Evaluation Measures for the SemEval-2016 Task 4
“Sentiment Analysis in Twitter”
(Draft: Version 1.13)

Preslav Nakov♣, Alan Ritter♦, Sara Rosenthal♦, Fabrizio Sebastiani♣, Veselin Stoyanov♣
♣Qatar Computing Research Institute, Hamad bin Khalifa University, Qatar
♦Department of Computer Science and Engineering, Ohio State University, US
♥Department of Computer Science, Columbia University, US
♠Facebook, US

This informal document details the evaluation measures that will be used in SemEval-2016 Task 4 “Sentiment Analysis in Twitter”, a revamped edition of SemEval-2015 Task 10 (Rosenthal et al., 2015). Note: changes that have been made to the present document since Version 1.0 onwards are typeset in blue.

Task 4 consists of five subtasks; the evaluation measures that we will use for them will be discussed in Sections 1 to 5. Subtasks B to E conceptually form a “2×2 matrix” (see Table 1), where the rows indicate the goal of the task (classification vs. quantification) and the columns indicate the granularity of the task (two-point scale vs. five-point scale).

Note that, for Subtasks B to E, the dataset is subdivided into a number of “topics”, and the subtask needs to be carried out independently for each topic. As a result, each of the evaluation measures described below is “macroaveraged” across the topics, i.e., we compute the measure individually for each topic, and we then average the results across the topics.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Classification</th>
<th>Subtask B</th>
<th>Subtask C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantification</td>
<td>Subtask D</td>
<td>Subtask E</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: The “2×2 matrix” of Subtasks B-E.

1 Subtask A: Message Polarity Classification

Subtask A consists of the following problem:
Given a tweet, predict whether the tweet is of positive, negative, or neutral sentiment. It is thus a “single-label multi-class” classification (SLMCC) task, in which each tweet must be classified as belonging to exactly one of the three classes \( C = \{ \text{Positive}, \text{Neutral}, \text{Negative} \} \). This subtask is a rerun; it was also present in SemEval-2013 (Nakov et al., 2013), SemEval-2014 (Rosenthal et al., 2014), SemEval-2015 (Rosenthal et al., 2015) as Subtask B.

For reasons of continuity with the 2013-2015 editions of this subtask, we will adopt the same evaluation measure that was used then, i.e.,

\[
F_{1}^{PN} = \frac{F_{1}^{Pos} + F_{1}^{Neg}}{2} \tag{1}
\]

\( F_{1}^{Pos} \) is defined

- by taking \( \rho^{Pos} \) to be the fraction of Positive tweets that are predicted to be such; in terms of the confusion matrix of Table 2, this means that \( \rho^{Pos} = \frac{PP}{PP + UP + NP} \);
- by taking \( \pi^{Pos} \) to be the fraction of tweets predicted to be Positive that are indeed Positive, i.e., \( \pi^{Pos} = \frac{PP}{PP + PU + PN} \);
- by taking \( F_{1}^{Pos} = \frac{2\pi^{Pos}\rho^{Pos}}{\pi^{Pos} + \rho^{Pos}} \).

\( F_{1}^{Neg} \) is defined similarly, and the evaluation measure we finally adopt is \( F_{1}^{PN} \) as from Equation 1.

2 Subtask B: Tweet classification according to a two-point scale

Subtask B consists of the following problem:
Given a tweet known to be about a given topic, classify whether the tweet conveys a positive or a negative sentiment towards the topic. As such, it is thus a binary classification task, in which each tweet must be classified as belonging to exactly one of the two classes \( C = \{ \text{Positive}, \text{Negative} \} \). This subtask is a simplification of Subtask C as from SemEval-2015,
which also required to filter out tweets that were not about the topic, and which (like Subtask A does now – see Section 1) also involved the **Neutral** class.

As an evaluation measure, for this task we will adopt macroaveraged recall, i.e.,

\[
\rho^{PN} = \frac{\rho^{Pos} + \rho^{Neg}}{2}
\]

where \(\rho^{Pos}\) and \(\rho^{Neg}\) are as defined in Section 1; note in fact that, since we do not have the **Neutral** class here, PU UP UU UN NU from the confusion matrix of Table 2 are 0. \(\rho^{PN}\) ranges in \([0, 1]\), where 1 is achieved only by the perfect classifier (the classifier that correctly classifies all items), 0 is achieved only by the perverse classifier (the classifier that misclassifies all items), while 0.5 is

- the value obtained by a trivial classifier (i.e., the classifier that assigns all tweets to the same class – be it **Positive or Negative**), and
- the expected value of a random classifier.

The advantage of \(\rho^{PN}\) over “standard” accuracy is that it is more robust to class imbalance, since for standard accuracy the score of the majority-class classifier is the relative frequency (aka “prevalence”) of the majority class, that may be much higher than 0.5 if the test set is imbalanced.

The advantage of \(\rho^{PN}\) over \(F_1\) is that it is more robust to class imbalance, since for \(F_1\) the score of the trivial acceptor may be much higher than 0.5 if the test set is imbalanced and the **Positive** class is the majority class. Another advantage of \(\rho^{PN}\) over \(F_1\) is that \(\rho^{PN}\) is invariant with respect to switching **Positive** with **Negative**, while \(F_1\) is not.

### 3 Subtask C: Tweet classification according to a five-point scale

Subtask C consists of the following problem: *Given a tweet known to be about a given topic, estimate the sentiment conveyed by the tweet towards the topic on a five-point scale*. As such, it is thus an ordinal classification (OC – also known as *ordinal regression*) task, in which each tweet must be classified in exactly one of the classes in \(C = \{\text{VeryPositive}, \text{Positive}, \text{OK}, \text{Negative}, \text{VeryNegative}\}\) (represented in our dataset by numbers in \([-2, +1, 0, -1, -2]\)), where there is a total order defined on \(C\). This subtask was not present in SemEval-2015.

The essential difference between SLMCC (see Section 1) and OC is that in the latter not all mistakes weigh equally; e.g., classifying as **VeryNegative** an item that should be classified as **VeryPositive** is a more serious mistake than classifying as **VeryNegative** an item that should be classified as **Negative**.

As our evaluation measure, we use macroaveraged mean absolute error (\(MAE^M\)):

\[
MAE^M(h, T_e) = \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{1}{|T_{e_j}|} \sum_{x_i \in T_{e_j}} |h(x_i) - y_i| \tag{3}
\]

where \(y_i\) denotes the true label of item \(x_i\), \(h(x_i)\) denotes its predicted label, \(T_{e_j}\) denotes the set of test documents whose true class is \(c_j\), \(|h(x_i) - y_i|\) denotes the “distance” between classes \(h(x_i)\) and \(y_i\) (e.g., the distance between **veryPositive** and **Negative** is 3), and the “M” superscript indicates “macroaveraging”.

The advantage of \(MAE^M\) over “standard” mean absolute error, which is defined as

\[
MAE^\mu(h, T_e) = \frac{1}{|T_e|} \sum_{x_i \in T_e} |h(x_i) - y_i| \tag{4}
\]

where the “\(\mu\)” superscript stands for “microaveraging”, is that it is robust to class imbalance (which is useful, given the imbalanced nature of our dataset) while coinciding with \(MAE^\mu\) on perfectly balanced datasets (i.e., datasets with exactly the same number of test documents for each class).
Note that, unlike the measures discussed in Sections 1 and 2, \( MAE^M \) is a measure of error, and not a measure of accuracy, so lower values are better. See (Baccianella et al., 2009) for more detail on \( MAE^M \).

4 Subtask D: Tweet quantification according to a two-point scale

Subtask D consists of the following problem: Given a set of tweets known to be about a given topic, estimate the distribution of the tweets across the Positive and Negative classes. It is thus a binary quantification task, in which each tweet belongs exactly to one of the classes in \( \mathcal{C} = \{ \text{Positive}, \text{Negative} \} \) and the task is to compute an estimate \( \hat{p}(c_j) \) of the relative frequency in the test set \( p(c_j) \) of each of the classes in \( \mathcal{C} \). This is subtask is related to (yet, different from) SemEval-2015 subtask E.

The essential difference between binary classification (as from Section 2) and binary quantification is that, in the latter, errors of different polarity (e.g., a false positive and a false negative for the same class) compensate each other.

The measure we are going to adopt is normalized cross-entropy, better known as Kullback-Leibler Divergence (KLD). KLD was proposed as a quantification measure in (Forman, 2005), and is defined as follows:

\[
KLD(\hat{p}, p, \mathcal{C}) = \sum_{c_j \in \mathcal{C}} p(c_j) \log \frac{p(c_j)}{\hat{p}(c_j)}
\]

\( KLD \) is a measure of the error made in estimating a true distribution \( p \) over a set \( \mathcal{C} \) of classes by means of a predicted distribution \( \hat{p} \). Like \( MAE^M \) in Section 3, \( KLD \) is a measure of error, so lower values are better. \( KLD \) ranges between 0 (best) and \( +\infty \) (worst).

Note that the upper bound of \( KLD \) is not finite since Equation 5 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 \( KLD \) infinitely large. To solve this problem, in computing \( KLD \) we smooth both \( p(c_j) \) and \( \hat{p}(c_j) \) via additive smoothing, i.e.,

\[
p_s(c_j) = \frac{p(c_j) + \epsilon}{\left( \sum_{c_j \in \mathcal{C}} p(c_j) \right) + \epsilon \cdot |\mathcal{C}|} = \frac{p(c_j) + \epsilon}{1 + \epsilon \cdot |\mathcal{C}|}
\]

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 \( \epsilon = \frac{1}{2|\mathcal{C}|} \) is used as a smoothing factor, where \( T_e \) denotes the test set. The smoothed versions of \( p(c_j) \) and \( \hat{p}(c_j) \) are then used in place of their original versions in Equation 5; as a result, \( KLD \) is always defined and still returns a value of 0 when \( p \) and \( \hat{p} \) coincide.

5 Subtask E: Tweet quantification according to a five-point scale

Subtask E consists of the following problem: Given a set of tweets known to be about a given topic, estimate the distribution of the tweets across the five classes of a five-point scale.

It is an ordinal quantification (OQ) task, in which (as in OC) each tweet belongs exactly to one of the classes in \( \mathcal{C} = \{ \text{VeryPositive}, \text{Positive}, \text{OK}, \text{Negative}, \text{VeryNegative} \} \), where there is a total order on \( \mathcal{C} \) and (as in binary quantification) the task is to compute an estimate \( \hat{p}(c_j) \) of the relative frequency \( p(c_j) \) in the test tweets of all the classes \( c_j \in \mathcal{C} \). This subtask was not present in SemEval-2015.

The measure we adopt for OQ is the Earth Mover’s Distance (Rubner et al., 2000), a measure well known in the field of computer vision. When there is a total order on the classes in \( \mathcal{C} \), the Earth Mover’s Distance is defined as

\[
EMD(\hat{p}, p) = \sum_{j=1}^{|\mathcal{C}|-1} \left| \sum_{i=1}^{j} \hat{p}(c_i) - \sum_{i=1}^{j} p(c_i) \right|
\]

and can be computed in \( |\mathcal{C}| \) steps from the estimated and true class prevalences. Like \( KLD \) in Section 4, \( EMD \) is a measure of error, so lower values are better; \( EMD \) ranges between 0 (best) and \( |\mathcal{C}| - 1 \) (worst). See (Esuli and Sebastiani, 2010) for more detail on \( EMD \).
A Appendix: Useful pointers

Quantification. Several publications in the literature discuss methods for binary quantification: see e.g., (Ala´ız-Rodr´ıguez et al., 2011; Barranquero et al., 2015; Esuli and Sebastiani, 2015; Forman, 2008; Hopkins and King, 2010; Milli et al., 2013; Saerens et al., 2002). Some of these papers, e.g., (Esuli and Sebastiani, 2015; Hopkins and King, 2010), contain links for downloading the software for performing quantification. Sentiment quantification is discussed in (Esuli and Sebastiani, 2010); tweet sentiment quantification is discussed in (Gao and Sebastiani, 2015).

Ordinal classification. Ordinal classification has a very rich literature; papers proposing OC methods include, e.g., (Chu and Keerthi, 2007; Dembczy´nski et al., 2007; Fouad and Tino, 2012; Herbrich et al., 2000; Li and Lin, 2007; Lin and Li, 2006; Lin and Li, 2012; Sun et al., 2010; Xia et al., 2006). A survey on ordinal classification methods can be found in (Gutiérrez et al., 2015). Some of these papers, e.g., (Chu and Keerthi, 2007), contain links for downloading software performing OC.

References


Andrea Esuli and Fabrizio Sebastiani. 2015. Optimizing text quantifiers for multivariate loss functions. ACM Transactions on Knowledge Discovery and Data, 9(4):Article 27.


