CLARK at SemEval-2019 Task 3: Exploring the Role of Context to Identify Emotion in a Short Conversation

Joseph R. Cummings
Northwestern University
jcummings@u.northwestern.edu

Jason R. Wilson
Northwestern University
jrw@northwestern.edu

Abstract

With text lacking valuable information available in other modalities, *context* may provide useful information to better detect emotions. In this paper, we do a systematic exploration of the role of context in recognizing emotion in a conversation. We use a Naïve Bayes model to show that inferring the *mood* of the conversation before classifying individual utterances leads to better performance. Additionally, we find that using context while training the model significantly decreases performance. Our approach has the additional benefit that its performance rivals a baseline LSTM model while requiring fewer resources.

1 Introduction

Recognizing affect (emotional content) in text has been an ongoing research challenge for roughly 20 years. While earlier work focused on larger bodies of text, like movie reviews for sentiment analysis (Pang et al., 2002) or classifying mood in blog posts (Mishne et al., 2005), more recent work has looked at small bodies of text, particularly text from social media. With smaller bodies of text inherently having less information, current efforts are investigating how context may supplement the information. However, it is not yet clear how best to incorporate context. To this end, we explore how mood and emotion from previous messages may be used to better recognize emotions.

Mood and emotion are generally regarded as two types of affect. Emotions are reactions and have a limited duration (Ortony et al., 1990; Schwarz and Clore, 2006). While emotions are dynamic and constantly changing, mood reflects a more persistent affect that can influence cognitive processes (Busemeyer et al., 2007), including how people recognize emotions (Schmid and Mast, 2010). For this work, we view mood as the affect present in the whole conversation and emotion as what is expressed in a given turn.

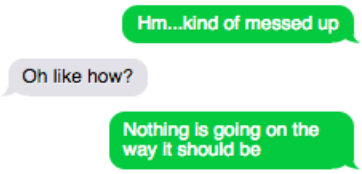

Figure 1: Example of a three turn conversation

Our goal is to take a short, online conversation (see Figure 1) and categorize the last utterance as happy, sad, angry, or others. In this paper, we present our model Conversational Lexical Affect Recognition Kit (CLARK), which is the result of a systematic exploration into how context may be used during the training and classification phases of a model to improve emotion recognition. To assess context we infer the mood of the conversation and the emotions of previous utterances. Although context would seem to be useful, providing additional information, we find that is only beneficial during classification. Conversely, including context while training the model leads to significantly degraded performance.

2 Related Work

There are several approaches to recognizing affect in a body of text. Many have used classification methods on a large body of text, such as movie reviews (Pang et al., 2002), blogs (Mishne et al., 2005; Mihalcea and Liu, 2006), and fairy tales (Alm et al., 2005; Mohammad, 2011), using techniques like SVMs (Pang et al., 2002; Mishne et al., 2005), Naïve Bayes (Mihalcea and Liu, 2006), HMMs (Ho and Cao, 2012), and Deep Learning (Zhang et al., 2018).

More recently, the rise of instant messaging and social media has led to greater interest in recognizing emotion in a smaller body of text. While lexicon based approaches were initially used for detecting emotions in smaller bodies of text (Thelwall et al., 2010; Staiano and Guerini, 2014), Deep Learning models dominate the recent

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 159–163
Minneapolis, Minnesota, USA, June 6–7, 2019.

work (Abdul-Mageed and Ungar, 2017; Chatterjee et al., 2019a).

Our approach is a blend of using a larger and smaller body of text. For the larger body, we detect the mood in a whole conversation. Additionally, we consider a smaller body of text, a single message in a conversation, and detect the emotion in that message. In contrast to many recent approaches using Deep Learning techniques, we use a Naïve Bayes model that requires less data and is trained faster while exhibiting no noticeable degradation in performance in comparison to a baseline SS-LSTM model.

3 Model

We model the task of detecting emotions as a multi-class classification problem. Given a user utterance, the model outputs probabilities of it belonging to the four output classes: happy, sad, angry, or others. Our approach uses CLARK, which at its base level, utilizes a Naïve Bayes model (McCallum and Nigam, 1998) with prior probabilities, which we take to be the frequency of tokens per class. To explore the role of context, we examine several variants of training and classification, detailed later. Keeping the feature set small, we use only unigram and bigrams. We also remove stop words and the following set of punctuation: period, dash, underscore, ampersand, tilde, comma, and backslash. To tokenize the tweets, we utilize Natural Language Toolkit's (NLTK) casual tokenize functionality, which places an emphasis on informal language and is able to pick up emoticons and collections of characters that are semantically equivalent to emoticons, e.g. ':)' is a smiley face.

3.1 Training

The model is trained on three turn conversations from Twitter with the last utterance classified according to the context of the first two utterances via semi-automated techniques (Chatterjee et al., 2019b). 30,160 conversations were provided for training and validation, consisting of 4,243 happy, 5,463 sad, 5,506 angry, and 14,947 others.

We test two variants for training the model, Conv, which we use to infer mood, and only Turn 3 (T3), to calculate feature probabilities given our set of four emotions. Conv consists of all words from the entire conversation, whereas T3 is the third and final utterance.

3.2 Classification

Our classification is a two step chaining process as shown in Algorithm 1. In the first step we find the initial probabilities for each class, denoted by the variable *post*. If we are using mood, denoted by the variable $Mood$, then this distribution is calculated using our model on $Conv$ (see line 7). Otherwise, it is set to the prior probabilities generated from the training (line 9). The resulting probabilities for each class are then used as the priors in step two.

In the second step, we classify the following combinations of individual turns in the conversation: {T3}, {T1, T3}, {T1, T2, T3}. Processing a combination consists of finding the posterior of the first turn and using it as the prior for the next turn and continuing until getting a final posterior, from which we take the highest probability class and return it as the final classification.

Algorithm 1 Classification sequence

```
1:  procedure ClassifyInContext
2:      Mood ← True ∨ False
3:      Conv ← words from current conversation
4:      default ← prior probabilities of all classes
5:      Turns ∈ {{T3}, {T1, T3}, {T1, T2, T3}}
6:      if Mood then                                   ▷ Step 1
7:          post = runModel(Conv, default)
8:      else
9:          post = default
10:     end if
11:     for each turn t do                             ▷ Step 2
12:         if t ∈ Turns then
13:             post = runModel(t, post)
14:         end if
15:     end for
16:     return argmax(post)
17: end procedure
```

4 Results

Our results show that inferring mood via Conv in the conversation before recognizing emotion in individual utterances yields improved performance. Furthermore, the best performances focus on the first user, utilizing only the first and third utterances in the second step of classification. We also see that in training the model, the best performance comes from limiting our set to just T3.

Results are organized by analysis on the internal model, followed by a comparison against a baseline Deep Learning model - the one provided by the EmoContext organizers. CLARK is tested with two parameters - classification method and training method. Our best results on the test set yielded a micro F_1 score of 0.5637, roughly equiv-

Classification Method	Acc.	Precision	Recall	F_1
Conv (C)	0.4256	0.2788	0.4170	0.3342
Turn 3 (T3)	0.2651	0.2211	0.4105	0.2874
C, T3	0.5575	0.4073	0.5571	0.4705
T1, T2, T3	0.5308	0.3730	0.5120	0.4316
C, T1, T2, T3	0.5358	0.3823	0.5276	0.4433
T1, T3	0.5369	0.3874	0.5235	0.4431
C, T1, T3	0.5732	0.41915	0.5684	0.4825

Table 1: Results from varying parameter classification method and training on the whole conversation.

Classification Method	Acc.	Precision	Recall	F_1
Conv (C)	0.7125	0.7283	0.5366	0.6178
Turn 3 (T3)	0.7155	0.6366	0.7456	0.6867
C, T3	0.8131	0.7875	0.7766	0.782
T1, T2, T3	0.7997	0.7745	0.7516	0.7629
C, T1, T2, T3	0.7979	0.769	0.7573	0.7631
T1, T3	0.8168	0.7921	0.7778	0.7848
C, T1, T3	**0.8169**	**0.7848**	**0.7892**	**0.7870**

Table 2: Results from varying parameter classification method and training on only T3.

alent to the model from the EmoContext organizers. This score is considerably lower than we got from our evaluations on the training data, possibly attributed to the quality of the labels for the training set versus the test set. We chose not to focus our analysis on the test set because we are not able to do a deep analysis as a result of the data not being readily available at the time. The remaining results we discuss are obtained using a 10-fold cross validation on the training set. Tables 1 and 2 show results from using the 30,160 conversations.

From the difference in results between these tables, it is clear that the biggest improvements in the model comes from training on only T3 as opposed to Conv. The difference in the average micro F_1 score (Rijsbergen, 1979) of training on T3 and the average F_1 score of training on Conv is determined to be statistically significant ($p < 0.005$) using a t-test (Kim, 2015).

Within the classification method, we see that using T1 in coordination with T3 provides F_1 scores at least 14% higher than just classifying with T3. In addition, using Conv consistently provides better performances, albeit close. Thus, the best model is one which utilizes a classification combination of Conv, T1, and T3, with a micro F_1 score of 0.7870.

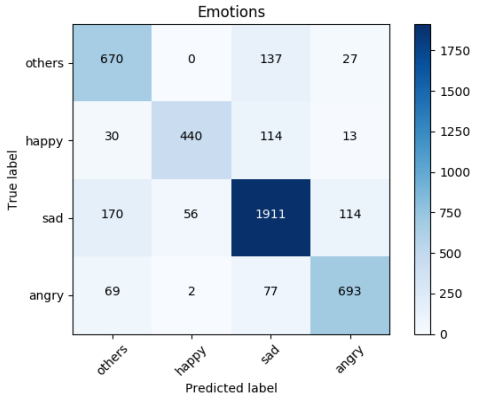

Figure 2: CLARK Confusion Matrix on a 9:1 ratio training to testing split.

As shown in Figure 2, CLARK is incredibly precise in the classification of the sad class and does moderately well in others and angry. However, others and sad are commonly predicted even when not the correct class predicted leading to lower recall scores.

A few concrete examples of these strengths and weaknesses are shown in Table 3. Because CLARK does not place weight on the specific T2 utterance, we see that in no. 2, we miss the positive emotion and misclassify the conversation as others. However, this is largely an anomaly - in fact, we see that T2 usually involves an interrogative or can be associated with a tangential class.

To demonstrate the efficiency of CLARK, we compare it to the baseline SS-LSTM model provided by the SemEval-2019 Task 3 organizers (Chatterjee et al., 2019a) as shown in Table 4. We compare both on time to train and quantity of data needed to produce certain performance. Time to train is normalized to CLARK. The SS-LSTM model performs at 0.6796 when trained with 1 Epoch (used as the very minimum required for a neural model) and takes 26 times longer than CLARK. For 3 Epochs, it performs at 0.7832 and takes 70 times longer than CLARK.

We also examine the effect the size of the training dataset has on the performance of each model, as shown in Figure 3. CLARK vastly outperforms the SS-LSTM models with minimal data. The SS-LSTM (3) model takes the full 30,160 data point

#	T1 (User 1)	T2 (User 2)	T3 (User 1)	True Label	Predicted Label
1	Same to you	:)...hws life going?	Bad	Sad	Sad
2	You are funny	Why, thank you!	So start a fun conversation	Happy	Others
4	Yes	annoys me so much	I really love it	Happy	Happy

Table 3: Results from adjusting classification method and training on only T3.

dataset to achieve an equivalent F_1 score.

Model	Time	Micro F_1
CLARK	1	0.7870
SS-LSTM(1)	26	0.6796
SS-LSTM(3)	70	0.7834

Table 4: Comparison between CLARK and baseline Deep Learning models in terms of normalized time to train and classification performance.

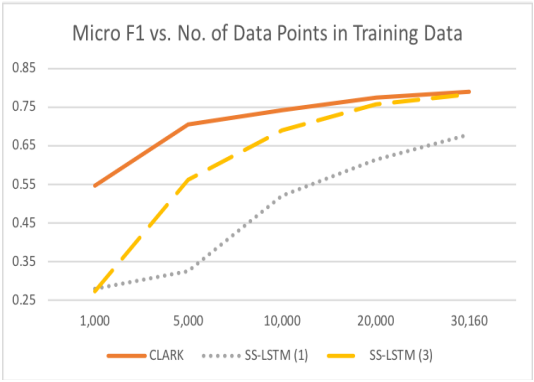

Figure 3: Comparisons of CLARK, SS-LSTM(1), and SS-LSTM(3) at varying amount of training data.

5 Discussion

We investigated a way to model emotion from text in the context of a conversation, instead of a single utterance. In doing so, we analyzed the performance of two different types of models, one based on a Naïve Bayes approach, which we call CLARK, and one on a Deep Learning approach. CLARK trained on T3 and classified using {Conv, T1, T3} leads to the best performance.

One way to utilize context is during training, but our results in experiments with CLARK show that the including more context (i.e., the whole conversation) significantly degrades performance. Training just on T3 produces much better results than training on Conv. This makes sense as T3 is the utterance directly associated with the assigned label and as such, represents the words that we can associate to the label with the highest confidence.

Some notion of "context" is important in determining the overall emotion of a conversation. When classification uses Conv and the final utterance (T3), the model produces the best results, as demonstrated by consistently producing a better F_1 score. This reflects the idea that as humans, mood affects how we judge the emotion a person is currently expressing (Schmid and Mast, 2010).

Our approach to incorporating context is fundamentally different from the approach taken in the baseline model. The SS-LSTM is more similar to a training method using all three utterances and classification method using {T1, T2, T3}. It also takes exponentially longer to train than CLARK and produces roughly equivalent performance, when examining the full dataset. Any attempt to speed up the model by using less training data would be in vain as shown in Figure 3. In cases where efficiency is paramount, the Deep Learning approach is lacking because of these requirements. Being able to produce good results with less training data can be a valuable asset.

Many of this work's limitations come from the data and the way the data was processed. The set of 30,160 three turn conversations is not balanced - there is far more in the others class than the rest. Because Naïve Bayes is a probabilistic model, it will prefer the others class. A solution could be to utilize a Complement Naïve Bayes, which estimates parameters from data using complement classes (Rennie et al., 2003). In addition, the data was labelled using a semi-automatic technique. Human subjects labeled a small subset of tweets, and key word embeddings were then extrapolated to label the rest of the conversations. This method leaves a lot of room for error and even suggests the function our model is trying to learn is this labelling mechanism. In future work, we will use only data labelled by human subjects.

6 Conclusion

Context plays an important role in recognizing emotions, but blindly including context can actually make recognizing emotions more difficult. As a response to the SemEval-2019 Task 3 challenge, we performed a systematic exploration of how to use context in classifying emotions in a short conversation. The resulting model, CLARK, performs best when trained on just the third turn of conversations (no context) and then classification uses Conv to infer mood and emotions from previous turns (with context). The relatively simple Naïve Bayes model, which performs on par with a baseline LSTM model while requiring less data and time to train, demonstrates one successful approach to using context that is usable in resource-constrained scenarios. Furthermore, we believe that while our results are demonstrated using a Naive Bayes model, our approach to using context only when classifying has the potential of being applicable to other classification approaches.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. Emonet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 718–728.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics.

Jerome R. Busemeyer, Eric Dimperio, and Ryan K. Jessup. 2007. Integrating emotional processes into decision-making models. In *Integrated models of cognitive systems.*

Ankush Chatterjee, Umang Gupta, Manoj Kumar Chinnakotla, Radhakrishnan Srikanth, Michel Galley, and Puneet Agrawal. 2019a. Understanding emotions in text using deep learning and big data. *Computers in Human Behavior*, 93:309–317.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019b. Semeval-2019 task 3: Emocontext: Contextual emotion detection in text. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval-2019)*, Minneapolis, Minnesota.

Dung T. Ho and Tru H. Cao. 2012. A high-order hidden markov model for emotion detection from textual data. In *Knowledge Management and Acquisition for Intelligent Systems*, pages 94–105, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tae Kyun Kim. 2015. T test as a parametric statistic. In *Korean journal of anesthesiology*.

A. McCallum and K. Nigam. 1998. A comparison of event models for naive bayes text classification. In *Proceedings in Workshop on Learning for Text Categorization*, AAAI98, pages 41–48.

Rada Mihalcea and Hugo Liu. 2006. A corpus-based approach to finding happiness. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 139–144.

Gilad Mishne et al. 2005. Experiments with mood classification in blog posts. In *Proceedings of ACM SIGIR 2005 workshop on stylistic analysis of text for information access*, volume 19, pages 321–327.

Saif Mohammad. 2011. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 105–114. Association for Computational Linguistics.

Andrew Ortony, Gerald L Clore, and Allan Collins. 1990. *The cognitive structure of emotions*. Cambridge university press.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. 2003. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pages 616–623. AAAI Press.

C. J. Van Rijsbergen. 1979. *Information Retrieval*, 2nd edition. Butterworth-Heinemann, Newton, MA, USA.

Petra Claudia Schmid and Marianne Schmid Mast. 2010. Mood effects on emotion recognition. *Motivation and Emotion*, 34(3):288–292.

Norbert Schwarz and Gerald L. Clore. 2006. Feelings and phenomenal experiences. In A. Kruglansk and E. T. Higgins, editors, *Social psychology: Handbook of basic principles*, 2nd edition, pages 385–407. Guilford, New York.

Jacopo Staiano and Marco Guerini. 2014. Depechemood: a lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605*.

Mike Thelwall, Kevan Buckley, Georgios Paltoglou, Di Cai, and Arvid Kappas. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.