey towards a given entity (be it a product, a person, a political party, or a policy) has many applications in political science, social science, market research, and many others. In this paper we contend that most previous studies dealing with tweet sentiment classification (TSC) use a suboptimal approach. The reason is that the final goal of most such studies is not estimating the class label (e.g., Positive, Negative, or Neutral) of individual tweets, but estimating the relative frequency (a.k.a. "prevalence") of the different classes in the dataset. The latter task is called *quantification*, and recent research has convincingly shown that it should be tackled as a task of its own, using learning algorithms and evaluation measures different from those used for classification. In this paper we show, on a multiplicity of TSC datasets, that using a quantification-specific algorithm produces substantially better class frequency estimates than a state-of-the-art classification-oriented algorithm routinely used in TSC. We thus argue that researchers interested in tweet sentiment prevalence should switch to quantification-specific (instead of classification-specific) learning algorithms and evaluation measures.**

1. Introduction

Sentiment classification is the task of detecting, given an opinion-laden textual item (e.g., a product review, a blog post, an editorial, etc.), whether it expresses a positive or a negative opinion about a given entity (e.g., a product, a person, a political party, or a policy). The above scenario is a simple instance of *binary* classification, with Positive and Negative as the classes. Slightly more complex scenarios result when the Neutral class is added to the picture, which makes the task an instance of *single-label multi-class* (SLMC) classification, or when sentiment strength needs to be assessed on an ordered scale consisting of VeryPositive, Positive, OK-ish, Negative, VeryNegative, which makes the task one of *ordinal classification.* In any of the above incarnations, sentiment classification has become a ubiquitous enabling technology in the Twittersphere, since classifying tweets according to the sentiment they convey towards a given entity has many applications in political science, social science, market research, and many others [25]. The tweet sentiment classification (TSC) shared task which has taken place in the context of the last three SemEval evaluation campaigns (where it is called "Sentiment Analysis in Twitter" – see [28], [32], [33]) has been, in all three editions, the SemEval task with the highest number of participants.

In this paper we contend that most previous studies dealing with TSC use a suboptimal approach. The rest of this section is devoted to arguing why this is so.

Usually, the final goal of most such studies is *not* estimating the label of an individual tweet, but studying the distribution of a set of tweets across the classes of interest; in other words, the interest in such studies is not at the individual level, but at the *aggregate* level. For instance, when Borge-Holthoefer and colleagues [6] use Twitter to study the polarization of sentiments during the 2013 Egyptian coup, they are not interested in the sentiments of the specific individual behind a specific Twitter account, but are interested in the aggregate data (possibly broken down according to various criteria) that can be extracted from the entire dataset under study. Similarly, when Dodds and colleagues [12] use Twitter in order to study the spatio-temporal patterns of happiness throughout the US population, they are not interested in how and when a specific person is happy, but are interested in the conclusions that the aggregate data allow them to draw. These examples are not isolated, and it is fair to say that most (if not all) TSC studies conducted, e.g., within political science [6], [21], [24], economics [5], [29], social science [12], and market research [7], [31], use Twitter with an interest in aggregate data and *not* in individual data.

Without loss of generality, we may say that TSC studies that focus on the aggregate level are concerned with estimating the *prevalence* (or "relative frequency") of each class of interest in the unlabelled dataset, i.e., with estimating the distribution of the unlabelled data across the classes of interest. This task is known as *quantification*

Fabrizio Sebastiani is on leave from Consiglio Nazionale delle Ricerche, Italy.

ASONAM '15, August 25-28, 2015, Paris, France
 ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2809327

[2], [4], [13], [15], [26] – a.k.a. *prevalence estimation* [3], or *class prior estimation* [8]. The obvious method for dealing with it is "classify and count", i.e., classifying each unlabelled document via a standard classifier and estimating class prevalence by counting the documents that have been labelled with the class. However, this strategy is suboptimal, since a good classifier is not necessarily a good "quantifier". To see this consider that a binary classifier h_1 for which $FP = 20$ and $FN = 20$ (FP and FN standing for "false positives" and "false negatives", respectively) is worse than a classifier h_2 for which, on the same test set, $FP = 18$ and $FN = 20$. However, h_1 is intuitively a better binary quantifier than h_2; indeed, h_1 is a perfect quantifier, since FP and FN are equal and thus, when it comes to class frequency estimation, compensate each other, so that the distribution of the test items across the class and its complement is estimated perfectly. In other words, a good quantifier needs not only have high (classification) accuracy, it also needs to have small *bias* (i.e., needs to distribute its errors as evenly as possible across the FPs and the FNs). Recent research (e.g., [2], [4], [13], [15]) has convincingly shown that, since classification and quantification pursue different goals, quantification should be tackled as a task of its own, using different evaluation measures and, as a result, different learning algorithms. In this paper we show, on a multiplicity of TSC datasets, that quantification-specific algorithms indeed outperform, at prevalence estimation, state-of-the-art classification-oriented learning algorithms. We thus argue that researchers interested in tweet sentiment prevalence should switch to using quantification-specific (instead of classification-specific) learning algorithms (and evaluation measures).

The paper is organized as follows. In Section 2 we discuss previous work in tweet sentiment classification and previous work in quantification, arguing that these two research streams have never crossed paths. In order to introduce tweet sentiment quantification, in Section 3 we first look at the evaluation measures that are used in the quantification literature. In Section 4 we describe the two tweet sentiment quantification systems we compare in this work, one based on "traditional" classification technology and one based on a quantification-specific learning algorithm. Section 5 describes the results of our experiments, while Section 6 concludes.

2. Related work

Quantification methods. Different quantification methods have been proposed over the years, the two main classes being the *aggregative* and the *non-aggregative* methods. While the former require the classification of each individual item as an intermediate step, the latter do not, and estimate class prevalences holistically. Most methods (e.g., the ones described in [2], [4], [13], [15], [26]) fall in the former class, while the latter has few representatives (e.g., [16], [22]). Within the class of aggregative methods, a further distinction can be made between methods that use general-purpose learning algorithms (e.g., [4], [15]), sometimes tweaking them or post-processing their prevalence estimates to account for their estimated bias, and methods that instead make use of learning algorithms explicitly devised for quantification (e.g., [2], [13], [26]); the one we use in this paper belongs to this latter category.

Applications of quantification. Quantification has been applied to fields as diverse as epidemiology [22], resource allocation [15], word sense disambiguation [8], political science [18], and veterinary [16], [35]. King and Lu [22] apply quantification to the estimation of disease prevalences from "verbal autopsies", i.e., verbal descriptions of the symptoms suffered from deceased persons before dying. Chan and Ng [8] use quantification in order to estimate word sense priors from a text dataset to disambiguate, so as to tune a word sense disambiguator to the estimated sense priors. Hopkins and King [18] estimate the prevalence of support for different political candidates from blog posts. Forman [15] uses quantification for estimating the prevalence of different issues from logs of calls to customer support; these estimates allow a company to allocate more human resources to the issues which have elicited more calls. Finally, in [16] and [35] quantification is used for establishing the prevalence of damaged sperm cells in a given sample for veterinary applications.

To date, we are not aware of any work that has applied quantification-specific algorithms to Twitter data (or to data from other social media, for that matter).

3. Evaluation measures for quantification

Let us look at the measures which are currently being used in the literature for evaluating quantification error.

Our task requires estimating the distribution of a set $\mathcal{S}$ of unlabelled tweets across a set $\mathcal{C}$ of available classes; we will typically deal with the case in which $|\mathcal{C}| = 3$, where the classes are Positive, Negative, and Neutral. Ours is thus a *single-label multi-class* (SLMC) quantification task, and we will thus concentrate on the measures that have been proposed for evaluating it. Note that a measure for SLMC quantification is also a measure for binary quantification, since the latter task is a special case of the former; this would be relevant for datasets in which the Neutral class is absent. Notation-wise, by $\Lambda(\hat{p}, p, S, \mathcal{C})$ we will indicate a *quantification loss*, i.e., a measure Λ of the error made in estimating a distribution p defined on set $\mathcal{S}$ and classes $\mathcal{C}$ by another distribution $\hat{p}$; we will often simply write $\Lambda(\hat{p}, p)$ when $\mathcal{S}$ and $\mathcal{C}$ are clear from the context[1].

1. Consistently with most mathematical literature we use the caret symbol (ˆ) to indicate estimation.

The simplest measure for SLMC quantification is *absolute error* (AE), which corresponds to the average (across the classes in $\mathcal{C}$) absolute difference between the predicted class prevalence and the true class prevalence; i.e.,

$$AE(\hat{p}, p) = \frac{1}{|\mathcal{C}|} \sum_{c_j \in \mathcal{C}} |\hat{p}(c_j) - p(c_j)| \tag{1}$$

It is easy to show that AE ranges between 0 (best) and

$$\frac{2(1 - \min_{c_j \in \mathcal{C}} p(c_j))}{|\mathcal{C}|}$$

(worst); a normalized version of AE that always ranges between 0 (best) and 1 (worst) can thus be obtained as

$$NAE(\hat{p}, p) = \frac{\sum_{c_j \in \mathcal{C}} |\hat{p}(c_j) - p(c_j)|}{2(1 - \min_{c_j \in \mathcal{C}} p(c_j))} \tag{2}$$

The main advantage of AE and NAE is that they are intuitive, and easy to understand to non-initiates too.

However, AE and NAE do not address the fact that the same absolute difference between predicted class prevalence and true class prevalence should count as a more serious mistake when the true class prevalence is small. For instance, predicting $\hat{p}(c) = 0.10$ when $p(c) = .01$ and predicting $\hat{p}(c) = 0.50$ when $p(c) = 0.41$ are equivalent errors according to AE, but the former is intuitively a more serious error than the latter. *Relative absolute error* (RAE) addresses this problem by relativizing the value $|\hat{p}(c_j) - p(c_j)|$ in Equation 1 to the true class prevalence, i.e.,

$$RAE(\hat{p}, p) = \frac{1}{|\mathcal{C}|} \sum_{c_j \in \mathcal{C}} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)} \tag{3}$$

RAE may be undefined in some cases, due to the presence of zero denominators. To solve this problem, in computing RAE we can smooth both $p(c_j)$ and $\hat{p}(c_j)$ via additive smoothing, i.e.,

$$p_s(c_j) = \frac{\epsilon + p(c_j)}{\epsilon|\mathcal{C}| + \sum_{c_j \in \mathcal{C}} p(c_j)} \tag{4}$$

where $p_s(c_j)$ denotes the smoothed version of $p(c_j)$ and the denominator is just a normalizing factor (same for the $\hat{p}_s(c_j)$'s); the quantity $\epsilon = \frac{1}{2|\mathcal{S}|}$ is often used as a smoothing factor. The smoothed versions of $p(c_j)$ and $\hat{p}(c_j)$ are then used in place of their original versions in Equation 3; as a result, RAE is always defined and still returns a value of 0 when p and $\hat{p}$ coincide. It is easy to show that RAE ranges between 0 (best) and

$$\frac{|\mathcal{C}| - 1 + \dfrac{1 - \min_{c_j \in \mathcal{C}} p(c_j)}{\min_{c_j \in \mathcal{C}} p(c_j)}}{|\mathcal{C}|}$$

(worst); a normalized version of RAE that always ranges between 0 (best) and 1 (worst) can thus be obtained as

$$NRAE(\hat{p}, p) = \frac{\displaystyle\sum_{c_j \in \mathcal{C}} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)}}{|\mathcal{C}| - 1 + \dfrac{1 - \min_{c_j \in \mathcal{C}} p(c_j)}{\min_{c_j \in \mathcal{C}} p(c_j)}} \tag{5}$$

A third measure, and the one that has become somehow standard in the evaluation of SLMC quantification, is *normalized cross-entropy*, better known as *Kullback-Leibler Divergence* (KLD – see e.g., [10]). KLD was proposed as a SLMC quantification measure in [14], and is defined as

$$KLD(\hat{p}, p) = \sum_{c_j \in \mathcal{C}} p(c_j) \log \frac{p(c_j)}{\hat{p}(c_j)} \tag{6}$$

KLD was originally devised as a measure of the inefficiency incurred when estimating a true distribution p over a set $\mathcal{C}$ of classes by means of a predicted distribution $\hat{p}$. KLD is thus suitable for evaluating quantification, since quantifying exactly means predicting how the items in set $\mathcal{S}$ are distributed across the classes in $\mathcal{C}$.

KLD ranges between 0 (best) and $+\infty$ (worst). Note that, unlike AE and RAE, the upper bound of KLD is not finite since Equation 6 has predicted probabilities, and not true probabilities, at the denominator: that is, by making a predicted probability $\hat{p}(c_j)$ infinitely small we can make KLD be infinitely large. A normalized version of KLD yielding values between 0 (best) and 1 (worst) may be defined by applying a logistic function, e.g.,

$$NKLD(\hat{p}, p) = \frac{e^{KLD(\hat{p},p)} - 1}{e^{KLD(\hat{p},p)}} \tag{7}$$

Also KLD (and, as a consequence, $NKLD$) may be undefined in some cases. While the case in which $p(c_j) = 0$ is not problematic (since continuity arguments indicate that $0 \log \frac{0}{a}$ should be taken to be 0 for any $a \geq 0$), the case in which $\hat{p}(c_j) = 0$ and $p(c_j) > 0$ is indeed problematic, since $a \log \frac{a}{0}$ is undefined for $a > 0$. To solve this problem, also in computing KLD and $NKLD$ we use the smoothed probabilities of Equation 4; as a result, KLD and $NKLD$ are always defined and still return a value of zero when p and $\hat{p}$ coincide.

While KLD is less easy to understand to non-initiates than AE or RAE, its advantage is that it is a very well-known measure, having been the subject of intense study within information theory [11] and, although from a more applicative angle, within the language modelling approach to information retrieval and to speech processing. As a consequence, it has emerged as the *de facto* standard in the SLMC quantification literature. We will thus pick it as the measure to optimize; however, in the experimental section we will report the results of all our experiments in terms of all six measures discussed above.

4. A tweet sentiment quantifier

In this section we will describe the quantification-specific system we will use in our experiments. We start (Section 4.1) by describing how to generate sentiment-oriented vectorial representations from tweets, while in Section 4.2 we describe the learning algorithm we adopt.

4.1. Features for detecting tweet sentiment

For building vectorial representation of tweets we have followed the approach discussed in [23, Section 5.2.1], since the representations presented therein are those used in the systems that performed best at both the SemEval 2013 [27] and SemEval 2014 [38] STC shared tasks.

The text is preprocessed by normalizing URLs and mentions of users to the constants `http://someurl` and `@someuser`, resp., after which tokenization and POS tagging is performed. Binary features (i.e., features denoting presence or absence) used include word n-grams, for $n \in \{1, 2, 3, 4\}$, and character n-grams, for $n \in \{3, 4, 5\}$, whether the last token contains an exclamation and/or a question mark, whether the last token is a positive or negative emoticon and, for each of the 1000 word clusters produced with the CMU Twitter NLP tool[2], where any token from the cluster is present. Integer-valued features include the number of all-caps tokens, the number of tokens for each POS tag, the number of hashtags, the number of negated contexts, the number of sequences of exclamation and/or question marks, and the number of elongated words (e.g., `cooooool`).

A key addition to the above is represented by features derived from both automatically generated and manually generated sentiment lexicons; for these features, we use the same sentiment lexicons as used in [23], which are all publicly available. We omit further details concerning our vectorial representations (and, in particular, how the sentiment lexicons contribute to them), both for brevity reasons and because these vectorial representations are not the central focus of this paper; the interested reader is invited to consult [23, Section 5.2.1] for details.

Finally, we should mention the fact that we did not perform any feature selection, since our learners could handle the resulting (huge) number of features fairly well from the standpoint of efficiency, and since learning algorithms in the SVM family are known to be fairly robust to overfitting.

4.2. Learning to quantify

As a state-of-the-art quantification algorithm we use SVM(KLD), introduced in [13]. SVM(KLD) is an instantiation of Thorsten Joachims' SVM-perf [19] that uses KLD as the loss to optimize[3].

SVM-perf is a "structured output prediction" algorithm in the support vector machines family. Unlike traditional SVMs, SVM-perf is capable of optimizing any nonlinear, multivariate loss function that can be computed from a contingency table (as all the measures presented in Section 3 are). Instead of handling hypotheses $h : \mathcal{X} \rightarrow \mathcal{Y}$ that map an individual item (in our case: a tweet) $\mathbf{x}_i$ into an individual label y_i, SVM-perf considers hypotheses $\bar{h} : \bar{\mathcal{X}} \rightarrow \bar{\mathcal{Y}}$ that map entire tuples of items (in our case: entire sets of tweets) $\bar{\mathbf{x}} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ into tuples of labels $\bar{\mathbf{y}} = (y_1, ..., y_n)$. Instead of learning the traditional hypotheses of type

$$h(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x} + b) \tag{8}$$

SVM-perf thus learns hypotheses of type

$$\bar{h}(\bar{\mathbf{x}}) = \arg\max_{\bar{y} \in \bar{\mathcal{Y}}} (\mathbf{w} \cdot \Psi(\bar{\mathbf{x}}, \bar{y})) \tag{9}$$

where $\mathbf{w}$ is the vector of parameters to be learnt during training and

$$\Psi(\bar{\mathbf{x}}, \bar{y}) = \sum_{i=1}^{n} \mathbf{x}_i y_i \tag{10}$$

(the *joint feature map*) is a function that scores the pair of tuples $(\bar{\mathbf{x}}, \bar{y})$ according to how "compatible" $\bar{\mathbf{x}}$ and $\bar{y}$ are. In other words, while classifiers trained via traditional SVMs classify individual instances $\mathbf{x}$ one at a time, models trained via SVM-perf classify entire sets $\bar{\mathbf{x}}$ of instances in one shot, and can thus make the labels assigned to the individual items mutually depend on each other. This is of fundamental importance in quantification, where, say, an additional false positive may even be *beneficial* when the rest of the data is expected to contain more false negatives than false positives.

While the optimization problem of classic soft-margin SVMs consists in finding

$$\begin{aligned} \arg\min_{\mathbf{w},\xi_i \geq 0} \quad & \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{|Tr|} \xi_i \\ \text{such that} \quad & y'_i[\mathbf{w} \cdot \mathbf{x}'_i + b] \geq (1 - \xi_i) \\ & \quad \text{for all } i \in \{1, ..., |Tr|\} \end{aligned} \tag{11}$$

(where the $(\mathbf{x}'_i, y'_i)$ denote the training examples and Tr indicates the training set) the corresponding problem of SVM-perf consists instead of finding

$$\begin{aligned} \arg\min_{\mathbf{w},\xi \geq 0} \quad & \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C\xi \\ \text{such that} \quad & \mathbf{w} \cdot [\Psi(\bar{\mathbf{x}}', \bar{y}') - \Psi(\bar{\mathbf{x}}', \bar{y}) + b] \\ & \quad \geq \Lambda(\bar{y}', \bar{y}) - \xi \text{ for all } \bar{y} \in \bar{\mathcal{Y}}/\bar{y}' \end{aligned} \tag{12}$$

2. http://www.ark.cs.cmu.edu/TweetNLP/

3. In [19] SVM-perf is actually called SVM-multi, but the author has released its implementation under the name SVM-perf; we will thus use this latter name.

where $(\bar{\mathbf{x}}', \bar{y}')$ indicates a sequence of training examples and the corresponding sequence of their true labels.

Here, the relevant thing to observe is that the multivariate loss Λ explicitly appears in the optimization problem.

We refer the interested reader to [19], [20], [37] for more details on SVM-perf (and on SVMs for structured output prediction in general). From the point of view of the user interested in applying it to a certain task, the implementation of SVM-perf made available by its author is essentially an off-the-shelf package, since for customizing it to a specific loss Λ one only needs to write a small module that describes how to compute Λ from a contingency table[4].

5. Experiments

We have carried out our experiments on a variety of TSC datasets previously used in the literature; the main characteristics of these datasets are listed in Table 1. The SemEval2013, SemEval2014, and SemEval2015 datasets are described more in detail in [28], [33], and [32], respectively, while all of the other datasets (Sanders, SST, OMD, HCR, GASP) are described in detail in [34]. Our choice of datasets has followed two main guidelines, i.e., (i) selecting publicly available datasets, so as to guarantee a high level of replicability, and (ii) selecting datasets whose sentiment labels are the result of manual annotation, so as to guarantee high label quality[5].

It is well known that, when Twitter data are concerned, the replicability of experimental results is limited since, due to terms of use imposed by Twitter, the datasets made available by researchers cannot contain the tweets themselves but only consists of their id's; the tweets corresponding to some of the id's may become unavailable over time, which means that the datasets we use here are typically subsets of the original datasets. Luckily enough, this problem affects us only marginally since (i) of the three SemEval datasets we owned an original copy before starting this research, and (ii) we were able to recover all of the original tweets in all of the other datasets (except for Sanders).

Most of the above datasets classify tweets across the three classes Positive, Negative, Neutral; some others (Sanders, OMD, HCR, GASP) also use additional classes (e.g., Mixed, Irrelevant, Other), and SST uses 10 different levels of sentiment strength (from VeryPositive to VeryNegative). For reasons of uniformity, we have removed the tweets belonging to the additional classes (OMD, HCR, GASP), and converted sentiment strengths into Positive, Negative, Neutral using the same heuristics as described in [34] (SST); in all of the datasets we use, the task is thus to quantify the Positive, Negative, Neutral classes, which represent a partition of the dataset.

Because of the reasons above, the numbers reported in Table 1 refer not to the original datasets but to the versions we have used[6].

5.1. Experimental protocol

In our experiments, on each dataset we compare the quantification-specific learning algorithm of Section 4.2 against a baseline consisting of a representative, state-of-the-art, classification-specific learning algorithm. As this baseline algorithm we use a standard support vector machine with a linear kernel, in the implementation made available in the LIBSVM system[7] [9]; it is a strong baseline, and is (among others) the learning algorithm used in the systems that performed best at both the SemEval 2013 [27] and SemEval 2014 [38] STC shared tasks. While SVM(KLD) explicitly minimizes KLD, the above baseline minimizes the well-known Hinge Loss; from now on we will thus refer to it as SVM(HL).

Both learning algorithms are fed the same vectorial representations, as described in Section 4.1. For both learning algorithms we have optimized the C parameter (which sets the tradeoff between the training error and the margin – see Equations 11 and 12) via validation on a separate held-out set, performing a grid search on all values of type 10^x with $x \in \{-6, ..., 7\}$; we have optimized C individually for each learner–dataset pair. We have instead left the other parameters at their default value; in particular, with both learning algorithms we have used a linear kernel. Some of the datasets we use (SemEval2013, SemEval2014, SemEval2015, and HCR) already come with a predefined split between training set and held-out set, with (for the SemEval datasets) roughly six times as many training items as held-out items; for the datasets where such split is not predefined, we have randomly selected the held-out examples from the training examples, using the same ratio as in the SemEval datasets. For all datasets, after the optimal parameter values have been selected we have retrained the classifier on the union of the training and the held-out sets.

As noted in Section 3, ours is a single-label multi-class task. This does not pose any problem to our baseline system, since LIBSVM is equipped with a built-in SLMC option; this ensures that the baseline is a strong one. It instead poses a problem to SVM(KLD), which is a binary learning algorithm. We circumvent this problem by (i)

4. SVM-perf is available from http://svmlight.joachims.org/svm_struct.html, while the module that customizes it to KLD is available from http://hlt.isti.cnr.it/quantification/

5. This means that we avoid STC datasets in which the labels are automatically derived from, say, the emoticons present in the tweets.

6. In order to enhance the reproducibility of our experimental results, we make available (at http://alt.qcri.org/~wgao/data/tweet_sentiment_quantification.zip) the vectorial representations we have generated for all the datasets (split into training / validation / test sets) used in this paper.

7. LIBSVM is available from http://www.csie.ntu.edu.tw/~cjlin/libsvm/

TABLE 1. Datasets used in this work and their main characteristics. The last column indicates distribution drift measured in terms of $KLD(p_{Te}, p_{Tr})$, i.e., indicates how much the distribution in the test set diverges from that in the training set; higher values indicate higher divergence.

Dataset	# of features used	# of training tweets	# of held-out tweets	# of test tweets	Total # of tweets	Distribution drift
SemEval2013	1,215,742	9,684	1,654	3,813	15,151	0.001662
SemEval2014	1,215,742	9,684	1,654	1,853	13,191	0.022222
SemEval2015	1,215,742	9,684	1,654	2,390	13,728	0.003723
Sanders	229,399	1,847	308	923	3,078	0.000010
SST	376,132	2,546	425	1,271	4,242	0.003603
OMD	199,151	1,576	263	787	2,626	0.000580
HCR	222,046	797	797	798	2,392	0.008174
GASP	694,582	7,532	1,256	3,765	12,553	0.000187

using SVM(KLD) to train $|\mathcal{C}|$ "one-against-all" binary predictors, (ii) having each binary predictor output a prevalence estimate for the corresponding class, and (iii) normalizing these prevalence estimates so that they sum up to 1.

5.2. Results

The results of our experiments are reported in Table 2. SVM(KLD) is explicitly designed to perform well when KLD is the evaluation measure, so let's first look at the column reporting KLD results. This column shows that SVM(KLD) outperforms SVM(HL) on 6 out of 8 datasets, only being outperformed on HCR and marginally outperformed on SemEval2015. SVM(KLD) also outperforms SVM(HL) on average (across the 8 tested datasets), bringing about a 29.4% reduction in prevalence estimation error (.022 vs. 0.31) as measured by KLD.

We can also see that the results measured according to the other five metrics substantially confirm the KLD results: RAE, $NRAE$, and $NKLD$ always agree with KLD on who the best performer is, while AE and NAE almost always agree, the exceptions being SemEval2015 (where SVM(HL) is no more the winner, and the two learners are judged as being equally good) and GASP (where AE and NAE decree SVM(HL) to be the best performer)[8].

One of the reasons why SVM(KLD) did not *always* outperform SVM(HL) might reside in the fact that, as shown by the experiments in [13] (see Table II of that paper), SVM(KLD) especially excels at prevalence estimation for classes with low ($< .10$) class prevalence in the training set, while other quantification-specific learning algorithms seem to perform better for more frequent classes. In the future we plan to repeat the experiments reported in Table 2 also by using these latter algorithms.

8. That KLD and $NKLD$ always agree is true by definition, since they are monotonically increasing functions of each other; same for AE and NAE, and for RAE and $NRAE$.

6. Conclusion

In this paper we have argued that the real goal of most research efforts dealing with in the sentiment conveyed by tweets is not classification, but quantification (i.e., prevalence estimation). As a result, those who pursue this goal by using the learning algorithms and evaluation measures that are standard in the classification arena may obtain inaccurate prevalence estimates. We have experimentally shown, on a multiplicity of tweet sentiment classification (TSC) datasets, that more accurate prevalence estimates may be obtained by considering quantification as a task in its own right, i.e., by using (i) learning algorithms specifically optimised for quantification accuracy and (ii) evaluation metrics that directly measure the accuracy of prevalence estimates. Adopting a quantification-specific approach in gauging tweet sentiment may benefit many applications, especially in fields (such as political science, social science, and market research) that are usually less interested in finding the needle in the haystack than in characterising the haystack itself.

Finally, we want to note that, while this paper has addressed *sentiment* classification, the same arguments we have made apply to many studies where tweets are classified along dimensions other than sentiment. For instance, aggregate (rather than individual) results from tweet classification are the real goal in [1], which analyses Twitter data in order to predict box office revenues for movies; in [30], whose authors try to determine the percentage of tweets that are about infrastructure damage vs. those which are about donations, in order to do rapid damage assessment during humanitarian crises;

TABLE 2. Quantification accuracy (last six columns) obtained with a classification-oriented learning algorithm (SVM(HL)) and a quantification-oriented learning algorithm (SVM(KLD)) on several STC datasets; the two bottom rows indicate the average performance of each learner across all the datasets. The Pos, Neg, Neu columns indicate (true or predicted) prevalences. **Boldface** indicates the best system.

Dataset	System	Pos	Neg	Neu	AE	NAE	RAE	$NRAE$	KLD	$NKLD$
SemEval2013	[Gold Standard]	.412	.158	.430	—	—	—	—	—	—
	SVM(HL)	.318	.107	.575	.096	.172	.295	.121	.043	.042
	SVM(KLD)	.304	.158	.538	**.072**	**.129**	**.172**	**.070**	**.029**	**.029**
SemEval2014	[Gold Standard]	.530	.109	.361	—	—	—	—	—	—
	SVM(HL)	.404	.084	.513	.101	.170	.297	.088	.046	.045
	SVM(KLD)	.403	.124	.473	**.084**	**.142**	**.227**	**.067**	**.033**	**.033**
SemEval2015	[Gold Standard]	.434	.153	.413	—	—	—	—	—	—
	SVM(HL)	.270	.139	.591	**.119**	**.210**	**.301**	**.120**	**.073**	**.070**
	SVM(KLD)	.256	.225	.519	**.119**	**.210**	.379	.151	.076	.073
Sanders	[Gold Standard]	.149	.164	.687	—	—	—	—	—	—
	SVM(HL)	.080	.137	.784	.064	.113	.257	.100	.033	.032
	SVM(KLD)	.142	.184	.674	**.013**	**.024**	**.063**	**.024**	**.001**	**.001**
SST	[Gold Standard]	.312	.207	.481	—	—	—	—	—	—
	SVM(HL)	.223	.217	.560	.060	.113	.167	.086	.022	.022
	SVM(KLD)	.304	.269	.427	**.041**	**.078**	**.146**	**.075**	**.011**	**.011**
OMD	[Gold Standard]	.280	.437	.283	—	—	—	—	—	—
	SVM(HL)	.238	.537	.225	.067	.139	.195	.128	.020	.020
	SVM(KLD)	.306	.456	.238	**.030**	**.063**	**.100**	**.066**	**.006**	**.006**
HCR	[Gold Standard]	.193	.167	.640	—	—	—	—	—	—
	SVM(HL)	.139	.173	.687	**.036**	**.065**	**.130**	**.056**	**.011**	**.011**
	SVM(KLD)	.131	.221	.648	.041	.075	.219	.094	.020	.020
GASP	[Gold Standard]	.086	.407	.506	—	—	—	—	—	—
	SVM(HL)	.066	.415	.519	**.014**	**.023**	.095	.023	.003	.003
	SVM(KLD)	.091	.428	.481	.017	.028	**.053**	**.013**	**.001**	**.001**
Average	SVM(HL)	—	—	—	.070	.126	.217	.090	.031	.031
	SVM(KLD)	—	—	—	**.052**	**.094**	**.170**	**.070**	**.022**	**.022**

in [36], where hay fever maps are generated from geo-located tweets of fever-stricken people; in [17], where the authors generate a heat map of a natural disaster from geo-located tweets that report on it; and in many others.

The present paper thus urges researchers involved in tweet mining to take the distinction between classification and prevalence estimation at heart, and optimize their systems accordingly.

Acknowledgements

We are grateful to Chih-Chung Chang and Chih-Jen Lin for making LIBSVM available, to Thorsten Joachims for making SVM-perf available, to Andrea Esuli for making available the code for obtaining SVM(KLD) from SVM-perf, and to Carlos Castillo for several pointers to the literature.

References

[1] Sitaram Asur and Bernardo A. Huberman. Predicting the future with social media. In *Proceedings of the 10th IEEE/WIC/ACM International Conference on Web Intelligence (WI 2010)*, pages 492–499, Toronto, CA, 2010.

[2] Jose Barranquero, Jorge Díez, and Juan José del Coz. Quantification-oriented learning based on reliable classifiers. *Pattern Recognition*, 48(2):591–604, 2015.

[3] Jose Barranquero, Pablo González, Jorge Díez, and Juan José del Coz. On the study of nearest neighbor algorithms for prevalence estimation in binary problems. *Pattern Recognition*, 46(2):472–482, 2013.

[4] Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. Quantification via probability estimators. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2010)*, pages 737–742, Sydney, AU, 2010.

[5] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[6] Javier Borge-Holthoefer, Walid Magdy, Kareem Darwish, and Ingmar Weber. Content and network dynamics behind Egyptian political polarization on Twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW 2015)*, pages 700–711, Vancouver, CA, 2015.

[7] S. Burton and A. Soboleva. Interactive or reactive? Marketing with Twitter. *Journal of Consumer Marketing*, 28(7):491–499, 2011.

[8] Yee Seng Chan and Hwee Tou Ng. Estimating class priors in domain adaptation for word sense disambiguation. In *Proceedings of the 44th Annual Meeting of the Association for*

Computational Linguistics (ACL 2006), pages 89–96, Sydney, AU, 2006.

[9] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 2011.

[10] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, New York, US, 1991.

[11] Imre Csiszár and Paul C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, 2004.

[12] Peter S. Dodds, Kameron D. Harris, Isabel M. Kloumann, Catherine A. Bliss, and Christopher M. Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter. *PLoS ONE*, 6(12), 2011.

[13] Andrea Esuli and Fabrizio Sebastiani. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery and Data*, 9(4):1–27, 2015.

[14] George Forman. Counting positives accurately despite inaccurate classification. In *Proceedings of the 16th European Conference on Machine Learning (ECML 2005)*, pages 564–575, Porto, PT, 2005.

[15] George Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008.

[16] Víctor González-Castro, Rocío Alaiz-Rodríguez, and Enrique Alegre. Class distribution estimation based on the Hellinger distance. *Information Sciences*, 218:146–164, 2013.

[17] Benjamin Herfort, Svend-Jonas Schelhorn, João P. de Albuquerque, and Alexander Zipf. Does the spatiotemporal distribution of tweets match the spatiotemporal distribution of flood phenomena? A study about the river Elbe flood in June 2013. In *Proceedings of the 11th International Conference on Information Systems for Crisis Response and Management (ISCRAM 2014)*, pages 747–751, Philadelphia, US, 2014.

[18] Daniel J. Hopkins and Gary King. A method of automated nonparametric content analysis for social science. *American Journal of Political Science*, 54(1):229–247, 2010.

[19] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)*, pages 377–384, Bonn, DE, 2005.

[20] Thorsten Joachims, Thomas Hofmann, Yisong Yue, and Chun-Nam Yu. Predicting structured objects with support vector machines. *Communications of the ACM*, 52(11):97–104, 2009.

[21] Mesut Kaya, Guven Fidan, and Ismail Hakki Toroslu. Transfer learning using Twitter data for improving sentiment classification of Turkish political news. In *Proceedings of the 28th International Symposium on Computer and Information Sciences (ISCIS 2013)*, pages 139–148, Paris, FR, 2013.

[22] Gary King and Ying Lu. Verbal autopsy methods with multiple causes of death. *Statistical Science*, 23(1):78–91, 2008.

[23] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

[24] Micol Marchetti-Bowick and Nathanael Chambers. Learning for microblogs with distant supervision: Political forecasting with Twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 603–612, Avignon, FR, 2012.

[25] Eugenio Martínez-Cámara, Maria Teresa Martín-Valdivia, Luis Alfonso Ureña López, and Arturo Montejo Ráez. Sentiment analysis in Twitter. *Natural Language Engineering*, 20(1):1–28, 2014.

[26] Letizia Milli, Anna Monreale, Giulio Rossetti, Fosca Giannotti, Dino Pedreschi, and Fabrizio Sebastiani. Quantification trees. In *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM 2013)*, pages 528–536, Dallas, US, 2013.

[27] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 321–327, Atlanta, US, 2013.

[28] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. SemEval-2013 Task 2: Sentiment analysis in Twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, US, 2013.

[29] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the 4th AAAI Conference on Weblogs and Social Media (ICWSM 2010)*, Washington, US, 2010.

[30] Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW 2015)*, pages 994–1009, Vancouver, CA, 2015.

[31] Muhammad A. Qureshi, Colm O'Riordan, and Gabriella Pasi. Clustering with error estimation for monitoring reputation of companies on Twitter. In *Proceedings of the 9th Asia Information Retrieval Societies Conference (AIRS 2013)*, pages 170–180, Singapore, SN, 2013.

[32] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. SemEval-2015 Task 10: Sentiment analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, US, 2015.

[33] Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. SemEval-2014 Task 9: Sentiment analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80, Dublin, IE, 2014.

[34] Hassan Saif, Miriam Fernez, Yulan He, and Harith Alani. Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold. In *Proceedings of the 1st International Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM 2013)*, pages 9–21, Torino, IT, 2013.

[35] Lidia Sánchez, Víctor González, Enrique Alegre, and Rocío Alaiz. Classification and quantification based on image analysis for sperm samples with uncertain damaged/intact cell proportions. In *Proceedings of the 5th International Conference on Image Analysis and Recognition (ICIAR 2008)*, pages 827–836, Póvoa de Varzim, PT, 2008.

[36] Tetsuro Takahashi, Shuya Abe, and Nobuyuki Igata. Can Twitter be an alternative of real-world sensors? In *Proceedings of the 14th International Conference on Human-Computer Interaction (HCI International 2011)*, pages 240–249, Orlando, US, 2011.

[37] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[38] Xiaodan Zhu, Svetlana Kiritchenko, and Saif M. Mohammad. NRC-Canada-2014: Recent improvements in the sentiment analysis of tweets. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 443–447, Dublin, IE, 2014.