In Proceedings of NAACL HLT 2018

Human Needs Categorization of Affective Events Using Labeled and Unlabeled Data

Haibo Ding
School of Computing
University of Utah
Salt Lake City, UT 84112
hbding@cs.utah.edu

Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Abstract

We often talk about events that impact us positively or negatively. For example *"I got a job"* is good news, but *"I lost my job"* is bad news. When we discuss an event, we not only understand its affective polarity but also the reason *why* the event is beneficial or detrimental. For example, getting or losing a job has affective polarity primarily because it impacts us financially. Our work aims to categorize affective events based upon *human need* categories that often explain people's motivations and desires: PHYSIOLOGICAL, HEALTH, LEISURE, SOCIAL, FINANCIAL, COGNITION, and FREEDOM. We create classification models based on event expressions as well as models that use contexts surrounding event mentions. We also design a co-training model that learns from unlabeled data by simultaneously training event expression and event context classifiers in an iterative learning process. Our results show that co-training performs well, producing substantially better results than the individual classifiers.

1 Introduction

Recent research has focused on identifying *affective events* in text, which are activities or states that positively or negatively affect the people who experience them. Recognizing affective events in text is challenging because they appear as factual expressions and their affective polarity is often implicit. For example, *"I broke my arm"* and *"I got fired"* are usually negative experiences, while *"I broke a record"* and *"I went to a concert"* are typically positive experiences. Several NLP techniques have been developed to recognize affective events, including patient polarity verb bootstrapping (Goyal et al., 2010, 2013), implicature rules (Deng and Wiebe, 2014), label propagation (Ding and Riloff, 2016), pattern-based learning (Vu et al., 2014; Reed et al., 2017), and semantic consistency optimization (Ding and Riloff, 2018).

Our research aims to probe deeper and understand not just the polarity of affective events, but *the reason for* the polarity. Events can impact people in many ways, and understanding *why* an event is beneficial or detrimental is a fundamental aspect of language understanding and narrative text comprehension. Additionally, many applications could benefit from understanding the nature of affective events, including text summarization, conversational dialogue processing, and mental health therapy or counseling systems. As an illustration, a mental health therapy system can benefit from understanding why someone is in a negative state. If the triggering event for depression is *"I broke my leg"* then the reason is about the person's Health, but if the triggering event is *"I broke up with my girlfriend"* then the reason is based on Social relationships.

We hypothesize that the polarity of affective events can often be attributed to a relatively small set of *human need* categories. Our work is motivated by theories in psychology that explain people's motivations, desires, and overall well-being in terms of categories associated with basic human needs, such as Maslow's Hierarchy of Needs (Maslow et al., 1970) and Fundamental Human Needs (Max-Neef et al., 1991). Drawing upon these works, we propose that the polarity of affective events often arises from 7 types of human needs: PHYSIOLOGICAL, HEALTH, LEISURE, SOCIAL, FINANCIAL, COGNITION, and FREEDOM. For example, *"I broke my arm"* has negative polarity because it negatively impacts one's Health, *"I got fired"* is negative because it negatively impacts one's Finances, and *"I am confused"* is negative because it reflects a problem related to Cognition.

We explore this hypothesis and tackle the chal-

lenge of categorizing affective events in text with respect to these 7 human need categories. As our evaluation data, we use events extracted from personal blog posts and manually labeled with affective polarity in previous work (Ding and Riloff, 2018). These affective events were then subsequently annotated for the human need categories.

In this paper, we design several types of classification models that learn from both labeled and unlabeled data. First, we present supervised learning models that use lexical and embedding features for the words in event expressions, as well as models that learn from the sentence contexts surrounding mentions of event expressions. Next, we explore self-training and co-training models that exploit both labeled and unlabeled data for training. The most effective system is a co-training model that uses two classifiers with two different views in an iterative learning process: one classifier only uses the words in an event expression, and the other classifier only uses the contexts surrounding instances of an event expression. Our results show that this co-training model effectively uses unlabeled data to substantially improve results compared to classifiers trained only with labeled data, yielding gains in both precision and recall.

2 Related Work

Recently, there has been growing interest in recognizing the affective polarity of events. For example, Goyal et al. (2013) developed a bootstrapped learning method to learn *patient polarity verbs*, which impart affective polarities to their patients. Li et al. (2015) designed methods to extract verb expressions that imply negative opinions from reviews. Rashkin et al. (2016) recently proposed connotation frames to incorporate the connotative polarities for a verb's arguments from the writer's and other event entities' perspectives. Li et al. (2014) proposed a bootstrapping approach to extract major life events from tweets using congratulation and condolence speech acts. Most of these major life events are affective although their work did not identify polarity. Another group of researchers have studied +/- effect events (Deng et al., 2013; Choi and Wiebe, 2014) which they previously called benefactive/malefactive events. Their work mainly focused on inferring implicit opinions through implicature rules (Deng and Wiebe, 2014, 2015).

Ding and Riloff (2016) designed an event context graph model to identify affective events using label propagation. Reed et al. (2017) demonstrated that automatically acquired patterns could benefit the recognition of first-person related affective sentences. Most recently, Ding and Riloff (2018) developed a semantic consistency model to induce a large set of affective events using three types of semantic relations in an optimization framework. (We use their annotated affective event data set in our work.) All of this previous work only identifies affective events and their polarities. In contrast, our work aims to identify the reason for the affective polarity of an event.

The human need categories are inspired by two prior theories. The first one is Maslow's Hierarchy of Needs (Maslow et al., 1970) which was developed to study people's motivations and personalities. The second one is Fundamental Human Needs (Max-Neef et al., 1991) which was developed to help communities identify their strengths and weaknesses. The human need categories are also related to the concept of "goals", which has been proposed by (Schank and Abelson, 1977) to understand narrative stories. Goals could be very specific to a character in a particular narrative story. However, but many types of goals originate from universal needs and desires shared by most people (Max-Neef et al., 1991). In addition, our work is also related to research on wish detection (Goldberg et al., 2009), desire fulfillment (Chaturvedi et al., 2016), and modelling protagonist goals and desires (Rahimtoroghi et al., 2017).

Self-training is a semi-supervised learning method to improve performance by exploiting unlabeled data. Self-training has been successfully used in many NLP applications such as information extraction (Ding and Riloff, 2015) and syntactic parsing (McClosky et al., 2006). Co-training (Blum and Mitchell, 1998) uses both labeled and unlabeled data to train models that have two different views of the data. Co-training has been previously used for many NLP tasks including spectral clustering (Kumar and Daumé, 2011), word sense disambiguation (Mihalcea, 2004), coreference resolution (Phillips and Riloff, 2002), and sentiment analysis (Wan, 2009; Xia et al., 2015).

3 Affective Event Data

The goal of our research is to categorize affective events based on 7 categories of human needs. To facilitate this work, we build upon a large data set

Physiological	Health	Leisure	Social	Finance	Cognition	Freedom	Emotion	None
19 (4%)	52 (10%)	75 (14%)	108 (20%)	29 (5%)	26 (5%)	7 (1%)	128 (24%)	98 (18%)

Table 1: Distribution of Human Need Categories (each cell shows the frequency and percentage).

created for prior research (Ding and Riloff, 2018) which aims to identify affective events. We will refer to this data as the AffectEvent dataset. We will briefly describe this data and the human need category annotations that we added on top of it.

The AffectEvent dataset contains events extracted from a personal story corpus that was created by applying a personal story classifier (Gordon and Swanson, 2009) to 177 million blog posts. The personal story corpus contains 1,383,425 personal story blogs. StanfordCoreNLP (Manning et al., 2014) was used for POS and NER tagging and SyntaxNet (Andor et al., 2016) for parsing. Each event is represented using a frame-like structure to capture the meanings of different types of events. Each event representation contains four components: ⟨**Agent**, **Predicate**, **Theme**, **PP**⟩. The Predicate is a simple verb phrase corresponding to an action or state. The Agent is a named entity, nominal, or pronoun, and is extracted using syntactic heuristics rather than semantic role labeling. We use "Theme" loosely to allow a NP or adjective to fill this role. The PP component is composed of a preposition and a NP. All words in the event are lemmatized, and active and passive voices are normalized to have the same representation. See (Ding and Riloff, 2018) for more details of the event representation. Table 2 shows some examples of extracted events.

Positive Events	**Human Need**
⟨ our pizza; arrive, -, -⟩	Physiological
⟨ ear, be, better, - ⟩	Health
⟨ I, watch, Hellboy II, -⟩	Leisure
⟨ we, get, marry, -⟩	Social
⟨ I, get, my new laptop, -⟩	Finance
⟨ my memory, be, vivid, -⟩	Cognition
⟨ my heart, feel, happy, -⟩	Emotion
⟨ we, be, legal, -⟩	None
Negative Events	**Human Need**
⟨ I, grow, hungry, - ⟩	Physiological
⟨ my face, look, pale, - ⟩	Health
⟨ -, rain out, game, -⟩	Leisure
⟨ you, confront, me, -⟩	Social
⟨ I, be, unemployed, at time ⟩	Finance
⟨ my memory, not serve, me, -⟩	Cognition
⟨ I, be, scared, -⟩	Emotion
⟨ it, not work, -, for me⟩	None

Table 2: Examples of Affective Events with Human Need Category Labels

3.1 Human Need Category Annotations

Affective events impact people in a positive or negative way for a variety of reasons. We hypothesized that the polarity of most affective events arises from the satisfaction or violation of basic human needs. Psychologists have developed theories that explain people's motivations, desires, and overall well-being in terms of categories associated with basic human needs, such as Maslow's Hierarchy of Needs (Maslow et al., 1970) and Fundamental Human Needs (Max-Neef et al., 1991). Based upon this work, we defined 7 human need categories, which are briefly described below.

Physiological Needs maintain our body's basic functions (e.g., air, food, water, sleep). *Health Needs* are to be physically healthy and safe. *Leisure Needs* are to have fun, to be relaxed, to have leisure time, to appreciate and enjoy beauty. *Social Needs* are to have good social relations (e.g., family, friendship), to have good self-worth and self-esteem, and to be respected by others. *Financial Needs* are to obtain and protect financial income, to acquire and maintain valuable possessions, to have a job and satisfying work. *Cognition Needs* are to obtain skills, information, and knowledge, to receive education, to improve one's intelligence, and to mentally process information correctly. *Freedom Needs* are the ability to move or change positions freely, and to access things or services in a timely manner. We also defined two categories for event expressions that represent explicit emotions and opinions (*Emotions/Sentiments/Opinions*) and events that do not fall into any other categories (*None of the Above*).

We added manual annotations for human need categories on top of the manually annotated positive and negative affective events in the AffectEvent dataset. Three people were asked to assign a human need category label to each of the 559 affective events in the AffectEvent test set. Annotators achieved good pairwise inter-annotator agreement ($\kappa \geq .65$) on this task. The Cohen's kappa scores were κ=.69, κ=.66 and κ=.65. We assigned a single category to each event because most of the affective events fell into just one category in our preliminary study, even though some cases could legitimately be argued

for multiple categories. We discuss this issue further in Section 5.4

The distribution of human need categories is shown in Table 1. Since very few affective events were found to belong to the *Freedom* category, this category was merged into None. Additionally, 17 events received three different labels from the annotators, so they were discarded. The majority label was then assigned to the remaining events, yielding a gold standard data set of 542 affective events with human need category labels. Some of the annotated examples are shown in Table 2. A more detailed description of the human need category definitions, data set, and manual annotation effort is described in (Ding et al., 2018). This data set is freely available for other researchers to use.

In the next section, we present classification models designed to tackle this human needs categorization task.

4 Categorizing Human Needs with Labeled and Unlabeled Data

Automatically categorizing affective events in text based on human needs is a new task, so we investigated several types of approaches. First, we designed supervised classifiers to categorize affective events based upon the words in the event expressions, which we will refer to as *Event Expression Classifiers*. We explored lexical features, word embedding features, and semantic category features, along with several types of machine learning algorithms.

Our task is to determine the human need category of an affective event based on the meaning of the event itself, independent of any specific context.[1] But we hypothesized that collecting the contexts around instances of the events could also provide valuable information to infer human need categories. So we also designed *Event Context Classifiers* to use the sentence contexts around event mentions as features.

Our gold standard data set is relatively small, so supervised learning that relies entirely on manually labeled data may not have sufficient coverage to perform well across the human need categories. However, the AffectEvent dataset contains a very large set of events that were extracted from the same blog corpus, but not manually labeled with affective polarity. Consequently, we explored two weakly supervised learning methods to exploit this large set of unlabeled events. First, we tried self-training to iteratively improve the event expression classifier. Second, we designed a co-training model that takes advantage of both an event expression classifier and an event context classifier to learn from the unlabeled events. These two types of classifiers provide complementary views of an event, so new instances labeled by one classifier can be used as valuable new data to benefit the other classifier, in an iterative learning cycle.

4.1 Event Expression Classifiers

The most obvious approach is to use the words in event expressions as features for recognizing human need categories (e.g., {ear, be, better} for the event <ear, be, better>). We experimented with both lexical (string) features and pre-trained word embedding features. For the latter, we used GloVe (Pennington et al., 2014) vectors (200d) pretrained on 27B tweets. For each event expression, we compute its embedding as the average of its words' embeddings.

We also designed semantic features using the lexical categories in the LIWC lexicon (Pennebaker et al., 2007) to capture a more general meaning for each word. LIWC is a dictionary of words associated with "psychologically meaningful" lexical categories, some of which are directly relevant to our task, such as AFFECTIVE, SOCIAL, COGNITIVE, and BIOLOGICAL PROCESS. We identify the LIWC category of the head word of each phrase in the event representation and use them as *Semantic Category* features.

We experimented with three types of supervised classification models: logistic regression (LR), support vector machines (SVM), and recurrent neural network classifiers (RNN). One advantage of the RNN is that it considers the word order in the event expression, which can be important. In our experiments, we used the Scikit-learn implementation (Pedregosa et al., 2011) for the LR classifier, and LIBSVM (Chang and Lin, 2011) with a linear kernel for the SVM classifier. For the RNN, we used the example LSTM implementation from Keras (Chollet et al., 2015) github, which was developed to build a sentiment classifier. We used the default parameters in our experiments[2].

[1] We view this as assuming the most common interpretation of an event, which would be the default in the absence of context.

[2] LR and SVM use the one-vs-rest (ovr) scheme, while RNN is a single multi-class classifier.

4.2 Event Context Classifiers

The event dataset was originally extracted from a large collection of blog posts, which contain many instances of the events in different sentences. We hypothesized that the contexts surrounding instances of an event can also provide strong clues about the human need category associated with the event. Therefore, we also created *Event Context Classifiers* to exploit the sentence contexts around event mentions. We explored several designs for event context classifiers, which are explained below.

Context$^{\text{SentBOW}}$: For each event in the training set, we first collect all sentences mentioning this event and assign the event's human need category as the label for each sentence. Each sentence is then used as a training instance for the event context classifier. We use a bag-of-words representation for each sentence.

Context$^{\text{SentEmbed}}$: This variation labels sentences exactly the same way as the previous model. But each sentence is represented as a dense embedding vector, which is computed as the average of the embeddings for each word in the sentence. We used GloVe (Pennington et al., 2014) vectors (200d) pretrained on 27B tweets.

Context$^{\text{AllBOW}}$: Instead of treating each sentence as a training instance, for this model we aggregate all of the sentences that mention the same event to create one giant context for the event. Each event corresponds to one training instance in this model, which is represented using bag-of-word features.

Context$^{\text{AllEmbed}}$: This variation aggregates the sentences that mention an event exactly like the previous model. But each sentence is represented as a dense embedding vector. First, we compute an embedding vector for each sentence as the average of the embeddings of its words. Then we compute a single context embedding by averaging all of the sentence embeddings.

In the data, some events appear in many sentences, while others appear in just a few sentences. To maintain balance, we randomly sample 10 sentences for each event to use as its contexts.

To predict the human need category of an event, we first apply the event context classifier to contexts that mention the event, which produces a probability distribution over the human need categories. For each category, we compute its mean probability. Finally, we assign the event with the human need category that has the highest mean probability (i.e. argmax).

4.3 Self-Training the Event Expression Classifier

Our labeled data set is relatively small, but as mentioned previously, the AffectEvent dataset contains a large set of unlabeled events as well. So we designed a self-training model to try to iteratively improve the event expression classifier by exploiting the unlabeled event data.

The self-training process works as follows. Initially, the event expression classifier is trained using the manually labeled events. Then the classifier is applied to the unlabeled events and assigns a human need category to each event with a confidence value. For each human need category, we select the unlabeled event that has been assigned to that category with the highest confidence. Therefore, each category will have one additional labeled event at each iteration. The newly labeled events are added to the labeled data set, and the classifier is re-trained for the next iteration.

4.4 Co-Training with Event Expression and Event Context Classifiers

The sentence contexts in which an event appears contain complementary information to the event expression itself. So we designed co-training models to exploit these complementary types of classifiers to iteratively learn from unlabeled data.

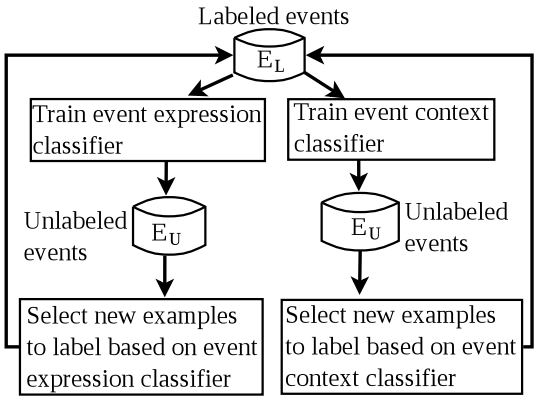

Figure 1: The Co-Training Model

Figure 1 shows the architecture of our co-training model. Initially, an event expression classifier and an event context classifier are independently trained on the manually labeled training data. Each classifier is then applied to the large collection of unlabeled events E_U. For each hu-

man need category, we then select the event that has been assigned to the category with the highest confidence value as a new instance to label. Consequently, each category will receive two additional labeled events at each iteration, one from the event expression classifier and another one from the event context classifier.[3] Both sets of newly labeled events are then added to the labeled set E_L, and each of the classifiers is re-trained on the expanded set of labeled data. Because the classifiers have different views of the events, the new instances labeled by one classifier serve as fresh training instances for the other, unlike self-training with a single classifier where it is learning entirely from its own predictions. The following section describes the co-training algorithm in more detail.

4.4.1 The Co-Training Algorithm

Our co-training algorithm is shown in Algorithm 1. The input to the algorithm are the sets of labeled events E_L and unlabeled events E_U. Each event is associated with both an event expression and the set of sentences in which it occurs in the blogs corpus.

For each iteration, the event expression classifier is first trained using the labeled events E_L with the event expression view. Then, we construct an event context view X_{con} for each event in the labeled set E_L. The context sentences are used differently depending on the type of context model (described in Section 4.2). An event context classifier is then trained using the context view X_{con}. Both classifiers are then independently applied to the unlabeled events E_U. For each human need category, each classifier selects one event to label based on its most confident prediction. All of the newly labeled events are then added to the labeled training set E_L, and the process repeats.

4.4.2 Prediction with Co-Trained Classifiers

The co-training process simultaneously trains two classifiers, so here we explain how we use the resulting classifiers after the co-training process has finished. For each event e in the test set, we apply both the event expression classifier and the event context classifier, which each produce a probability distribution over the human need categories. Then we explore two different methods to combine the two probability distributions for each test event: (1) **sum**, we compute the final probability vector $\boldsymbol{p}(e)$ by applying the element-wise summarization operation to the two predicted probability vectors; (2) **product**, we compute the final $\boldsymbol{p}(e)$ as the element-wise product of the two vectors. Then, the final probability vector is normalized to make sure the sum of probabilities over all classes is 1. Finally, we predict an event's human need category as the one with the highest probability.

[3]The event expression classifier first selects from unlabeled events, then the event context classifier does the selection. This ensures that there are 16 new events in total at each iteration.

Algorithm 1 Co-Training Algorithm

Input: Labeled E_L, unlabeled E_U events
while Not maximum iteration **do**
  Train the event expression classifier on E_L
  Construct context view (X_{con}) of E_L
  Train the event context classifier on X_{con}
  Apply the event expression classifier to E_U and select new labeled events (E_{exp})
  Apply the event context classifier to E_U and select new labeled events (E_{con})
  Update labeled events:
$$E_L = E_L \cup E_{exp} \cup E_{con}$$
end while

5 Evaluation

We conducted experiments to evaluate the methods described in Section 4. For all of our experiments, the results are reported based on 3-fold cross-validation on the 542 affective events manually labeled with human need categories. We show the average results over 3-folds in the following sections. For development, we used a distinct set of events labeled during preliminary studies. We did not tune any of the models, using only their default parameter settings. We present experimental results in terms of precision, recall, and F1 score, macro-averaged over the human need categories.

5.1 Performance of Event Expression Classifiers

Table 4 shows the results[4] for the event expression classifiers. We also evaluated the ability of the LIWC lexicon (Pennebaker et al., 2007) to label the event expressions. We manually aligned the relevant LIWC categories with our human need categories, as shown in Table 3. Then we labeled each event by identifying the human need category of each word in the event phrase and assign-

[4]Since we report the average precision, recall, F1 score over 3-folds, the F1 score can be smaller than both precision and recall in some cases.

ing the most frequent category to the event.[5] If no words were assigned to our categories, we labeled the event as None. The top row of Table 4 shows that LIWC achieved 39% recall but only 47.7% precision. The reason is that some categories in LIWC are more generalized compared with the definitions of our corresponding categories. For example, the words "abandon" and "damage" belong to the Affect category (corresponding to our Emotion category) in LIWC. However, based on our definition the event "my house was damaged" actually belongs to the Finance category. In this way, the Emotion category is overly generalized which leads to low precision for this class.

LIWC Category		**Human Need Category**
Ingest	→	Physiological
Health, Body, Death	→	Health
Leisure	→	Leisure
Social	→	Social
Money, Work	→	Finance
Inhib, Insight	→	Cognition
Affect	→	Emotion

Table 3: LIWC Mapping to Human Need Categories.

The LR and SVM rows in Table 4 show the performance of the logistic regression (LR) and support vector machine (SVM) classifiers, respectively. We evaluated classifiers with bag-of-words features (BOW) and classifiers with event embedding features (Embed), computed as the average of the embeddings for all words in the event expression. We also tried adding semantic category features from LIWC to each feature set, denoted as +SemCat. The results show that the Embed features performed best for both the LR and SVM classifiers. Adding the SemCat features improved upon the bag-of-word representations, but not the embeddings.

The last two rows of Table 4 show the performance of two RNN classifiers, one using lexical words as input ($\text{RNN}^{\text{Words}}$) and one using pre-trained word embeddings as input ($\text{RNN}^{\text{EmbedSeq}}$). The $\text{RNN}^{\text{EmbedSeq}}$ system takes the sequence of word embeddings as input rather than the average embeddings. As with the other classifiers, the word embedding feature representations performed best, achieving an F1 score 54.4%, which is comparable to the F1 score of the LR^{Embed} system. However, the RNN's precision was only 58%, compared to 64.2% for the logistic regression model, with only 2% higher recall that does not fully compensate for the lower precision. Neural net models often need large training sets, so the relatively small size of our training data may not be ideal for an RNN.

[5]For ties, we remove a component one by one in the order of Agent, PP, Theme until we obtain a majority label.

Method	Precision	Recall	F1
LIWC	47.7	39.0	38.6
LR^{BOW}	33.6	28.7	27.3
$\text{LR}^{\text{BOW+SemCat}}$	55.2	39.6	41.9
$\text{LR}^{\text{Embed+SemCat}}$	60.1	49.3	51.9
LR^{Embed}	**64.2**	51.7	**54.8**
SVM^{BOW}	52.3	43.1	44.8
$\text{SVM}^{\text{BOW+SemCat}}$	51.0	45.9	46.8
$\text{SVM}^{\text{Embed+SemCat}}$	50.4	48.4	48.6
$\text{SVM}^{\text{Embed}}$	51.3	50.7	50.5
$\text{RNN}^{\text{Words}}$	45.2	39.6	40.1
$\text{RNN}^{\text{EmbedSeq}}$	58.0	**53.7**	54.4

Table 4: Performance of Event Expression Classifiers

Overall, we concluded that the logistic regression classifier with event embedding features (LR^{Embed}) achieved the best performance because of its F1 score (54.8%) and higher precision (64.2%).

5.2 Performance of Event Context Classifiers

Table 5 shows the performance[4] of the event context classifiers described in Section 4.2. Since logistic regression worked best in the previous experiments, we only evaluated logistic regression classifiers in our remaining experiments. The results show that using each context sentence as an individual training instance ($\text{Context}^{\text{SentBOW}}$ and $\text{Context}^{\text{SentEmbed}}$) substantially outperformed the classifiers that merged all the context sentences as a single training instance ($\text{Context}^{\text{AllBOW}}$ and $\text{Context}^{\text{AllEmbed}}$). Overall, the best performing system $\text{Context}^{\text{SentEmbed}}$ achieved an F1 score of 44.3% with 59.1% Precision.

Method	Precision	Recall	F1
$\text{Context}^{\text{AllBOW}}$	20.6	18.0	17.8
$\text{Context}^{\text{AllEmbed}}$	38.2	29.9	29.1
$\text{Context}^{\text{SentBOW}}$	48.2	31.4	32.8
$\text{Context}^{\text{SentEmbed}}$	**59.1**	**41.9**	**44.3**

Table 5: Performance of Event Context Classifiers

It is worth noting that the precision of the best contextual classifier was only 5% below that of the best event expression classifier, while there was a 10% difference in their recall. Since they achieved (roughly) similar levels of precision and represent complementary views of events, a co-training

framework seemed like a logical way to use them together to gain additional benefits from unlabeled event data.

We also created a classifier that combined event expression features and event context features together. But combining them did not improve performance.

5.3 Performance of Self-Training and Co-Training Models

In this section, we evaluate the weakly supervised self-training and co-training methods that additionally use unlabeled data. To keep the number of unlabeled events manageable, we only used events in the AffectEvent dataset that had frequency ≥ 100, which produced an unlabeled data set of 23,866 events.

We used the best performing event expression classifier (LR^{Embed}) in these models, and the co-training framework includes the best performing event context classifier ($Context^{SentEmbed}$) as well. We also experimented with the **sum** and **product** variants for co-training (described in Section 4.4.2), which are denoted as $CoTrain^{sum}$ and $CoTrain^{prod}$. We ran both the self-training and co-training methods for 20 iterations.

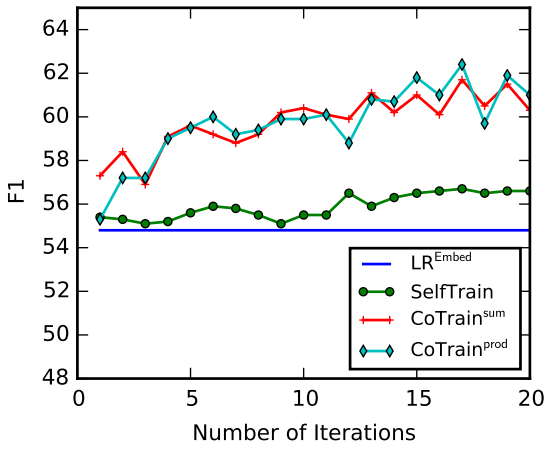

Figure 2: Learning Curves

Figure 2 tracks the performance of the self-training and co-training models after each iteration, in terms of F1 score. The flat line shows the F1 score for the best classifier that uses only labeled data (LR^{Embed}). Both types of models yield performance gains from iteratively learning with the unlabeled data, but the co-training models perform substantially better than the self-training model. Even after just 5 iterations, co-training achieves an F1 score over 58%, and by 20 iterations performance improves to $> 60\%$.

Table 6 shows the results for these models after 20 iterations, which was an arbitrary stopping criterion, and after 17 iterations, which happened to produce the best results for all three systems. The first two rows show the results of the best performing event context classifier ($Context^{SentEmbed}$) and best performing event expression classifier (LR^{Embed}) from the previous experiments, for the sake of comparison.

Table 6 shows that after 20 iterations, the $CoTrain^{prod}$ model performed best, yielding an F1 score of 61% compared to 54.8% for the LR^{Embed} model. Furthermore, we see gains in both recall and precision.

All three systems performed best after 17 iterations, so we show those results as well to give an idea of additional gains that would be possible if we could find an optimal stopping criterion. Our data set was small so we did not feel that we had enough data to fine-tune parameters, but we see the potential to further improve performance given additional tuning data.

Method	Precision	Recall	F1
Supervised Models			
$Context^{SentEmbed}$	59.1	41.9	44.3
LR^{Embed}	64.2	51.7	54.8
After 20 Iterations			
SelfTrain	63.2	54.2	56.6
$CoTrain^{sum}$	66.2	58.2	60.3
$CoTrain^{prod}$	**67.1**	**58.7**	**61.0**
Best Results, After 17 Iterations			
SelfTrain	63.5	54.1	56.7
$CoTrain^{sum}$	68.6	59.0	61.7
$CoTrain^{prod}$	**69.7**	**59.5**	**62.4**

Table 6: Performance of Self-Training and Co-Training

Table 7 shows a breakdown of the performance across the individual human need categories for two models: the best event expression classifier and the best co-training model ($CoTrain^{prod}$ after 17 iterations). We see that the co-training model outperformed the LR^{Embed} model on every category. Co-training improved performance the most for the Finance and Cognition categories, yielding F1 score gains of +12% and +16%, respectively, and notably improving both recall and precision.

5.4 Analysis

We manually examined our system's predictions to better understand its behavior. We found that most of the correctly classified Physiological

Category	LR^{Embed} Pre	Rec	F1	CoTrain^{Prod} Pre	Rec	F1
Physiological	82	57	67	81	68	**74**
Health	65	40	49	68	50	**57**
Leisure	62	59	60	69	63	**66**
Social	61	72	66	68	79	**73**
Finance	61	31	40	67	44	**52**
Cognition	75	31	42	92	46	**58**
Emotion	60	75	66	64	74	**69**
None	47	49	48	48	52	**50**

Table 7: Breakdown of results across Human Need categories. Each cell shows Precision, Recall, and F1.

events were related to food, while the correctly classified Cognition events were primarily about learning and understanding. Our method missed many events for the Health, Finance, and Cognition classes. For Health, many medical symptoms were not recognized, such as *"my face looks pale"* and *"I puked"*. For Finance, the system missed events related to possessions (e.g., *"engine stopped running"* and *"my clock is wrong"*) and jobs (e.g., *"I went to resign"*).

We also took a closer took at which categories were confused with other categories. Figure 3 shows the confusion matrix between CoTrain^{Prod} and the gold annotations. Each cell shows the total number of confusions across the 3-folds of cross-validation. The category names are abbreviated as Physiological (Phy), Health (Hlth), Leisure (Leis), Social (Socl), Finance (Fnc), Cognition (Cog), and Emotion (Emo). #Tot denotes the total number of events in each row or column.

Pred. \ Gold	Phy	Hlth	Leis	Socl	Fnc	Cog	Emo	None	#Tot
Phy	13	1	0	0	1	0	0	2	17
Hlth	1	26	1	0	1	1	4	8	42
Leis	1	1	48	4	0	1	4	10	69
Socl	0	6	4	84	2	3	10	11	120
Fnc	1	0	2	0	13	0	1	5	22
Cog	0	0	0	0	0	12	1	2	15
Emo	1	5	12	12	3	1	91	16	141
None	2	13	8	8	9	8	17	51	116
#Tot	19	52	75	108	29	26	128	105	542

Figure 3: Confusion between Predictions and Gold.

The co-training model had difficulty distinguishing the None category from other classes, presumably because None does not have its own semantics but is used for affective events that do not belong to any of the other categories. We also see that the system often confuses Emotion with Leisure and Social. This happens because many event expressions contain words that refer to emotions. Our guidelines instructed annotators to focus on the event and assign the Emotion label only when no event is described beyond an emotion (e.g., *"I was thrilled"*). Consequently, the gold label of *"I love journey"* is Leisure and *"I'm worried about my mom"* is Social, but both were classified by the system as Emotion. In future work, it may be advantageous to allow event expressions to be labeled as both an explicit Emotion and a Human Need category based on the target of the emotion.

6 Conclusions

In this work, we introduced a new challenge to recognize the reason for the affective polarity of events in terms of basic human needs. We designed four types of classification methods to categorize affective events according to human need categories, exploiting both labeled and unlabeled data. We first evaluated event expression and event context classifiers, trained using only labeled data. Then we designed self-training and co-training methods to additionally exploit unlabeled data. A co-training model that simultaneously trains event expression and event context classifiers produced substantial performance gains over the individual models. However, performance on the human need categories still has substantial room for improvement. In future work, obtaining more human annotations will be useful to build a better human needs categorization system. In addition, applying and analyzing the human needs of affective events in narrative stories and conversations is a fruitful and interesting direction for future research.

7 Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Number IIS-1619394. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We are very grateful to Tianyu Jiang and Yuanyuan Gao for participating in the manual annotation effort.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally Normalized Transition-Based Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*. ACM.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Sector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27.

Snigdha Chaturvedi, Dan Goldwasser, and Hal Daumé III. 2016. Ask, and Shall You Receive? Understanding Desire Fulfillment in Natural Language Text. In *Processings of the 30th AAAI Conference on Artificial Intelligence*.

Yoonjung Choi and Janyce Wiebe. 2014. +/-EffectWordNet: Sense-level Lexicon Acquisition for Opinion Inference. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing*.

François Chollet et al. 2015. Keras. https://keras.io.

Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. 2013. Benefactive/Malefactive Event and Writer Attitude Annotation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Lingjia Deng and Janyce Wiebe. 2014. Sentiment Propagation via Implicature Constraints. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Lingjia Deng and Janyce Wiebe. 2015. Joint Prediction for Entity/Event-Level Sentiment Analysis using Probabilistic Soft Logic Models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Haibo Ding, Tianyu Jiang, and Ellen Riloff. 2018. Why is an Event Affective? Classifying Affective Events based on Human Needs. In *AAAI-18 Workshop on Affective Content Analysis*.

Haibo Ding and Ellen Riloff. 2015. Extracting Information about Medication Use from Veterinary Discussions. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Haibo Ding and Ellen Riloff. 2016. Acquiring Knowledge of Affective Events from Blogs Using Label Propagation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Haibo Ding and Ellen Riloff. 2018. Weakly Supervised Induction of Affective Events by Optimizing Semantic Consistency. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Andrew B Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. 2009. May All Your Wishes Come True: A Study of Wishes and How to Recognize Them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Andrew Gordon and Reid Swanson. 2009. Identifying Personal Stories in Millions of Weblog Entries. In *Third International Conference on Weblogs and Social Media, Data Challenge Workshop*.

A. Goyal, E. Riloff, and H. Daumé III. 2010. Automatically Producing Plot Unit Representations for Narrative Text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

A. Goyal, E. Riloff, and H. Daumé III. 2013. A Computational Model for Plot Units. *Computational Intelligence* 29(3):466–488.

Abhishek Kumar and Hal Daumé. 2011. A Co-training Approach for Multi-view Spectral Clustering. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*.

Huayi Li, Arjun Mukherjee, Jianfeng Si, and Bing Liu. 2015. Extracting Verb Expressions Implying Negative Opinions. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. 2014. Major Life Event Extraction from Twitter based on Congratulations/Condolences Speech Acts. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*.

Abraham Harold Maslow, Robert Frager, James Fadiman, Cynthia McReynolds, and Ruth Cox. 1970. *Motivation and Personality*, volume 2. Harper & Row New York.

Manfred Max-Neef, Antonio Elizalde, and Martin Hopenhayn. 1991. *Human Scale Development: Conception, Application and Further Reflections*. The Apex Press.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-training for Parsing. In *Proceedings of the Conference on Human Language Technology and North American Chapter of the Association for Computational Linguistics*.

Rada Mihalcea. 2004. Co-training and Self-training for Word Sense Disambiguation. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*.

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.

James W Pennebaker, Roger J Booth, and Martha E Francis. 2007. Linguistic Inquiry and Word Count: LIWC2007. *Austin, TX: liwc.net* .

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

William Phillips and Ellen Riloff. 2002. Exploiting Strong Syntactic Heuristics and Co-Training to Learn Semantic Lexicons. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*.

Elahe Rahimtoroghi, Jiaqi Wu, Ruimin Wang, Pranav Anand, and Marilyn A Walker. 2017. Modelling Protagonist Goals and Desires in First-Person Narrative. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation Frames: A Data-Driven Investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Lena Reed, Jiaqi Wu, Shereen Oraby, Pranav Anand, and Marilyn A. Walker. 2017. Learning Lexico-Functional Patterns for First-Person Affect. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ.

Hoa Trong Vu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Acquiring a Dictionary of Emotion-Provoking Events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Xiaojun Wan. 2009. Co-training for Cross-lingual Sentiment Classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.

Rui Xia, Cheng Wang, Xin-Yu Dai, and Tao Li. 2015. Co-training for Semi-supervised Sentiment Classification Based on Dual-view Bags-of-words Representation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.