

# TwISE at SemEval-2016 Task 4: Twitter Sentiment Classification

Georgios Balikas and Massih-Reza Amini

University of Grenoble-Alpes

firstname.lastname@imag.fr

## Abstract

This paper describes the participation of the team “TwISE” in the SemEval 2016 challenge. Specifically, we participated in Task 4, namely “Sentiment Analysis in Twitter” for which we implemented sentiment classification systems for subtasks A, B, C and D. Our approach consists of two steps. In the first step, we generate and validate diverse feature sets for twitter sentiment evaluation, inspired by the work of participants of previous editions of such challenges. In the second step, we focus on the optimization of the evaluation measures of the different subtasks. To this end, we examine different learning strategies by validating them on the data provided by the task organisers. For our final submissions we used an ensemble learning approach (stacked generalization) for Subtask A and single linear models for the rest of the subtasks. In the official leaderboard we were ranked 9/35, 8/19, 1/11 and 2/14 for subtasks A, B, C and D respectively. The code can be found at [https://github.com/balikasg/SemEval2016-Twitter\\_Sentiment\\_Evaluation](https://github.com/balikasg/SemEval2016-Twitter_Sentiment_Evaluation).

## 1 Introduction

During the last decade, short-text communication forms, such as Twitter microblogging, have been widely adopted and have become ubiquitous. Using such forms of communication, users share a variety of information. However, information concerning one’s sentiment on the world around her has attracted a lot of research interest (Nakov et al., 2016; Rosenthal et al., 2015).

Working with such short, informal text spans poses a number of different challenges to the Natural Language Processing (NLP) and Machine Learning (ML) communities. Those challenges arise from the vocabulary used (slang, abbreviations, emojis) (Maas et al., 2011), the short size, and the complex linguistic phenomena such as sarcasm (Rajadesingan et al., 2015) that often occur.

We present, here, our participation in Task 4 of SemEval 2016 (Nakov et al., 2016), namely Sentiment Analysis in Twitter. Task 4 comprised five different subtasks: Subtask A: Message Polarity Classification, Subtask B: Tweet classification according to a two-point scale, Subtask C: Tweet classification according to a five-point scale, Subtask D: Tweet quantification according to a two-point scale, and Subtask E: Tweet quantification according to a five-point scale. We participated in the first four subtasks under the team name “TwISE” (Twitter Sentiment Evaluation). Our work consists of two steps: the preprocessing and feature extraction step, where we implemented and tested different feature sets proposed by participants of the previous editions of SemEval challenges (Tang et al., 2014; Kiritchenko et al., 2014a), and the learning step, where we investigated and optimized the performance of different learning strategies for the SemEval subtasks. For Subtask A we submitted the output of a stacked generalization (Wolpert, 1992) ensemble learning approach using the probabilistic outputs of a set of linear models as base models, whereas for the rest of the subtasks we submitted the outputs of single models, such as Support Vector Machines and Logistic

Regression.<sup>1</sup>

The remainder of the paper is organised as follows: in Section 2 we describe the feature extraction and the feature transformations we used, in Section 3 we present the learning strategies we employed, in Section 4 we present a part of the in-house validation we performed to assess the models' performance, and finally, we conclude in Section 5 with remarks on our future work.

## 2 Feature Engineering

We present the details of the feature extraction and transformation mechanisms we used. Our approach is based on the traditional  $N$ -gram extraction and on the use of sentiment lexicons describing the sentiment polarity of unigrams and/or bigrams. For the data pre-processing, cleaning and tokenization<sup>2</sup> as well as for most of the learning steps, we used Python's Scikit-Learn (Pedregosa et al., 2011) and NLTK (Bird et al., 2009).

### 2.1 Feature Extraction

Similar to (Kiritchenko et al., 2014b) we extracted features based on the lexical content of each tweet and we also used sentiment-specific lexicons. The features extracted for each tweet include:

- $N$ -grams with  $N \in [1, 4]$ , character grams of dimension  $M \in [3, 5]$ ,
- # of exclamation marks, # of question marks, # of both exclamation and question marks,
- # of capitalized words and # of elongated words,
- # of negated contexts; negation also affected the  $N$ -grams features by transforming a word  $w$  in a negated context to  $w\_NEG$ ,
- # of positive emoticons, # of negative emoticons and a binary feature indicating if emoticons exist in a given tweet, and

<sup>1</sup>To enable replicability we make the code we used available at [https://github.com/balikasg/SemEval2016-Twitter\\_Sentiment\\_Evaluation](https://github.com/balikasg/SemEval2016-Twitter_Sentiment_Evaluation).

<sup>2</sup>We adapted the tokenizer provided at <http://sentiment.christopherpotts.net/tokenizing.html>

- Part-of-speech (POS) tags (Gimpel et al., 2011) and their occurrences partitioned regarding the positive and negative contexts.

With regard to the sentiment lexicons, we used:

- manual sentiment lexicons: the Bing Liu's lexicon (Hu and Liu, 2004), the NRC emotion lexicon (Mohammad and Turney, 2010), and the MPQA lexicon (Wilson et al., 2005),
- # of words in positive and negative context belonging to the word clusters provided by the CMU Twitter NLP tool<sup>3</sup>
- positional sentiment lexicons: sentiment 140 lexicon (Go et al., 2009) and the Hashtag Sentiment Lexicon (Kiritchenko et al., 2014b)

We make, here, more explicit the way we used the sentiment lexicons, using the Bing Liu's lexicon as an example. We treated the rest of the lexicons similarly. For each tweet, using the Bing Liu's lexicon we obtain a 104-dimensional vector. After tokenizing the tweet, we count how many words ( $i$ ) in positive/negative contexts belong to the positive/negative lexicons (4 features) and we repeat the process for the hashtags (4 features). To this point we have 8 features. We generate those 8 features for the lowercase words and the uppercase words. Finally, for each of the 24 POS tags the (Gimpel et al., 2011) tagger generates, we count how many words in positive/negative contexts belong to the positive/negative lexicon. As a result, this generates  $2 \times 8 + 24 \times 4 = 104$  features in total for each tweet.

For each tweet we also used the distributed representations provided by (Tang et al., 2014) using the min, max and average composition functions on the vector representations of the words of each tweet.

### 2.2 Feature Representation and Transformation

We describe the different representations of the extracted  $N$ -grams and character-grams we compared when optimizing our performance on each of the classification subtasks we participated. In the rest of this subsection we refer to both  $N$ -grams and

<sup>3</sup><http://www.cs.cmu.edu/~ark/TweetNLP/>

character-grams as words, in the general sense of letter strings. We evaluated two ways of representing such features: (i) a bag-of-words representation, that is for each tweet a sparse vector of dimension  $|V|$  is generated, where  $|V|$  is the vocabulary size, and (ii) a hashing function, that is a fast and space-efficient way of vectorizing features, i.e. turning arbitrary features into indices in a vector (Weinberger et al., 2009). We found that the performance using hashing representations was better. Hence, we opted for such representations and we tuned the size of the feature space for each subtask.

Concerning the transformation of the features of words, we compared the tf-idf weighing scheme and the  $\alpha$ -power transformation. The latter, transforms each vector  $x = (x_1, x_2, \dots, x_d)$  to  $x' = (x_1^\alpha, x_2^\alpha, \dots, x_d^\alpha)$  (Jegou et al., 2012). The main intuition behind the  $\alpha$ -power transformation is that it reduces the effect of the most common words. Note that this is also the rationale behind the *idf* weighting scheme. However, we obtained better results using the  $\alpha$ -power transformation. Hence, we tuned  $\alpha$  separately for each of the subtasks.

### 3 The Learning Step

Having the features extracted we experimented with several families of classifiers such as linear models, maximum-margin models, nearest neighbours approaches and trees. We evaluated their performance using the data provided by the organisers, which were already split in training, validation and testing parts. Table 1 shows information about the tweets we managed to download. From the early validation schemes, we found that the two most competitive models were Logistic Regression from the family of linear models, and Support Vector Machines (SVMs) from the family of maximum margin models. It is to be noted that this is in line with the previous research (Mohammad et al., 2013; Büchner and Stein, 2015).

#### 3.1 Subtask A

Subtask A concerns a multiclass classification problem, where the general polarity of tweets has to be classified in one among three classes: “Positive”, “Negative” and “Neutral”, each denoting the tweet’s overall polarity. The evaluation measure used for the

subtask is the Macro-F<sub>1</sub> measure, calculated only for the Positive and Negative classes (Nakov et al., 2016).

Inspired by the winning system of SemEval 2015 Task 10 (Büchner and Stein, 2015) we decided to employ an ensemble learning approach. Hence, our goal is twofold: (i) to generate a set of models that perform well as individual models, and (ii) to select a subset of models of (i) that generate diverse outputs and to combine them using an ensemble learning step that would result in lower generalization error.

We trained four such models as base models. Their details are presented in Table 2. In the stacked generalization approach we employed, we found that by training the second level classifier on the probabilistic outputs, instead of the predictions of the base models, yields better results. Logistic Regression generates probabilities as its outputs. In the case of SVMs, we transformed the confidence scores into probabilities using two methods, after adapting them to the multiclass setting: the Platt’s method (Platt and others, 1999) and the isotonic regression (Zadrozny and Elkan, 2002). To solve the optimization problems of SVMs we used the Liblinear solvers (Fan et al., 2008). For Logistic Regression we used either Liblinear or LBFSGS, with the latter being a limited memory quasi Newton method for general unconstrained optimization problems (Yu et al., 2011). To attack the multiclass problem, we selected among the traditional one-vs-rest approach, the crammer-singer approach for SVMs (Crammer and Singer, 2002), or the multinomial approach for Logistic Regression (also known as MaxEnt classifier), where the multinomial loss is minimised across the entire probability distribution (Malouf, 2002).

For each of the four base models the tweets are represented by the complete feature set described in Section 2. For transforming the n-grams and character-grams, the value of  $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1\}$ , the dimension of the space where the hashing function projects them, as well as the value of  $\lambda \in \{10^{-7}, \dots, 10^6\}$  that controls the importance of the regularization term in the SVM and Logistic regression optimization problems were selected by grid searching using 5-fold cross-validation on the training part of the provided data. We tuned each model independently before integrat-

	Train	Development	DevTest	Test
Subtask A	5,500	1,831	1,791	32,009
Subtask B & D	4,346	1,325	1,417	10,551
Subtask C & E	5,482	1,810	1,778	20,632

**Table 1:** Size of the data used for training and development purposes. We only relied on the SemEval 2016 datasets.

Algorithm	Multiclass approach	Optimizer	Probabilistic Outputs
SVMs	crammer-singer	Liblinear	isotonic regression
SVMs	crammer-singer	Liblinear	Platt’s
Logistic Regression	one-vs-all	Liblinear	native
Logistic Regression	multinomial loss function	LBFGS	native

**Table 2:** Description of the base learners used in the stacked generalization approach.

ing it in the stacked generalization.

Having the fine-tuned probability estimates for each of the instances of the test sets and for each of the base learners, we trained a second layer classifier using those fine-grained outputs. For this, we used SVMs, using the crammer-singer approach for the multi-class problem, which yielded the best performance in our validation schemes. Also, since the classification problem is unbalanced in the sense that the three classes are not equally represented in the training data, we assigned weights to make the problem balanced. Those weights were inversely proportional to the class frequencies in the input data for each class.

### 3.2 Subtask B

Subtask B is a binary classification problem where given a tweet known to be about a given topic, one has to classify whether the tweet conveys a positive or a negative sentiment towards the topic. The evaluation measure proposed by the organisers for this subtask is macro-averaged recall (MaR) over the positive and negative class.

Our participation is based on a single model. We used SVMs with a linear kernel and the Liblinear optimizer. We used the full feature set described in Section 2, after excluding the distributed embeddings because in our local validation experiments we found that they actually hurt the performance. Similarly to subtask A and due to the unbalanced nature of the problem, we use weights for the classes of the problem. Note that we do not consider the topic of the tweet and we classify the tweet’s overall polarity.

Hence, we ignore the case where the tweet consists of more than one parts, each expressing different polarities about different parts.

### 3.3 Subtask C

Subtask C concerns an ordinal classification problem. In the framework of this subtask, given a tweet known to be about a given topic, one has to estimate the sentiment conveyed by the tweet towards the topic on a five-point scale. Ordinal classification differs from standard multiclass classification in that the classes are ordered and the error takes into account this ordering so that not all mistakes weigh equally. In the tweet classification problem for instance, a classifier that would assign the class “1” to an instance belonging to class “2” will be less penalized compared to a classifier that will assign “-1” as the class . To this direction, the evaluation measure proposed by the organisers is the macroaveraged mean absolute error.

Similarly to Subtask B, we submitted the results of a single model and we classified the tweets according to their overall polarity ignoring the given topics. Instead of using one of the ordinal classification methods proposed in the bibliography, we use a standard multiclass approach. For that we use a Logistic Regression that minimizes the multinomial loss across the classes. Again, we use weights to cope with the unbalanced nature of our data. The distributed embeddings are excluded by the feature sets.

We elaborate here on our choice to use a conventional multiclass classification approach instead

of an ordinal one. We evaluated a selection of methods described in (Pedregosa-Izquierdo, 2015) and in (Gutiérrez et al., 2015). In both cases, the results achieved with the multiclass methods were marginally better and for simplicity we opted for the multiclass methods. We believe that this is due to the nature of the problem: the feature sets and especially the fine-grained sentiment lexicons manage to encode the sentiment direction efficiently. Hence, assigning a class of completely opposite sentiment can only happen due to a complex linguistic phenomenon such as sarcasm. In the latter case, both methods may fail equally.

### 3.4 Subtask D

Subtask D is a binary quantification problem. In particular, given a set of tweets known to be about a given topic, one has to estimate the distribution of the tweets across the Positive and Negative classes. For instance, having 100 tweets about the new iPhone, one must estimate the fractions of the Positive and Negative tweets respectively. The organisers proposed a smoothed version of the Kullback-Leibler Divergence (KLD) as the subtask’s evaluation measure.

We apply a classify and count approach for this task (Bella et al., 2010; Forman, 2008), that is we first classify each of the tweets and we then count the instances that belong to each class. To this end, we compare two approaches both trained on the features sets of Section 2 excluding the distributed representations: the standard multiclass SVM and a structure learning SVM that directly optimizes KLD (Esuli and Sebastiani, 2015). Again, our final submission uses the standard SVM with weights to cope with the imbalance problem as the model to classify the tweets. That is because the method of (Gao and Sebastiani, 2015) although competitive was outperformed in most of our local validation schemes.

## 4 The evaluation framework

Before reporting the scores we obtained, we elaborate on our validation strategy and the steps we used when tuning our models. In each of the subtasks we only used the data that were realised for the 2016 edition of the challenge. Our validation had the following steps:

	DevTest	Test	Rank
Subtask A	56.01	58.61	9/35
Subtask B	0.748	0.748	8/18
Subtask C	0.8121	0.72	1/11
Subtask D	0.00018	0.053	2/14

**Table 3:** The performance obtained on the “devtest” data and the SemEval 2016 Task 4 test data.

1. Training using the released training data,
2. validation on the validation data,
3. validation again, in the union of the devtest and training data (when applicable), after retraining on training and validation data.

For each parameter, we selected its value by averaging the optimal parameters with respect to the output of the above-listed steps (2) and (3). It is to be noted, that we strictly relied on the data released as part of the 2016 edition of SemEval; we didn’t use past data.

We now present the performance we achieved both in our local evaluation schemas and in the official results released by the challenge organisers. Table 3 presents the results we obtained in the “DevTest” part of the challenge dataset and the scores on the test data as they were released by the organisers. In the latter, we were ranked 9/35, 8/19, 1/11 and 2/14 for subtasks A, B, C and D respectively. Observe, that for subtasks A and B, using just the “devtest” part of the data as validation mechanism results in a quite accurate performance estimation.

## 5 Future Work

That was our first contact with the task of sentiment analysis and we achieved satisfactory results. We relied on features proposed in the framework of previous SemEval challenges and we investigated the performance of different classification algorithms.

In our future work we will investigate directions both in the feature engineering and in the algorithmic/learning part. Firstly, we aim to deal with tweets in a finer level of granularity. As discussed in Section 3, in each of the tasks we classified the overall polarity of the tweet, ignoring cases where the tweets were referring to two or more subjects. In the

same line, we plan to improve our mechanism for handling negation. We have used a simple mechanism where a negative context is defined as the group of words after a negative word until a punctuation symbol. However, our error analysis revealed that punctuation is rarely used in tweets. Finally, we plan to investigate ways to integrate more data in our approaches, since we only used this edition's data.

The application of an ensemble learning approach, is a promising direction towards the short text sentiment evaluation. To this direction, we hope that an extensive error analysis process will help us identify better classification systems that when integrated in our ensemble (of subtask A) will reduce the generalization error.

## Acknowledgments

We would like to thank the organisers of the Task 4 of SemEval 2016, for providing the data, the guidelines and the infrastructure. We would also like to thank the anonymous reviewers for their insightful comments.

## References

- [Bella et al.2010] 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.
- [Bird et al.2009] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.
- [Büchner and Stein2015] Matthias Hagen Martin Potthast Michel Büchner and Benno Stein. 2015. We-bis: An ensemble for twitter sentiment detection. *SemEval-2015*, page 582.
- [Crammer and Singer2002] Koby Crammer and Yoram Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292.
- [Esuli and Sebastiani2015] Andrea Esuli and Fabrizio Sebastiani. 2015. Optimizing text quantifiers for multivariate loss functions. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):27.
- [Fan et al.2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Lib-linear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.
- [Forman2008] George Forman. 2008. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.
- [Gao and Sebastiani2015] Wei Gao and Fabrizio Sebastiani. 2015. Tweet sentiment: From classification to quantification. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 97–104. ACM.
- [Gimpel et al.2011] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.
- [Go et al.2009] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1:12.
- [Gutiérrez et al.2015] P.A. Gutiérrez, M. Pérez-Ortiz, J. Sánchez-Monedero, F. Fernandez-Navarro, and C. Hervás-Martínez. 2015. Ordinal regression methods: survey and experimental study. *IEEE Transactions on Knowledge and Data Engineering*, Accepted.
- [Hu and Liu2004] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.
- [Jegou et al.2012] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. 2012. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, Sept.
- [Kiritchenko et al.2014a] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014a. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442. Association for Computational Linguistics and Dublin City University Dublin, Ireland.
- [Kiritchenko et al.2014b] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014b. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, pages 723–762.
- [Maas et al.2011] Andrew L Maas, Andrew Y Ng, and Christopher Potts. 2011. Multi-dimensional sentiment analysis with learned representations.
- [Malouf2002] Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics.

- [Mohammad and Turney2010] Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics.
- [Mohammad et al.2013] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.
- [Nakov et al.2016] 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 2016)*, San Diego, California, June. Association for Computational Linguistics.
- [Pedregosa et al.2011] 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.
- [Pedregosa-Izquierdo2015] Fabian Pedregosa-Izquierdo. 2015. *Feature extraction and supervised learning on fMRI: from practice to theory*. Ph.D. thesis, Université Pierre et Marie Curie.
- [Platt and others1999] John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- [Rajadesingan et al.2015] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 97–106. ACM.
- [Rosenthal et al.2015] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of SemEval-2015*.
- [Tang et al.2014] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565, Baltimore, Maryland, June. Association for Computational Linguistics.
- [Weinberger et al.2009] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM.
- [Wilson et al.2005] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.
- [Wolpert1992] David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.
- [Yu et al.2011] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin. 2011. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 85(1-2):41–75.
- [Zadrozny and Elkan2002] Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699. ACM.