A Case Study on using Crowdsourcing for Ambiguous Tasks

Ankush Chatterjee[1], Umang Gupta[2] and Puneet Agrawal[2]
[1]*Indian Institute of Technology, Kharagpur, India*
[2]*Microsoft, India*

Keywords: Crowdsourcing, Deep Learning, Label Aggregation Techniques.

Abstract: In our day to day life, we come across situations which are interpreted differently by different human beings. A given sentence may be offensive to some humans but not to others. Similarly, a sentence can convey different emotions to different human beings. For instance, *"Why you never text me!"*, can either be interpreted as a sad or an angry utterance. Lack of facial expressions and voice modulations make detecting emotions in textual sentences a hard problem. Some textual sentences are inherently ambiguous and their true emotion label is difficult to determine. In this paper, we study how to use crowdsourcing for an ambiguous task of determining emotion labels of textual sentences. Crowdsourcing has become one of the most popular medium for obtaining large scale labeled data for supervised learning tasks. However, for our task, due to the intrinsic ambiguity, human annotators differ in opinions about the underlying emotion of certain sentences. In our work, we harness the multiple perspectives of annotators for ambiguous sentences to improve the performance of an emotion detection model. In particular, we compare our technique against the popularly used technique of majority vote to determine the label of a given sentence. Our results indicate that considering diverse perspective of annotators is helpful for the ambiguous task of emotion detection.

1 INTRODUCTION

Emotions such as happiness, anger, sadness etc. are basic human traits that we experience everyday. In the field of cognitive computing, where we develop technologies to mimic the functioning of the human brain, understanding emotions is an important area of research (Thilmany, 2007). Emotions have been studied by researchers in the fields of psychology, sociology, medicine, computer science etc. for the past several years. Some of the prominent work in understanding and categorizing emotions include Ekman's six class categorization (Ekman, 1992) and Plutchik's "Wheel of Emotion" which suggested eight primary bipolar emotions (Plutchik and Kellerman, 1986). Given the vast nature of study in this field, there is naturally no broader consensus on the granularity of emotion classes.

In our work, we improve the emotion detection model, by using different strategies on consuming judgments of textual sentences. Crowdsourcing has become one of the most popular medium for obtaining large scale labeled data for supervised learning tasks. Since human annotators are prone to errors, multiple annotations for a given sentence are required to increase the accuracy of labels. Once the human annotation for a sentence is done, labels from all annotators are aggregated to produce a single aggregated label. One of the most effective strategy used for this aggregation is to output the class that receives the majority votes and discard other votes (Karger et al., 2011). However, these aggregation methods make an underlying assumption that there is a single correct label for a given sentence. In case of emotion labels, since a sentence can have an inherent ambiguity and a true emotion label cannot be determined for ambiguous scenarios, the assumption of a single correct label breaks. Table 1 provides examples where emotion of the given sentence is ambiguous. The diverse annotation from multiple annotators for such examples reinforces our belief that these data points are ambiguous and difficult to predict. In such scenarios, strategy of majority vote is expected to fail. In this paper, we explore the role of ambiguous data in training dataset and strategies to incorporate this ambiguous data for improving model performance.

In a nutshell, we explore various strategies and techniques to find answers to the following questions in our work:

- **Question 1 -** Should ambiguous data be a part of the training data? The presence of ambiguous data

Chatterjee, A., Gupta, U. and Agrawal, P.
A Case Study on using Crowdsourcing for Ambiguous Tasks.
DOI: 10.5220/0006955002420247
In *Proceedings of the 10th International Joint Conference on Computational Intelligence (IJCCI 2018)*, pages 242-247
ISBN: 978-989-758-327-8

Table 1: Examples of textual sentences with ambiguous emotions.

Sentence	Ambiguity of Emotion
Why you never text me!	Sad or Angry
I think I am going to cry.	Joy or Sadness

will affect the model's generalization?

- **Question 2 -** If ambiguous data is present in the training data, is majority voting the best way to consolidate multiple labels?

2 RELATED WORK

Any supervised algorithm requires labeled training data. Crowdsourcing has emerged as a popular mechanism to obtain labels at a large scale. However, a major problem faced in acquiring labels via crowdsourcing is that labels can be diverse and unreliable. Several researchers (Kazai et al., 2011; Steinhardt et al., 2016; Wang et al., 2014) have focussed on detecting "spammers", who are careless, submit random answers, and "adversaries", who may deliberately give wrong answers. A common strategy to improve reliability is to assign multiple annotators for each task and aggregate the workers' labels. The class of algorithms that infer true labels from multiple observations made by annotators can be put into the following two categories -

(a) *Discriminative Models* - Discriminative models do not model the observations from the annotators. Instead, they directly obtain true labels using aggregation schemes. Majority Voting, where the label receiving majority votes is chosen, is one such scheme (Karger et al., 2011). However, it is known to be error-prone, because it considers all the annotators equally skilled. In general, efficient aggregation methods should account for the differences in the workers' skills. The weighted majority voting that takes workers' reliability into consideration is an improvement over Majority Voting (Li and Yu, 2014).

(b) *Generative Models* - A generative method builds a flexible probabilistic model for generating noisy observations, conditioned on unknown true labels and some behavioral assumptions. Examples of such models being the Dawid-Skene (DS) estimator (Dawid and Skene, 1979), the minimax entropy (Entropy) estimator (Zhou et al., 2012; Zhou et al., 2014), and their variants. Early research characterized the annotators using confusion matrices, and inferred the labels using the EM algorithm (Raykar et al., 2010; Smyth et al., 1995). Recently more complicated generative models have been developed for this task (Whitehill et al., 2009; Welinder et al., 2010; Raykar and Yu, 2012; Wauthier and Jordan, 2011; Liu et al., 2012; Checco et al., 2017). However, all the above mentioned methods make an underlying assumption that there is a single correct label for a given data point. The assumption of a single correct label breaks when a sentence has an inherent ambiguity and a true emotion label cannot be determined for it. The diverse perspectives of annotators informs the model about ambiguity of a given data point and improves generalization of the model.

3 APPROACH AND EXPERIMENTS

In this section, we describe the experiments conducted to explore answers to the two research questions raised in Section 1.

3.1 Training Data Collection

To collect data for our experiments, conversational data from Twitter Firehose covering the four-year period from 2012 through 2015 is used. Tweets as well as replies to tweets are considered. This data is further preprocessed to remove twitter handles, hashtags etc. and serves as the base data for the data collection. The dataset D containing 61296 sentences belonging to 4 classes - Happy, Sad, Angry and Others is obtained. We consider the three primary emotions - Happy, Sad, Angry and group the other emotions under the Others category. Each sentence in this data is annotated by 7 annotators. Describing the details of data collection is out of scope of this paper due to space constraints. Readers can refer to (Gupta et al., 2017) where a similar data collection technique is used.

3.2 Certainty in Data

We experiment with datasets of varying degrees of certainty which is defined as follows:

Definition: *The Certainty p_i is the ratio of the number of annotators in favor of the majority class to the total number of annotators annotating the sentence.*

Hence, for a sentence S, if 4 annotators annotate it as Sad and 3 annotators annotate the same sentence as Angry, then the certainty of the label Sad is $4/7 = 0.57$.

To study effect of various degrees of certainty in data on model performance, we obtain subsets $D_{p_1}, D_{p_2}, ..., D_{p_k}$ of the dataset D, where D_{p_i} is the dataset consisting of only those samples whose labels have a minimum certainty of p_i. Here, $p_i < p_j$ for $i < j$, rendering D_{p_k} as the least ambiguous dataset

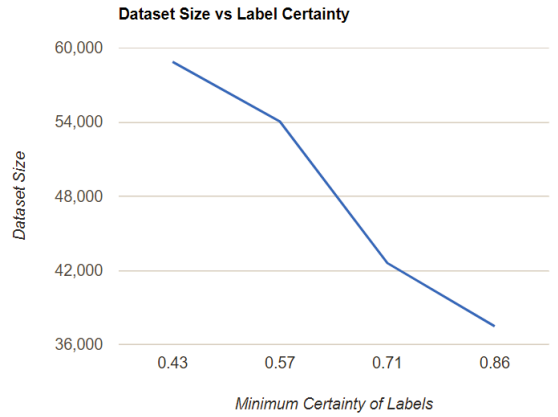

Figure 1: Size of dataset for different levels of Certainty.

of all. Since our original dataset was annotated by 7 annotators and a sentence could have the same label from 3 to 7 annotators to be marked as majority class, we obtain subsets $D_{0.43}, D_{0.57}, D_{0.71}, D_{0.86}$. As one can imagine, the size of the dataset reduces as certainty increases and the same behavior is observed in our dataset as well. Figure 1 shows the decrease in size of the dataset as the ambiguity of labels is reduced. We evaluate the performance of our model using these datasets with varying certainty as training data to explore answer to Question 1.

3.3 Majority Vote vs Certainty Representation

We introduce the approach of Certainty Representation (CR) and compare it with popularly used Majority Vote (MV). To understand the difference between the two approaches, consider an example sentence where 4 out of 7 annotators annotate it as Sad and 3 of them annotate the same as Angry. During the training process, considering the order of encoding as - Others, Happy, Sad, Anger, multi-class model for MV sees target vector as [0,0,1,0] whereas CR sees target vector as [0.0, 0.0, 0.57, 0.43].

We train one model with MV and two models with CR to explore answers to Question 2. The first model of CR trains in the same way as MV and the only difference is in the target vectors used for modeling the loss function. The target vector in majority class picks out one emotion class as the true emotion class, which may force the model to learn inconsistent representations for ambiguous sentence. We call this model CR-Uniform, since the training phase treats all input samples uniformly, irrespective of their ambiguity levels.

The 2^{nd} CR model is trained with the idea that noisy data hinders performance and hence sample with high certainty should be given more importance during the training process. Keeping this in mind, we calculate a Certainty-Score (CScore) for each training sample and multiply it with the loss function to account for the effects of noise. If the target vector is of the form $(t1, t2, t3, t4)$, we find out the highest and 2^{nd} highest values as Max1 and Max2, respectively. We then calculate Certainty-Score as

$$CScore = exp(Max1 * (Max1 - Max2)) - 1 \quad (1)$$

where $1/n \leq Max1 \leq 1.0$, n being the number of classes and $0.0 \leq Max2 \leq 0.5$
This returns a value close to 0 for highly ambiguous data and a value close to 1 for highly certain data. The CScore for some of the target vectors are shown in Table 2. When a highly ambiguous sample is encountered, its contribution to the loss function is reduced by a low CScore. On the other hand, a highly certain sample gets more importance in training because of a high CScore. Thus, the model learns from all the samples in the training data, but focuses less on the ambiguous data. We call this model CR-weighted.

Table 2: Comparison of CScore values for different possible target values. A more ambiguous target leads to a very low score, whereas a less ambiguous target produces a score close to 1.

Target	Max1*(Max1-Max2)	CScore
(0 .8 .1 .1)	0.56	0.751
(0 .8 .2 0)	0.48	0.616
(0 .6 .2 .2)	0.24	0.271
(0 .6 .4 0)	0.12	0.127
(.2 .4 .2 .2)	0.8	0.083

The MV and CR-Uniform models are rather simple and evaluated to analyze the change in performance with change in ambiguity levels in data. The CR-Weighted method is expected to leverage the advantage of multiple perspectives of annotators while not being drastically affected by noise.

3.4 Architecture of the Model

We approach the emotion understanding problem in a multi-class classification setting, where the model outputs probabilities of an input sentence belonging to either one of the four output classes - Happy, Sad, Angry and Others. The architecture used is the same as (Gupta et al., 2017) and can be seen in Figure 2. The model relies on combining Long Short Term Memory Networks, popularly known as LSTMs (Hochreiter and Schmidhuber, 1997), which are well-known for their sequence modeling abilities. The input user utterance is projected onto two different embedding spaces using two different embedding matrices. The two embeddings of a word are processed by two LSTM

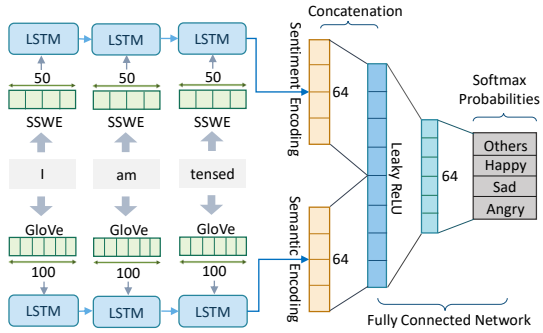

Figure 2: The architecture of Sentiment and Semantic Based Emotion Detector Model.

layers which learn semantic and sentiment feature representation. These two feature representations are then concatenated to form a sentence representation, which is fed as an input to a fully connected network with one hidden layer. The fully connected network models complex relations between these features and finally outputs probabilities of the input utterance belonging to each emotion class. The dimensions of each of the layer can be seen in Figure 2.

For each word in the input utterance, the authors experiment with multiple semantic word representations. They tried Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and FastText (Joulin et al., 2016). To get the sentiment representations, they considered Sentiment Specific Word Embedding (SSWE) (Tang et al., 2014). The authors train a simple LSTM model to test the effectiveness of each of the embeddings for emotion detection. Cross validation results indicate that GloVe gives the best macro F1 score, followed closely by SSWE. They found significant differences in the behavior of GloVe and SSWE; a few examples are in Table 3. *"Depression"* and *":'("*, both having a similar negative sentiment are very similar in SSWE embedding space, whereas GloVe gives a low score. For the *"happy"* and *"sad"* pair, GloVe doesn't differentiate much between the two whereas SSWE rightly gives a low score. However, semantically similar words like *"best"* and *"great"* have a high cosine similarity with GloVe as is expected, but SSWE gives a low score. Based on these observations GloVe and SSWE were chosen as embeddings for Semantic and Sentiment LSTM layer, respectively.

A dropout layer is used to avoid over-fitting. Microsoft Cognitive Toolkit[1] is used for training the model, Cross Entropy with Softmax is used as the loss function (LeCun et al., 2015), and Stochastic Gradient Descent (SGD) (Robbins and Monro, 1951) used as the learner.

[1]https://www.microsoft.com/en-us/cognitive-toolkit/

Table 3: Comparison of GloVe and SSWE embeddings w.r.t cosine similarity of word pairs.

Word1, Word2	GloVe	SSWE
depression, :'(	0.23	0.63
happy, sad	0.59	-0.42
best, great	0.78	0.15

4 RESULTS

In this section, we outline our experimental results by exploring answers to both Question 1 and Question 2

We use the average of F1 scores across emotion classes as the evaluation metric, with a higher F1 score indicating better classification. On observing Figure 3, we find answers to questions - Question 1 and Question 2 raised in the Introduction section. We notice that for MV and CR-Uniform model, performance increases as ambiguity decreases. We also notice that there is a dip in performance of all models at extremely low ambiguity. Also, the CR Models perform better than the MV Model for almost all levels of ambiguity. The CR-Weighted model performs the best at all levels of ambiguity and its performance is consistent across ambiguity levels. The detailed analysis of the observations is explained in Subsections 4.2 and 4.3.

4.1 Test Data

We use the Twitter Firehose and extract sentences from Twitter conversations using data from 2016. Our evaluation dataset is preprocessed by removing URLs, User IDs, hashtags etc. and consists of 2226 sentences along with their emotion class labels (Happy, Sad, Angry, Others) provided by human annotators. While getting labels for these sentences, annotators are shown two previous turns of conversation to provide context and get more accurate labels. Our models, however, do not take the previous context into account. This evaluation dataset was not seen at time of training and all approaches are evaluated on this dataset.

4.2 Observations for Question 1

- **Increase in Performance with Decrease in Ambiguity:** The performance of MV and CR-Uniform model increases slowly as ambiguity decreases. This observation aligns with intuition that lesser quantity of ambiguous sentences leads to lesser noisy data, which helps improve model performance. But the cleaner data comes at the

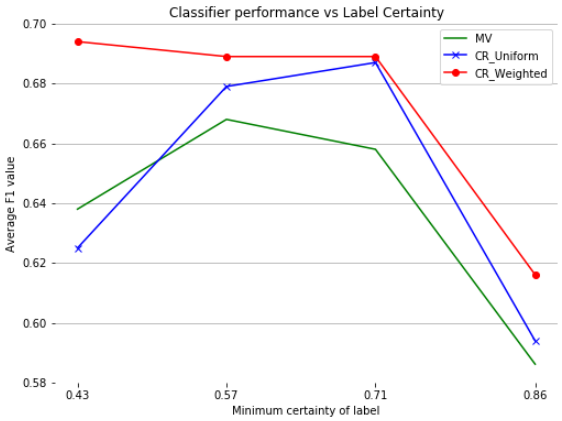

Figure 3: Comparison of Majority Voting and Certainty Representation methods for different levels of Certainty.

cost of lesser training data, which leads to a tipping point at minimum certainty level 0.71, where the model can learn most effectively.

- **Consistent Performance of the CR-Weighted Method:** The CR-Weighted model which was devised to eradicate the effect of noise, performs well even at high levels of ambiguity. We also note the performance is somewhat consistent across ambiguity levels. This is because the model focuses on the least ambiguous data in every dataset for training, while still learning from the more ambiguous examples.
- **Loss of Information at Extremely Low Ambiguity:** As ambiguity decreases, so does the number of samples in the training data. After a point, when there is almost no ambiguity in the dataset, the performance drops as the model is unaware of many sentences which might represent emotions in complex ways. This is true even for the CR-Weighted model as a lack of information containing training samples hinders its performance.

A highly non-ambiguous training data leads to loss of information and thus poor generalization towards unseen data. We conclude that although lesser ambiguity leads to better classification performance, it also leads to information loss and the presence of some ambiguous data in the training set helps model performance. We propose a novel way to deal with highly ambiguous samples in training data by accounting with a sample's contribution towards training in proportion to the ambiguity it contains. This method achieves somewhat consistent performance across ambiguity levels.

4.3 Observations for Question 2

- **CR Performs Better than MV:** The Certainty Representation Models perform better than the Majority Vote Model for almost all levels of ambiguity. This is due to the fact that CR Models are aware of the multiple perspectives of a sample whereas the MV Model ignores that information. MV model forces the model to draw inconsistent representations for ambiguous sentences.
- **Similar Performance at Low Ambiguity:** At high level of certainty, all three models perform almost equally at certainty level 0.86. This is due to the fact that at high level of certainty, there is hardly any ambiguity in the data and majority vote is a good indicator of the true emotion class of a sentence. In such a scenario, the CR Model can not derive any significant advantage from the multiple perspectives and tends to perform as well as the MV Model.

We thus conclude that Majority Voting, although a good aggregation function for most classification tasks, is not appropriate for the task of emotion detection. Due to the underlying complexity and ambiguity of emotions in text, alternative methods that consider the diverse perspectives of annotators should be used. We propose a novel method for training models with ambiguous emotion data and find its performance to be better than the Majority Vote method.

5 CONCLUSION

We solve the problem of emotion detection in text using Deep Learning algorithms, for which training data is annotated via crowdsourcing. However, some data points have an inherent ambiguity and their true emotion label is difficult to annotate, even with multiple annotators. We study how this ambiguous data should be considered in training a model. We notice that performance of our model increases with the decrease in ambiguity of labels in training data. We also notice that there is a dip in performance at extremely low ambiguity in training data labels. We propose the technique of Certainty Representation which takes into consideration the diverse perspectives of annotators and performs better than the Majority Vote. With a small hack in the training procedure (CR-Weighted), we can achieve consistently superior performance with the Certainty Representation approach. In future, we would like to understand how the presence of ambiguous data in the test dataset affects the evaluation performance of the model. We also intend to test performance of the model at various levels

of Certainty keeping the size of the training dataset constant.

REFERENCES

Checco, A., Roitero, A., Maddalena, E., Mizzaro, S., and Demartini, G. (2017). Lets agree to disagree: Fixing agreement measures for crowdsourcing. In *Proceedings of the Fifth AAAI Conference on Human Computation and Crowdsourcing (HCOMP-17)*, pages 11–20. AAAI Press.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28.

Ekman, P. (1992). An argument for basic emotions. *Cognition & emotion*, 6:169–200.

Gupta, U., Chatterjee, A., Srikanth, R., and Agrawal, P. (2017). A sentiment-and-semantics-based approach for emotion detection in textual conversations. *arXiv preprint arXiv:1707.06996*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–1780.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Karger, D. R., Oh, S., and Shah, D. (2011). Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems*, pages 1953–1961.

Kazai, G., Kamps, J., and Milic-Frayling, N. (2011). Worker types and personality traits in crowdsourcing relevance labels. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1941–1944. ACM.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Li, H. and Yu, B. (2014). Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*.

Liu, Q., Peng, J., and Ihler, A. T. (2012). Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 692–700.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Plutchik, R. and Kellerman, H. (1986). *Emotion: theory, research and experience*. Academic press New York.

Raykar, V. C. and Yu, S. (2012). Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13(Feb):491–518.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Smyth, P., Fayyad, U. M., Burl, M. C., Perona, P., and Baldi, P. (1995). Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems*, pages 1085–1092.

Steinhardt, J., Valiant, G., and Charikar, M. (2016). Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *Advances in Neural Information Processing Systems*, pages 4439–4447.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Thilmany, J. (2007). The emotional robot: Cognitive computing and the quest for artificial intelligence. *EMBO reports*, 8(11):992–994.

Wang, G., Wang, T., Zheng, H., and Zhao, B. Y. (2014). Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *USENIX Security Symposium*, pages 239–254.

Wauthier, F. L. and Jordan, M. I. (2011). Bayesian bias mitigation for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 1800–1808.

Welinder, P., Branson, S., Perona, P., and Belongie, S. J. (2010). The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432.

Whitehill, J., Wu, T.-f., Bergsma, J., Movellan, J. R., and Ruvolo, P. L. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, pages 2035–2043.

Zhou, D., Basu, S., Mao, Y., and Platt, J. C. (2012). Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203.

Zhou, D., Liu, Q., Platt, J., and Meek, C. (2014). Aggregating ordinal labels from crowds by minimax conditional entropy. In *International Conference on Machine Learning*, pages 262–270.