# ISTI-CNR at SemEval-2016 Task 4: Quantification on an Ordinal Scale

**Andrea Esuli**
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo"
via G. Moruzzi, 1
56124, Pisa, ITALY
andrea.esuli@isti.cnr.it

## Abstract

This paper details on the participation of ISTI-CNR to task 4 of Semeval 2016. Among the five subtasks, special attention has been paid to the five-point scale quantification subtask. The quantification method we propose is based on the observation that a standard document-by-document regression method usually has a bias towards assigning high prevalence labels. Our method models such bias with a linear model, in order to compensate it and to produce the quantification estimates.

## 1 Introduction

The participation of ISTI-CNR to task 4 of Semeval 2016 (Nakov et al., 2016) produced submissions for all the five proposed subtasks.

Submissions for subtasks A and B are based on a relatively typical machine learning pipeline, with B used as the base classification tool for subtask D, which uses a quantification via classification method. Subtask C uses an ordinal regression method, based on building a data-balanced tree of binary classifiers. The regression method of subtask C has been used as the base regression tool for the implementation of the quantification method for subtask E.

We propose a novel quantification method for subtask E, tweet quantification according to a five-point scale. The method stems from the intuition of measuring and compensating the bias a regression model may have for the labels with high prevalences.

All the code we produced for these tasks is mainly based on the scikit-learn python library (Pedregosa et al., 2011) and it is published under an open source license[1].

The next sections detail on the data and methods adopted to produce the five submissions.

## 2 Training data

The labeled training dataset has been downloaded using the tool suggested by the organizers[2]. Table 1 summarizes the total number of tweets available for download at the time of crawling (November 2015). The final number of tweets used to train the classifiers, or quantifiers, is 6223 for subtasks A, C and E, and 4475 for subtasks B and D.

A small number of tweets in the dataset appeared multiple times, some with conflicting labels. In the "train" data parts, for example, 25 tweets appeared twice for subtasks A, C, and E, six of them with conflicting labels for subtask A, and 14 with conflicting labels for subtasks C and E[3]. For subtasks B, D, the number of tweets appearing twice in training is 13, one of them with conflicting labels. Duplicate tweets have been reduced to a single instance and those with conflicting labels have been excluded from the dataset and from any analysis performed in this work.

[1] https://github.com/aesuli/semeval2016-task4

[2] https://github.com/aritter/twitter_download

[3] Subtask A uses a coarse-grained three-point scale, so it may happen that a conflicting '-1'/'-2' labeling of a tweet for subtasks C and E is reduced to a non-conflicting 'negative' labeling for subtask A.



*Proceedings of SemEval-2016*, pages 92–95,
San Diego, California, June 16-17, 2016. 

| subtask | set | labeled | downloaded |
|---|---|---|---|
| A | train | 6000 | 3804 |
| A | dev | 2000 | 1229 |
| A | devtest | 2000 | 1190 |
| A | *all* | 10000 | 6223 |
| C,E | train | 6000 | 3804 |
| C, E | dev | 2000 | 1229 |
| C, E | devtest | 2000 | 1190 |
| C, E | *all* | 10000 | 6223 |
| B, D | train | 4346 | 2764 |
| B, D | dev | 1325 | 836 |
| B, D | devtest | 1417 | 875 |
| B, D | *all* | 7088 | 4475 |

**Table 1:** Number of tweets with labeling provided by the organizers and number of such tweets available for download at the time of crawling.

Training data for subtasks A and B has been enriched by adding 5331 positive and 5331 negative sentences extracted from movie reviews, which are part of the movie review dataset (Pang and Lee, 2005)[4]. Even though these sentences are domain-specific, they are deemed to contribute to the learning process by enriching the vocabulary of expressions used to denote positive and negative sentiments. The final training set for subtask A is thus composed of 16885 examples, and 15137 for subtask B.

### 2.1 Features

The transformation of each tweet into its vectorial representation uses a relatively simple processing. The text of each tweet is tokenized, stopwords are removed. Word bigrams and trigrams, and character fourgrams are added to representation. Regular expressions are used to detect mentions, hashtag, URLs, and emoticons, and metafeatures for each of these special type of information are added to the representation, e.g., if a tweet has two hashtags, the '_hashtag' feature with frequency two is added to the representation of the tweet. The vectors are weighted by $tf \cdot idf$. Feature selection based on $\chi 2$ is used to retain only the $x$ most informative features, with $x$ determined for each subtask with a cross-validation on training data.

[4] http://www.cs.cornell.edu/people/pabo/movie-review-data/

| method | $MAE^M$ |
|---|---|
| BBTOR | 0.927 |
| DDAG | 1.227 |
| SVORIM | 1.066 |

**Table 2:** Subtask C: comparison of ordinal regression methods, based on 10-fold cross-validation on training data.

## 3 Subtasks A and B: classification

A linear SVM has been used for for both classification tasks: a simple binary classifier for subtask B, and three *one-vs-all* binary classifiers for subtask A. The value of the parameter $C$ of the SVM has been determined with a cross-validation on training data.

## 4 Subtask C: regression

The *Balanced Binary Tree for Ordinal Regression* (BBTOR) method we designed for subtask C is based on building a tree of binary classifiers that recursively split the ordinal scale on the points of maximum balance in the number of training example assigned to the two sides of the binary classification problem.

For example, let's suppose to have a dataset with the following distribution of training examples: $|c_1| = 20$, $|c_2| = 10$, $|c_3| = 20$, $|c_4| = 30$, $|c_5| = 50$, where $|c_i| = n$ means that label $c_i$ has $n$ training examples. The first binary classifier learns to separate $\{c_1, c_2, c_3\}$ from $\{c_4, c_5\}$, given that the partion with 50 vs 80 training examples is the most balanced one[5]. Then two second-level binary classifiers are trained on the $\{c_1, c_2\}$ vs $\{c_3\}$ split and the $\{c_4\}$ vs $\{c_5\}$ split. The training is completed learning a $\{c_1\}$ vs $\{c_2\}$ classifier. A linear SVM is used to train the binary classifier, optimizing its $C$ parameter with a cross-validation on training data.

This approach is in line with the proposal of *Data-Balanced Nested Dichotomies* of Dong et al. (2005) for multi-class problems, and extends it to consider the ordinal relations between labels. The method has been compared in cross-validation experiments on

[5] Note that also the $\{c_1, c_2, c_3.c_4\}$ from $\{c_5\}$ split produces an equivalent 80 vs 50 split. A second criterium is to prefer the splits in which also the number of labels is more balanced. In case of a tie also on the second criterium a random choice is made.

| method | *KLD* |
|---|---|
| CC | 0.742 |
| ACC | 0.756 |
| PCC | 0.230 |
| PACC | 0.319 |

**Table 3:** Subtask D: comparison of binary quantification methods, based on leave-one-topic-out validation on training data.

| method | *EMD* |
|---|---|
| RC | 0.547 |
| ARC | 0.374 |

**Table 4:** Subtask E: comparison of ordinal regression quantification methods, based on leave-one-topic-out validation on training data.

training data against other regression methods, i.e., SVORIM (Chu and Keerthi, 2007), based on linear regression, and DDAG (Aiolli et al., 2009), based on binary classifiers, and produced the best performance.

## 5 Subtask D: binary quantification

Four quantification methods based on classification have been compared, following the works of Forman (2008) and Bella et al. (2010). The four methods are: *classify and count* (CC), in which a classifier is applied to the test documents and the prevalences are determined by counting the documents assigned to each label; *adjusted classify and count* (ACC), in which the output of the CC method is corrected to take into account the bias in error towards one of the two labels the classifier may have; *probabilistic classify and count* (PCC) in which the contribution of each document to the counting is weighted on the confidence the classifier has on the assignment; *probabilistic adjusted classify and count* (PACC) which is the ACC method applied to the probabilistic model of PCC. From a cross validation on training data, in which each topic has been in turn used as test data and the remaining as training data, the PCC performed best and it was thus used for the final submission.

## 6 Subtask E: quantification on an ordinal scale

Two methods have been compared for subtask E. One is a simple *regress and count* (RC) method in which the BBTOR method used in subtask C is applied to documents of a topic and then the quantification values for the topic is determined by counting the number of documents assigned to each slot in the ordinal scale. We propose the *adjusted regress and count* (ARC) method, that is based on the intuition to measure, and compensate, the typical bias of regression methods to assign documents to the slots in the ordinal scale that have higher prevalences.

Let's denote the prevalences for a topic-label pair with $p_j(c_i)$, where $j$ indicates a topic in the set of topics $\{t_1, \ldots, t_n\}$ and $i$ a label in the set of ordered labels $\{c_1, \ldots, c_m\}$ that form the ordinal scale. On a given set of topics, the cumulative prevalence for each label is denoted as $P(c_i) = \sum_{j=1}^{n} p_j(c_i)$. Given a quantification method that produces estimations $\hat{p}_j(c_i)$, its cumulative prevalences are denoted as $\hat{P}(c_i) = \sum_{j=1}^{n} \hat{p}_j(c_i)$.

Under the hypothesis of a linear error model, knowing the estimate prevalences and the cumulative correct and estimate prevalences on a set of topics, the true prevalence for a topic can be determined as:

$$p_j(c_i) \simeq \frac{P(c_i)}{\hat{P}(c_i)} \hat{p}_j(c_i) = w_i \hat{p}_j(c_i) \qquad (1)$$

Note that the model uses a different linear correction weight $w_i = \frac{P(c_i)}{\hat{P}(c_i)}$ for each label $c_i$.

The correction value $w_i$ cannot be determined on the test data, since $P(c_i)$ is unknown. Following the ACC method for binary quantification (Forman, 2008) that estimates its correction parameter on the training set, also the $w_i$ values can be approximated on the training data using cross-validation, substituting $w_i$ with the $w_i^{Tr} = \frac{P^{Tr}(c_i)}{\hat{P}^{Tr}(c_i)}$ value. In this way the ARC quantification estimate can be derived from the RC estimate using the formula :

$$\hat{p}_j^{\mathrm{ARC}}(c_i) = \frac{1}{Z_j} w_i^{Tr\mathrm{RC}} \hat{p}_j^{\mathrm{RC}}(c_i) \qquad (2)$$

where $Z_j = \sum_{i=1}^{m} \hat{p}_j^{\mathrm{ARC}}(c_i)$ is a normalization factor to guarantee that the prevalences for a topic sum up to one.

The ARC method produced a sensible improvement over RC on a leave-one-topic-out validation on training data as reported in Table 4.

## 7 Future work

The features extracted from text in these experiments are based on a traditional vector space model in which each distinct feature is a represented by a dedicated dimension in the vector space. The limited amount of training data, and the variety of topics, produces an effect of data sparsity, in which there is little overlap between features from training and test data. We plan to repeat the experiments using semantically-richer features based on the use of language models, which should improve the vectorial representations by projecting onto similar vectorial representations the features with similar semantic properties, thus reducing the effect of data sparsity.

The participation to subtask E resulted in a bias correction method, ARC, that performed well. ARC sensibly improved on the baseline produced by the direct use of the original regression method, the one used to produce the submission for subtask C, without correction. Future work will explore the use of the bias correction method in combination with other ordinal regression methods, either based on classification or linear regression.

A strong assumption of the ARC method is that the error on each label has a linear relation with respect to the prevalence. This assumption can be considered to hold locally, i.e., when the variation of prevalence for a label across topics is limited, while it is harder to consider it valid when prevalences varies a lot across topics. Future work will explore the use of more complex models, e.g., fitting the differences observed between $p_j(c_i)$ and $\hat{p}_j(c_i)$ on the training set using a polynomial model, instead of a single $w_i$ weight.

## References


Fabio Aiolli, Riccardo Cardin, Fabrizio Sebastiani, and Alessandro Sperduti. 2009. Preferential text classification: Learning algorithms and evaluation measures. *Information retrieval*, 12(5):559–580.

Antonio Bella, Cesar Ferri, José Hernández-Orallo, and Maria Jose Ramirez-Quintana. 2010. Quantification via probability estimators. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 737–742. IEEE.

Wei Chu and S Sathiya Keerthi. 2007. Support vector ordinal regression. *Neural computation*, 19(3):792–815.

Lin Dong, Eibe Frank, and Stefan Kramer, 2005. *Knowledge Discovery in Databases: PKDD 2005: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005. Proceedings*, chapter Ensembles of Balanced Nested Dichotomies for Multi-class Problems, pages 84–95. Springer Berlin Heidelberg, Berlin, Heidelberg.

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

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.