Weakly Supervised Induction of Affective Events by Optimizing Semantic Consistency

Haibo Ding and **Ellen Riloff**
School of Computing
University of Utah
Salt Lake City, UT 84112
{hbding, riloff}@cs.utah.edu

Abstract

To understand narrative text, we must comprehend how people are affected by the events that they experience. For example, readers understand that graduating from college is a positive event (achievement) but being fired from one's job is a negative event (problem). NLP researchers have developed effective tools for recognizing explicit sentiments, but affective events are more difficult to recognize because the polarity is often implicit and can depend on both a predicate and its arguments. Our research investigates the prevalence of affective events in a personal story corpus, and introduces a weakly supervised method for large scale induction of affective events. We present an iterative learning framework that constructs a graph with nodes representing events and initializes their affective polarities with sentiment analysis tools as weak supervision. The events are then linked based on three types of semantic relations: (1) semantic similarity, (2) semantic opposition, and (3) shared components. The learning algorithm iteratively refines the polarity values by optimizing semantic consistency across all events in the graph. Our model learns over 100,000 affective events and identifies their polarities more accurately than other methods.

Introduction

When people discuss events, people understand not only the literal meaning of the event but they also infer the probable affective state of the person who experienced the event. For example, if someone says that they got a job, broke a record, or went to Disneyland, then most people assume these were desirable experiences and offer congratulations or shared excitement. Conversely, if someone says that they were fired from their jobs, broke their arms, or went to a funeral, then most people assume these were undesirable experiences and offer sympathy or assistance. Understanding the affective state associated with an event is essential for many NLP tasks including narrative text understanding (Goyal, Riloff, and Daumé III 2013; Lehnert 1981), opinion analysis (Deng, Wiebe, and Choi 2014), and sarcasm recognition (Riloff et al. 2013). We refer to events that typically affect people in positive or negative ways as *affective events*.

Many NLP tools have been developed for sentiment analysis, and some research has begun to focus specifically on affective events, but prior methods still do not consistently or accurately recognize them. Our research aims to improve affective event recognition by extracting a large collection of stereotypically affective events from a personal story corpus. This paper offers several contributions to this topic: (1) we present a manual annotation study of randomly sampled events, which demonstrates the prevalence of affective events (nearly 40% of all events), (2) we represent events as rich structures that include a predicate, agent, theme, and prepositional phrase, and (3) we present a novel weakly supervised method for inducing a large set of affective events ($>$ 100,000) from an unannotated text corpus.

This paper introduces an iterative learning framework that automatically induces a large collection of affective events from a personal story corpus. First, the corpus is parsed and events are extracted into a predicate-argument structure and incorporated into a graph, where each node represents a distinct event. The events are then linked based on three types of semantic relations: (1) *semantic similarity*, (2) *semantic opposition* and (3) *shared components*. Next, initial polarity values are assigned to events using sentiment analysis tools. Although sentiment tools are not very accurate for many affective events, they can recognize events that have explicitly affective language (e.g., "I had fun" or "I yelled in anger"). Consequently, the initialization step serves as noisy supervision. The learning algorithm is then tasked with inferring more accurate event polarities by iteratively refining the polarity values to optimize for the overall *semantic consistency* in the graph. Intuitively, the algorithm encourages semantically similar events to have similar polarity, semantically opposing events to have opposite polarity, and events to have polarity values consistent with their components. We applied this model to a corpus of nearly 1.4 million personal stories and induced a collection of $>$175,000 affective events, which achieved higher recall and precision on our affective event data set than existing affective lexicons and learning models.

Related Work

Many resources for sentiment analysis have been created, including the MPQA Subjectivity Lexicon (Wilson, Wiebe, and Hoffmann 2005), SenticNet (Cambria, Olsher, and Rajagopal 2014; Cambria et al. 2015), SentiWordNet (Baccianella, Esuli, and Sebastiani 2010), the NRC Emotion

Lexicon (Mohammad and Turney 2010), and many others. Most of this work has focused on recognizing sentiments and emotions explicitly expressed in text. However, there has been growing interest in recognizing other types of affective indicators. Research closely related to affective events includes bootstrapped learning of *patient polarity verbs*, which impart affective polarity to their patients (Goyal, Riloff, and Daumé III 2010; 2013), research on the connotation of words and word senses (Kang et al. 2014), and connotation frames (Rashkin, Singh, and Choi 2016) which infer connotative polarities for a verb's arguments from the writer's and entity's perspective. These works focus on individual verbs, in contrast to our richer event structures. Another related line of work is on +/- effect events (Choi and Wiebe 2014) that have a positive/negative effect on their entities, but the effect does not need to be "affective" per se (e.g., baking a cake is considered to be positive for the cake because the cake is created). This work is focused on opinion analysis through implicature rules (Deng, Wiebe, and Choi 2014), rather than the effects of events on people. Recently, Reed at al. (2017) learned patterns associated with first-person affect, which improved recognition of affective sentences when used alongside supervised learners.

Our work is also related to work on identifying "emotion-provoking events" (Vu et al. 2014) and "major life events" extraction work (Li et al. 2014) although their work did not identify polarity. We previously (Ding and Riloff 2016) designed an Event Context Graph (ECG) model to induce a set of affective events. However, our previous model is fundamentally different from the one in this paper. The ECG model used a traditional label propagation algorithm to learn affective events. In this paper we designed a new optimization framework to enforce semantic consistency. In addition, the ECG model was constructed based entirely on discourse properties and event co-occurrence. In contrast, the graph in this paper is built based on three types of semantic relations. We compare these two models in the Evaluation section.

Graph based learning methods have been previously used for sentiment lexicon induction (e.g., (Rao and Ravichandran 2009; Velikovich et al. 2010)). Most of this work aims to learn the prior polarity for individual words. In contrast, we use a rich event representation that includes a verb and its arguments to distinguish between specific types of events.

Affective Event Data

The goal of our research is to study the prevalence of affective events in narrative text and to develop a weakly supervised method to learn a large collection of affective events. As the text corpus, we used the ICWSM 2009 and 2011 Spinn3r data sets[1], which together contain over 177 million blog posts. To focus our efforts on narrative text about events in people's daily lives, we extracted personal blog posts by applying a *personal story classifier* (Gordon and Swanson 2009). We further removed stories with no first person mentions and then removed near-duplicates using SpotSigs (Theobald, Siddharth, and Paepcke 2008). This process resulted in 1,383,425 personal blog posts.

[1]http://www.icwsm.org/data/

Extracting Event Structures

Most previous affective resources and methods identify the polarity of individual words or short phrases. Our research focuses on events, so we wanted to create an event representation that is specific enough to distinguish between event expressions that have substantially different semantics. We settled on a frame-like event structure that has 4 components: ⟨**Agent**, **Predicate**, **Theme**, **PP**⟩. The **Predicate** is a simple verb phrase, which typically corresponds to an action or state. We require that an event must also have an **Agent** or a **Theme**[2]. Some previous work has used an Agent/Predicate/Object representation (namely (Ding and Riloff 2016)), but our event structure additionally includes a **prepositional phrase (PP)** argument, which we believe is essential to distinguish between dramatically different event types. For example, "go to beach" is a very different kind of event than "go to prison". Similarly "get into college" is fundamentally different from "get into argument". Although multiple PPs are common and can be important, we allow only a single PP to prevent the representation from becoming overly specific. If multiple PPs are attached to the VP, we include only the closest one.

To create the event structures, we used StanfordCoreNLP (Manning et al. 2014) for POS and NER tagging and SyntaxNet (Andor et al. 2016) for parsing. A **Predicate** is extracted for each finite verb and can also include a particle, infinitive verb, and negator, if they are present. For example, the Predicate could be "eat" or "not want to take off". The **Agent** and **Theme** are extracted from the dependency relations. We use the term "Theme" loosely and allow an adjective to fill the Theme role in predicate adjective constructions (e.g., "dad is brave"). We extract minimal noun phrases for the Agent, Theme, and PP, which could be named entities, nominals with noun premodifiers, or pronouns.[3] Active and passive voice constructions are normalized. For example, "I was killed by him" and "he killed me" are both represented by the structure: "⟨he, kill, me, -⟩". For the verbs "be" and "have", we require both an Agent and Theme. All words are lemmatized in the event structures.

Our goal is to analyze affective events from the perspective of the experiencer (i.e., the blogger). So we only keep events that satisfy at least one of the following criteria. (1) The event has a first person reference (e.g., "I", "my"). (2) The event mentions a family member (e.g., "mom"). We assume that the affective state of the blogger usually parallels that of family members (e.g., "mom is sick" is undesirable for both mom and the blogger). We manually compiled a list of 92 family terms. (3) The event does not mention any other people[4]. In this case, we assume that the event pertains to the blogger (e.g., "the computer died").[5] We do not extract

[2]These are approximated using syntax rules, not SRL.

[3]Our Agent and Theme representation also differs from (Ding and Riloff 2016) in that they only extract single words.

[4]An entity is identified as "other people" if it is a second or third person pronoun, a PERSON Named Entity, or nominal person mention based on WordNet (e.g. "plumber").

[5]This simple approach could undoubtedly be improved with discourse analysis, but we leave that for future work.

events that only mention other people because they may be describing someone else's experience, not the blogger's.

This process resulted in 19,794,187 unique events. Finally, we filtered events with frequency < 5 and obtained 571,424 unique events as our *affective event data set*.

Manual Analysis of Affective Events

A key question for research on this topic is: how prevalent are affective events? To answer this question and to create a test set for evaluation, we conducted a manual annotation effort to label a random set of events from our personal story data with affective polarities. We defined four categories:

Positive: An event that is desirable, enjoyable, pleasant or beneficial.

Negative: An event that is undesirable, unenjoyable, unpleasant or detrimental.

Neutral: An event that most people would not consider to be positive or negative.

Mixed: An event that is rarely neutral but is often considered positive by some people and negative by others.

Our work focuses on recognizing the prior polarities of events, that are stereotypical, independent of context. Therefore, we randomly selected 1,500 events from our affective event data and asked three people to manually label them. We measured their pairwise inter-annotator agreement (IAA) using Cohen's kappa (κ), which were κ=.76 κ=.70, and κ=.69. We then assigned the majority label to each event as the gold standard polarity. Only one event was labeled as Mixed, so we concluded that mixed polarity events are rare and abandoned this category. We discarded the 1 Mixed event, and also 9 events that received three different labels from the annotators, which resulted in a gold standard data set of 1,490 events labeled as Positive, Negative, or Neutral. The distribution of the polarities is shown below.

POS	NEG	NEU
295 (20%)	264 (18%)	931 (62%)

We see that 38% of the randomly selected events have a positive or negative affective polarity, with slightly more positive events. These results suggest that affective events are pervasive, comprising nearly 4 of every 10 events, which illustrates the importance of being able to recognize the affective polarity of events for narrative text understanding.

Table 1 shows examples of annotated events. Of these 1,490 manually annotated events, we randomly selected 1,000 as our *test set* for evaluation and use the remaining 490 events as a *development set* for tuning parameters.

POSITIVE:	⟨I, play, music, -⟩
⟨kid, look up, -, to me ⟩	⟨cost, be, low, -⟩
⟨I, go, -, to block party⟩	⟨someone, save, me, -⟩
⟨my confidence, rise, -, -⟩	⟨I, attend, show, -⟩
⟨I, dance, -, with my friend⟩	⟨I, kiss, her, -⟩
NEGATIVE:	⟨girl, laugh, -, at me⟩
⟨I, get, -, into argument⟩	⟨I, be, bummed, -⟩
⟨I, drop, my phone, in toilet⟩	⟨dog, pass away, -, -⟩
⟨house phone, not work, -, -⟩	⟨my face, look, pale, -⟩
⟨I, wake up, -, at 3 am⟩	⟨tear, pour, -, from eye⟩
NEUTRAL:	⟨I, pack up, my bag, -⟩
⟨I, decide to rent, car, -⟩	⟨trunk, be, open, -⟩
⟨tour bus, pull up, -, -⟩	⟨I, scribble, -, -⟩
⟨I, read, -, over post⟩	⟨I, have, staple, -⟩
⟨I, wake up, -, around 6 ⟩	⟨I, look, -, at sentence⟩

Table 1: Examples of Gold Standard Affective Events

Semantic Consistency Model

The goal of our work is to design a weakly supervised method to automatically learn a large set of affective events. The key idea is to define a graph of events and semantic relations between events, with noisy supervision providing initial affective polarities. Using an optimization framework, we can then learn the correct polarity values by enforcing semantic consistency across the relations in the graph.

Figure 1 shows an illustration of the semantic relations graph. The graph contains nodes for events and components and three types of edges: semantic similarity edges that link semantically similar event pairs, semantic opposition edges (dotted line) that link semantically opposing event pairs, and event-component edges that connect an event with its components individually. The learning model will prefer that semantically similar events have similar affective polarities, semantically opposing events have opposing affective polarities. Event-component relations are used by the learner to infer that the polarity of an event is related to the polarity of its individual components.

Although existing affective resources often fail to recognize many affective events, they do well at recognizing events that contain explicit emotions or strong positive/negative terms (e.g., "I had fun" or "the experience was a disaster"). So we take advantage of previously developed affective tools to provide initial polarity values for each node as noisy supervision for our model.

The basic flow of our method contains 3 steps. First, we build a graph containing event and component nodes using the semantic relations among events. Second, we obtain initial polarities for events and components using existing sentiment analysis tools. Finally, we design an iterative learning algorithm to infer the polarities of events by optimizing the semantic consistency in the graph.

Semantic Relations Graph

We create a graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ consists of event nodes (v_i) and component nodes (c_k). The event nodes correspond to the 571,424 unique events extracted from our personal story data. The component nodes are created by decomposing each event structure into its parts: a predicate and up to 3 arguments[6]. If a predicate is negated, then the negation is also attached to all of the event's arguments. For example, the event ⟨I, not get, award, -⟩ will yield two component nodes: "not get" and "not award".[7] A *polarity vector*

[6]We do not create component nodes for pronouns.

[7]This strategy for handling negation is overkill because the negation usually only applies to one part of an event. But determining the best scope for the negation is challenging (e.g., "not have beer" is roughly equivalent to "have no beer" but for our model "no beer" is more useful semantically than "not have"). More sophisticated negation handling is a worthwhile avenue for future work.

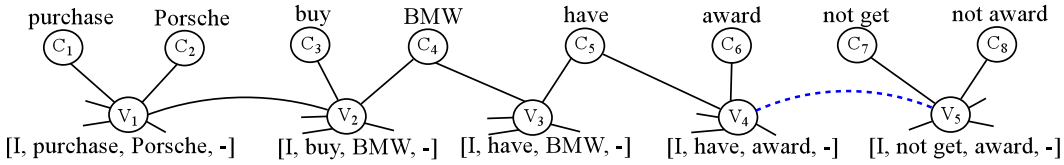

Figure 1: Semantic Relations Graph

is associated with each node, which denotes a distribution over 3 polarity values <POSITIVE, NEUTRAL, NEGATIVE> for the associated event or component.

The edge set consists of three types of edges: *similarity edges*, *opposition edges*, and *event-component edges*.

Similarity Edges: Our model assumes that events with similar semantic meaning will usually have similar affective polarity (e.g. "have party" and "have celebration"). We use semantic embeddings to assess the similarity of events. We compute an event embedding as the average of the GloVe vectors (Pennington, Socher, and Manning 2014) of its words[8]. For each event node i, we create an edge between i and its five most similar events. The edge weight W_{ij}^{sim} between nodes i and j is the cosine similarity of their embedding vectors.

Opposition Edges: Our model also assumes that events with opposite meanings often have opposite polarities (e.g. "I win" and "I was defeated"). To construct opposition edges, we identify events with a negated predicate. We refer to non-negated events as "affirmative". For each negated event i, we remove the negator and compute its embedding as described above. Then, we compute the cosine similarities between event i and all affirmative events and select the 10 most similar affirmative events as its opposition neighbors. The opposition edge weight W_{ij}^{opp} between nodes i and j is the cosine similarity of their embedding vectors.

Event-Component Edges: Many event expressions refer to the same or just slightly different activities (e.g., ⟨I, have, birthday party, -⟩ and ⟨I, attend, birthday party, -⟩). We hypothesized that learning the affective polarity of individual concepts could help to generalize beyond specific event expressions. For example, if "birthday party" has positive polarity, then events mentioning a birthday party will often have positive polarity too. Of course, many events include words that have different affective polarities. But if we link an event node with nodes for all of its components, then all of this information can be taken into account during the learning process. To explore this idea, we create edges between event node i and all of the nodes corresponding to its components. For example, the event ⟨phone, fall, in toilet, -⟩ will be connected with 3 component nodes: "phone", "fall", and "in toilet". The edge weight between event i and component k is set to be $W_{ik}^{cmp} = 1$.

Learning by Optimizing Semantic Consistency

This section presents our algorithm for learning the affective polarities of events.

[8]We use GloVe vectors (200d) pretrained on 27B tweets.

Initialization

Each event node i and component node k are assigned initial polarity vectors, which are obtained from external sentiment resources. Intuitively, the idea is to initialize the model with noisy supervision, which the learner uses in combination with the semantic relations, graph structure, and optimization function to infer the correct polarity for each node. For polarity initialization, we experimented with a variety of affective lexicons and classification models and found that the MPQA lexicon (Wilson, Wiebe, and Hoffmann 2005) combined with an aggregated contextual classifier performed best (this "Combo" method is described in the Evaluation section).

Semantic Consistency Metrics

Our model infers polarity values by optimizing the semantic consistency in the graph (i.e. minimizing the inconsistency in the graph). We use KL-divergence to measure the inconsistency between polarity vectors.

We will refer to the initial polarity vector for event i as $\boldsymbol{v}_i^0$. The model iteratively updates an estimated polarity vector, $\boldsymbol{v}_i$, which is encouraged to remain similar to the initial vector by minimizing the inconsistency between them. The initial polarity values never change and serve as an anchor to prevent thrashing during the learning process. Formally, the inconsistency between $\boldsymbol{v}_i$ and $\boldsymbol{v}_i^0$ is computed as: $D(\boldsymbol{v}_i||\boldsymbol{v}_i^0) = \sum_l^L v_i(l) \log \frac{v_i(l)}{v_i^0(l)}$, where L is the set of polarity labels. The inconsistency between the estimated polarity vector for a component node k and its initial polarity vector is similarly measured as $D(\boldsymbol{c}_k||\boldsymbol{c}_k^0)$.

Inconsistency is also measured across all three types of semantic relation edges. For similar event pairs i and j, their inconsistency is measured as the difference between their polarity vectors: $D(\boldsymbol{v}_i||\boldsymbol{v}_j)$. For opposing event pairs i and j, the inconsistency is computed as $D(\boldsymbol{v}_i||\boldsymbol{v}_j\boldsymbol{H})$. We use the *exchange matrix* $\boldsymbol{H} = \left[\begin{smallmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{smallmatrix}\right]$ to switch the positive and negative values of the polarity vector. The indices of $\boldsymbol{H}$ represent: 0(pos), 1(neu), 2(neg). Finally, we measure the inconsistency between an event and each of its components. Since KL-divergence is asymmetric, the inconsistency between an event i and component k needs to be decomposed into two parts: $D(\boldsymbol{v}_i||\boldsymbol{c}_k)$ and $D(\boldsymbol{c}_k||\boldsymbol{v}_i)$. This maintains the symmetric property of the final objective, which allows us to directly derive closed form update equations.

Weight Normalization

In our graph, some nodes are highly connected but others are not. To account for this, we normalize the semantic similar-

ity weight matrix as $\tilde{\boldsymbol{W}}^{sim} = \boldsymbol{A}^{-\frac{1}{2}}\boldsymbol{W}^{sim}\boldsymbol{A}^{-\frac{1}{2}}$ where $\boldsymbol{A}$ is a diagonal matrix and $A_{ii} = \sum_{j=1}^{n} W_{ij}^{sim}$. We similarly normalize the semantic opposition weight matrix $\boldsymbol{W}^{opp}$. For the event-component edges, different events may link to different numbers of components, and vice versa. To normalize the weights, we first obtain the transpose $\boldsymbol{W}^{cmp'}$ of $\boldsymbol{W}^{cmp}$, and then obtain $\tilde{\boldsymbol{W}}^{cmp}$ and $\tilde{\boldsymbol{W}}^{cmp'}$ by performing row normalization on $\boldsymbol{W}^{cmp}$ and $\boldsymbol{W}^{cmp'}$.

The Objective and Update Functions

Our complete semantic consistency (SC) model incorporates all of the previously mentioned inconsistency measures with a single objective function, shown in Eq.1.

$$J_{sc} = \beta \sum_{i=1}^{n} D(\boldsymbol{v}_i||\boldsymbol{v}_i^0) + \sum_{(i,j)} \tilde{W}_{ij}^{sim} D(\boldsymbol{v}_i||\boldsymbol{v}_j) + \sum_{(i,j)} \tilde{W}_{ij}^{opp} D(\boldsymbol{v}_i||\boldsymbol{v}_j\boldsymbol{H}) + \gamma \sum_{(i,k)} \tilde{W}_{ik}^{cmp} D(\boldsymbol{v}_i||\boldsymbol{c}_k) + \gamma \sum_{(k,i)} \tilde{W}_{ki}^{cmp'} D(\boldsymbol{c}_k||\boldsymbol{v}_i) + \eta \sum_{k=1}^{m} D(\boldsymbol{c}_k||\boldsymbol{c}_k^0) \quad (1)$$

This objective computes the overall inconsistency in the graph, and our goal is to minimize the objective to obtain the best polarity estimates. The n and m are the numbers of event and components nodes, and (i, j) denotes connected node pairs. The hyperparameters control the relative importance of each corresponding term. In our experiments, our full model uses the following values: $\beta = 0.6$, $\gamma = 0.8$, $\eta = 0.1$, which were selected on our development data.

Since KL-divergence is convex, the objective in Eq.1 is convex when the parameters are non-negative. This guarantees that our model will converge at a global minimum. We designed an iterative algorithm that alternately updates $\boldsymbol{v}$ and $\boldsymbol{c}$. Let $\boldsymbol{v}_i^t$ and $\boldsymbol{c}_k^t$ denote polarity vectors for event i and component k at iteration t. We first optimize the objective over $\boldsymbol{v}_i$, given $\boldsymbol{v}_i^t$ and $\boldsymbol{c}_i^t$ by computing the derivative for $\boldsymbol{v}_i^{t+1}$. The update for $\boldsymbol{v}_i^{t+1}$ is shown in Eq.2

$$\boldsymbol{v}_i^{t+1} \propto \exp \frac{1}{O_i}\Big(\beta \log \boldsymbol{v}_i^0 + \sum_j \tilde{W}_{ij}^{sim} \log \boldsymbol{v}_j^t + \sum_j \tilde{W}_{ij}^{opp} \log \boldsymbol{v}_j^t\boldsymbol{H} + \gamma \sum_k \tilde{W}_{ik}^{cmp} \log \boldsymbol{c}_k^t\Big) \quad (2)$$

where $O_i = \beta + \sum_j \tilde{W}_{ij}^{sim} + \sum_j \tilde{W}_{ij}^{opp} + \gamma \sum_k \tilde{W}_{ik}^{cmp}$.

Given $\boldsymbol{v}_i^{t+1}$, we obtain the update equation for $\boldsymbol{c}_k^{t+1}$ by computing the derivative for $\boldsymbol{c}_k^{t+1}$. The update equation is shown in Eq.3.

$$\boldsymbol{c}_k^{t+1} \propto \exp \frac{\eta \log \boldsymbol{c}_k^0 + \gamma \sum_i \tilde{W}_{ki}^{cmp'} \log \boldsymbol{v}_i^{t+1}}{\eta + \gamma \sum_i \tilde{W}_{ki}^{cmp'}} \quad (3)$$

The learning algorithm is shown below, which iteratively updates the polarity vectors on event nodes and component nodes. In our final experiments, the learning process converged after 52 iterations. When the learning is finished, for each event i we infer its polarity to be the polarity class with the highest score: $\arg\max_l \boldsymbol{v}_i(l)$.

Algorithm 1 Iterative Learning Algorithm

1: **Input:** $\boldsymbol{W}^{sim}$, $\boldsymbol{W}^{opp}$, $\boldsymbol{W}^{cmp}$, $\boldsymbol{v}^0$, $\boldsymbol{c}^0$
2: **Output:** $\boldsymbol{v} \in \mathcal{R}^{n\times|L|}$
3: **while** $\boldsymbol{v}$ has not converged **do**
4: Update $\boldsymbol{v}^t$ using Eq. 2
5: Update $\boldsymbol{c}^t$ using Eq. 3
6: **end while**
7: **return** $\boldsymbol{v}^t$

Improved Component Initialization

We hypothesized that we could improve the initial polarity values of the components through an independent learning process that exploits semantic similarities between component terms. We create a graph in which each component is connected to its 5 most similar components with edge weight U_{ij}, set to be the cosine similarity between their embeddings using GloVe vectors. The total inconsistency (J_{cmp}) is shown in Eq.4 where m is the number of components.

$$\sum_{(i,j)} \tilde{U}_{ij} D(\boldsymbol{c}_i||\boldsymbol{c}_j) + \sum_{i=0}^{m_l} D(\boldsymbol{c}_i||\boldsymbol{c}_i^s) + \mu \sum_{i=0}^{m} D(\boldsymbol{c}_i||\boldsymbol{c}_i^0) \quad (4)$$

The first term of Eq.4 measures the inconsistency between two semantically similar components. The second term measures inconsistency between the estimates and polarities ($\boldsymbol{c}^s$) from MPQA Lexicon for the m_l components contained in MPQA. The third term measures inconsistency between the estimated values and initial polarities assigned by the NRC$^{\text{AvgS}}$ aggregated contextual classifier (described in the Evaluation section). We use two types of initial values because the MPQA lexicon has high precision but low coverage, while the classifier has greater coverage but lower precision. Given the objective, we derive the update function for variable $\boldsymbol{c}_k$ by computing its derivative and iteratively updating the polarity values until convergence or 100 iterations.[9] The inferred polarity vectors are then used as the "initial" polarity vectors for the component nodes in our full SC model. This separate learning process for component nodes slightly improved our overall evaluation results.

Evaluation

We conducted extensive experiments to compare the performance of our Semantic Consistency Model with the performance of existing affective lexicons and classification models on our affective event data set. For these experiments, all lexicon or model parameters were tuned on our development set and the reported results are on our test set.

Prior Affective Lexicons and Learning Models

We evaluated the performance of five existing affective lexicons: **MPQA** (Wilson, Wiebe, and Hoffmann 2005), **SentiWordNet3.0** (**SentiWN**) (Baccianella, Esuli, and Sebastiani 2010), **+/-EffectWordNet** (**+/-EffectWN**) (Choi and Wiebe 2014), **ConnotationWordNet** (**ConnoWN**) (Kang et al. 2014), and **Connotation Frames** (Rashkin, Singh, and

[9] We used μ=0.1 in experiments based on the development set.

Choi 2016) for which we evaluated both the effect on subject (**ConnoFrameS**) and the effect on object (**ConnoFrameO**). Since our event structures contain multiple words, we computed the polarity score for an event as the average score of its words. Most of these lexicons assign polarity scores over a range of values, where high values mean strong polarity and low values mean weak polarity. To explore the best way to use each lexicon, we defined a threshold λ for each lexicon. For lexicons with polarity values ranging from [-1,+1], we assigned events with a score $> \lambda$ as positive, $<$-λ as negative and an absolute value $|\text{score}| \leq \lambda$ as neutral. For lexicons with polarity values ranging from [0,+1], we assigned events with a score between [0.5-λ, 0.5+λ] as neutral, $<$0.5-λ as negative, and $>$0.5+λ as positive. We found that the following values achieved the best F1 scores on our development data and were therefore used throughout our experiments: λ=0 for MPQA, λ=0.25 for ConnoFrameS, λ=0.3 for ConnoFrameO, λ=0.4 for ConnoWN, λ=0.5 for SentiWN, and λ=0.6 for +/-EffectWN.

Method	POS	NEG	NEU	AVG
Affective Lexicons				
ConnoWN	26.3	9.8	64.1	33.4
ConnoFrameS	32.6	21.0	64.8	39.5
ConnoFrameO	29.8	22.5	70.7	41.0
+/-EffectWN	36.3	36.7	55.3	42.8
SentiWN	33.5	27.3	73.9	44.9
MPQA	57.8	54.9	**80.1**	64.3
Event Structure Classifiers				
LR^{BOW}	25.6	16.1	78.2	40.0
StanfordSA	37.5	12.4	77.7	42.6
LR^{Embed}	50.8	44.9	79.7	58.5
NRC	58.6	55.9	79.6	64.7
Contextual Models				
ECG	28.1	46.1	65.9	46.7
NRC^{AvgS}	51.2	52.0	70.7	58.0
Combo	**60.7**	**58.3**	79.9	**66.3**

Table 2: F1 Scores for Lexicons and Models

The top portion of Table 2 shows the results for these lexicons, including F1 scores for the positive (POS), negative (NEG), and neutral (NEU) polarities, and the macro-averaged F1 score across all three polarities. The MPQA lexicon performs the best on our data.

For learning based methods, we first evaluated several *event structure classifiers* by applying them directly to the sequence of words in an event structure. We replicated the NRC-Canada sentiment classifier (**NRC**) (Mohammad, Kiritchenko, and Zhu 2013), and trained the classifier using the SemEval 2014 Task 9 tweet data. We also evaluated the Stanford sentiment analysis (**StanfordSA**) system, which is a neural network model. In addition, we trained two logistic regression classifiers on our development data. One classifier ($\mathbf{LR^{BOW}}$) uses bag of words features for all words in an event. A second classifier ($\mathbf{LR^{Embed}}$) uses word embedding features, which is computed as the average of the word embeddings in an event.[10] The middle of Table 2 shows that the NRC classifier achieved the best result.

[10]We use the GloVe vectors (200d) pretrained on 27B tweets.

We also evaluated two types of *contextual models*, which exploit the contexts surrounding an event. For each event, we applied the NRC classifier to every sentence that it occurs in and produced a distribution of polarity values across the sentences. We call this method $\mathbf{NRC^{AvgS}}$. We also evaluated the previous Event Context Graph (**ECG**) model (Ding and Riloff 2016) on this new set of randomly sampled events. We applied it to our full data set of nearly 1.4 million blog posts. The ECG model produces polarity values ranging from [-1, +1], so we tuned a λ parameter on our development data as we did for the lexicons. We used the best value: λ=0.15[11]. Table 2 shows that NRC^{AvgS} was the best contextual model.

MPQA was the best lexicon, and $\mathbf{NRC^{AvgS}}$ was the best contextual model, and we hypothesized that combining these complementary methods might perform even better.[12] So we created a **Combo** system that linearly combines the predictions of both models. For an event e, we compute its polarity vector as $\alpha * \text{PolarityVector}_{\text{NRC}^{\text{AvgS}}}(e) + (1-\alpha) * \text{PolarityVector}_{\text{MPQA}}(e)$. The last row of Table 2 shows the results for this **Combo** method, which achieved the highest F1 score, where α=0.7 based on the development set.

Results for the Semantic Consistency Model

Table 3 shows the results for our Semantic Consistency (SC) model alongside the best system (**Combo**) that utilized existing methods for comparison. We initialized the polarity vectors of the event nodes in the SC model using the **Combo** method, which produces a distribution over the 3 polarity values for each event.

Method	POS	NEG	NEU	Average		
	F1	F1	F1	Pr	Rec	F1
Combo	60.7	58.3	79.9	67.5	65.6	66.3
SC+sim	58.6	62.9	82.3	72.6	65.7	67.9
+opp	59.9	63.8	83.4	75.0	65.8	69.0
+cmp	**63.7**	**66.7**	**83.7**	75.2	68.9	**71.4**

Table 3: Results for Semantic Consistency (SC) Model

The **SC+sim** row shows results for the SC model using only the semantic similarity edges, which substantially improves precision (+5%) over the Combo baseline. The **+opp** row shows results for adding the semantic opposition edges as well, which further improves precision to 75% while maintaining the same level of recall. The **+cmp** row shows results for the full model, which also includes component nodes connected to corresponding events. These shared component relations improve recall from 65.8% to 68.9% without any loss of precision. Overall, the full semantic consistency model achieved both higher recall (65.6% $\rightarrow$ 68.9%) and higher precision (67.5% $\rightarrow$ 75.2%) compared to the best results achieved with previous methods. The macro-averaged F1 score improved from 66.3% to 71.4%, which is

[11]The original experiments by (Ding and Riloff 2016) evaluated only positive and negative events that received the highest polarity scores. Our experiments evaluate the polarity assigned to *all* events in our test set, which were randomly sampled.

[12]We also tried to combine MPQA and the NRC classifier, but using MPQA and NRC^{AvgS} was better.

statistically significant at $p < 0.01$ based on the paired bootstrap test (Berg-Kirkpatrick, Burkett, and Klein 2012).

Analysis

We took a closer look at the affective events identified by the SC model in terms of both quality and quantity. First, we identified all events whose initial polarity (produced by the Combo model) was changed by the SC model. Table 4 shows that the most frequent changes were from positive or negative polarity to neutral, and from neutral to negative. Table 5 shows the recall and precision differences between the models for each polarity. The large shifts from positive/negative to neutral correspond to the precision gains, and the shifts from neutral to negative correspond to the increased recall for negative polarity.

Combo → SC	#Total	#Correct	Accuracy
POS → NEU	24	19	79%
NEU → NEG	18	13	72%
NEG → NEU	45	32	71%
POS → NEG	4	2	50%
NEG → POS	8	3	38%
NEU → POS	3	1	33%

Table 4: Polarity Changes between Combo and SC models

Method	POS		NEG		NEU	
	Pr	Rec	Pr	Rec	Pr	Rec
Combo	67.7	55.1	56.3	60.5	78.4	81.4
SC Model	75.7	55.1	70.4	63.3	79.3	88.5

Table 5: Precision (Pr) and Recall (Rec) Breakdowns

Table 6 shows some correct and incorrect examples of events whose polarity changed. The SC model seems to have learned that certain predicates (verbs) are typically neutral, such as "open" and "want". We also observe that many of its errors involve negated terms, suggesting that more sophisticated negation handling may be needed.

A goal of this research is to produce an *affective event lexicon* that can be used by the NLP community as knowledge of affective events. Toward this end, we created lexicons of varying sizes by selecting events that were assigned a positive or negative polarity with value $\geq \tau$ in the polarity vector, to effect recall/precision trade-offs. Table 7 shows the precision and recall on our test set using different thresholds, and also the total number of affective events extracted from the corpus for corresponding thresholds. The bottom row (*max*) shows the lexicon produced by assigning every event the polarity that has the highest value. We notice that our model produced more negative than positive events, which is consistent with that of the initialization method (i.e. the Combo results in Table 5). So we believe this is influenced by the initialization method.

The bottom row of Table 7 shows that the complete lexicon has over 175K affective events with precision >70%. Setting τ=0.5 still produces 111K events with > 80% precision for NEG and > 90% for POS events. Increasing the threshold to 0.6 reduces the lexicon to > 69,000 affective events with > 93% precision for both polarities. This analysis shows that the SC model can be used to automatically generate large, high-quality collections of affective events. We plan to construct a lexicon of the affective events learned by the SC model and make it freely available to the research community.

POSITIVE → NEUTRAL	
Correct Examples:	⟨ I, open, my email, -⟩
⟨box, be, open, -⟩	⟨my friend, start, work, -⟩
⟨I, want, photo, -⟩	⟨I, want, bag, -⟩
Incorrect Examples:	⟨my family, stay, with me, -⟩
⟨I, win, class, -, -⟩	⟨band, rock, -, -⟩
NEUTRAL → NEGATIVE	
Correct Examples:	⟨food, not be, tasty, -⟩
⟨I, break, heart, -⟩	⟨friend, disappoint, me⟩
⟨I, be, bummed, -⟩	⟨I, start, sniffle, -⟩
⟨tear, pour, -, from eye⟩	⟨none, be, -, for me⟩
Incorrect Examples:	
⟨we, see, cave, -⟩	⟨we, steal, glance, -⟩
NEGATIVE → NEUTRAL	
Correct Examples:	⟨feeling, go, -, through me⟩
⟨I, feel, -, about stuff⟩	⟨I, need, bowl, -⟩
⟨I, call to work, -, -⟩	⟨answer, not be, one, -⟩
Incorrect Examples:	⟨my memory, not serve, me, -⟩
⟨house phone, not work, -, -⟩	⟨I, not function, -, at work ⟩

Table 6: Correct and Incorrect Examples

τ	POS		NEG		#AffectiveEvents		
	Pr	Rec	Pr	Rec	#pos	#neg	#total
0.7	100	18.7	93.7	16.9	19031	18947	37978
0.6	96.9	31.8	93.4	32.2	30584	38523	69107
0.5	90.1	41.4	80.2	45.8	48594	62998	111592
max	75.7	55.1	70.4	63.3	82398	92743	175141

Table 7: Quality and Size of Different Lexicons

Conclusion

In this work, we investigated the prevalence of affective events in personal story blogs, and designed a novel, weakly supervised semantic consistency model for automatically inducing a high-quality affective event lexicon. We did extensive experiments to evaluate existing sentiment lexicons and learning methods on a new affective event data set. The results show that our model achieves better performance than other methods, and learns over 100,000 affective events with high precision. However, the recall for positive and negative events have substantial room for improvement, so future work is needed to obtain more comprehensive coverage of affective events.

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.

References

Andor, D.; Alberti, C.; Weiss, D.; Severyn, A.; Presta, A.; Ganchev, K.; Petrov, S.; and Collins, M. 2016. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Baccianella, S.; Esuli, A.; and Sebastiani, F. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*.

Berg-Kirkpatrick, T.; Burkett, D.; and Klein, D. 2012. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Cambria, E.; Fu, J.; Bisio, F.; and Poria, S. 2015. Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Cambria, E.; Olsher, D.; and Rajagopal, D. 2014. Senticnet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*.

Choi, Y., and Wiebe, J. 2014. +/-EffectWordNet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing*.

Deng, L.; Wiebe, J.; and Choi, Y. 2014. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *Proceedings of the 25th International Conference on Computational Linguistics*.

Ding, H., and Riloff, E. 2016. Acquiring knowledge of affective events from blogs using label propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Gordon, A., and Swanson, R. 2009. Identifying personal stories in millions of weblog entries. In *Third International Conference on Weblogs and Social Media, Data Challenge Workshop*.

Goyal, A.; Riloff, E.; and Daumé III, H. 2010. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

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

Kang, J. S.; Feng, S.; Akoglu, L.; and Choi, Y. 2014. ConnotationWordNet: Learning connotation over the word+sense network. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Lehnert, W. G. 1981. Plot units and narrative summarization. *Cognitive Science* 5(4):293–331.

Li, J.; Ritter, A.; Cardie, C.; and Hovy, E. 2014. Major life event extraction from twitter based on congratulations/condolences speech acts. In *Proceedings of Empirical Methods in Natural Language Processing*.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics*.

Mohammad, S. M., and Turney, P. D. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*.

Mohammad, S. M.; Kiritchenko, S.; and Zhu, X. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Rao, D., and Ravichandran, D. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Rashkin, H.; Singh, S.; and Choi, Y. 2016. Connotation Frames: A data-driven investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Reed, L.; Wu, J.; Oraby, S.; Anand, P.; and Walker, M. A. 2017. Learning lexico-functional patterns for first-person affect. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; and Huang, R. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Theobald, M.; Siddharth, J.; and Paepcke, A. 2008. Spotsigs: robust and efficient near duplicate detection in large web collections. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*.

Velikovich, L.; Blair-Goldensohn, S.; Hannan, K.; and McDonald, R. 2010. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Vu, H. T.; Neubig, G.; Sakti, S.; Toda, T.; and Nakamura, S. 2014. Acquiring a dictionary of emotion-provoking events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.