Minimally Supervised Learning of Affective Events Using Discourse Relations

Jun Saito Yugo Murawaki Sadao Kurohashi
Graduate School of Informatics, Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
{saito, murawaki, kuro}@nlp.ist.i.kyoto-u.ac.jp

Abstract

Recognizing affective events that trigger positive or negative sentiment has a wide range of natural language processing applications but remains a challenging problem mainly because the polarity of an event is not necessarily predictable from its constituent words. In this paper, we propose to propagate affective polarity using discourse relations. Our method is simple and only requires a very small seed lexicon and a large raw corpus. Our experiments using Japanese data show that our method learns affective events effectively without manually labeled data. It also improves supervised learning results when labeled data are small.

1 Introduction

Affective events (Ding and Riloff, 2018) are events that typically affect people in positive or negative ways. For example, getting money and playing sports are usually positive to the experiencers; catching cold and losing one’s wallet are negative. Understanding affective events is important to various natural language processing (NLP) applications such as dialogue systems (Shi and Yu, 2018), question-answering systems (Oh et al., 2012), and humor recognition (Liu et al., 2018). In this paper, we work on recognizing the polarity of an affective event that is represented by a score ranging from $-1$ (negative) to $1$ (positive).

Learning affective events is challenging because, as the examples above suggest, the polarity of an event is not necessarily predictable from its constituent words. Combined with the unbounded combinatorial nature of language, the non-compositionality of affective polarity entails the need for large amounts of world knowledge, which can hardly be learned from small annotated data.

In this paper, we propose a simple and effective method for learning affective events that only requires a very small seed lexicon and a large raw corpus. As illustrated in Figure 1, our key idea is that we can exploit discourse relations (Prasad et al., 2008) to efficiently propagate polarity from seed predicates that directly report one’s emotions (e.g., “to be glad” is positive). Suppose that events $x_1$ and $x_2$ are in the discourse relation of CAUSE (i.e., $x_1$ causes $x_2$). If the seed lexicon suggests $x_2$ is positive, $x_1$ is also likely to be positive because it triggers the positive emotion. The fact that $x_2$ is known to be negative indicates the negative polarity of $x_1$. Similarly, if $x_1$ and $x_2$ are in the discourse relation of CONCESSION (i.e., $x_2$ in spite of $x_1$), the reverse of $x_2$’s polarity can be propagated to $x_1$. Even if $x_2$’s polarity is not known in advance, we can exploit the tendency of $x_1$ and $x_2$ to be of the same polarity (for CAUSE) or of the reverse polarity (for CONCESSION) although the heuristic is not exempt from counterexamples.

We transform this idea into objective functions and train neural network models that predict the polarity of a given event.

We trained the models using a Japanese web corpus. Given the minimum amount of supervision, they performed well. In addition, the combination of annotated and unannotated data yielded a gain over a purely supervised baseline when labeled data were small.

2 Related Work

Learning affective events is closely related to sentiment analysis. Whereas sentiment analysis usually focuses on the polarity of what are described (e.g., movies), we work on how people are typically affected by events. In sentiment analysis, much attention has been paid to compositionality. Word-level polarity (Takamura et al., 2005;
Wilson et al., 2005; Baccianella et al., 2010) and the roles of negation and intensification (Reitan et al., 2015; Wilson et al., 2005; Zhu et al., 2014) are among the most important topics. In contrast, we are more interested in recognizing the sentiment polarity of an event that pertains to common-sense knowledge (e.g., getting money and catching cold).

Label propagation from seed instances is a common approach to inducing sentiment polarities. While Takamura et al. (2005) and Turney (2002) worked on word- and phrase-level polarities, Ding and Riloff (2018) dealt with event-level polarities. Takamura et al. (2005) and Turney (2002) linked instances using co-occurrence information and/or phrase-level coordinations (e.g., “A and B” and “A but B”). We shift our scope to event pairs that are more complex than phrase pairs, and consequently exploit discourse connectives as event-level counterparts of phrase-level conjunctions.

Ding and Riloff (2018) constructed a network of events using word embedding-derived similarities. Compared with this method, our discourse relation-based linking of events is much simpler and more intuitive.

Some previous studies made use of document structure to understand the sentiment. Shimizu et al. (2018) proposed a sentiment-specific pre-training strategy using unlabeled dialog data (tweet-reply pairs). Kaji and Kitsuregawa (2006) proposed a method of building a polarity-tagged corpus (ACP Corpus). They automatically gathered sentences that had positive or negative opinions utilizing HTML layout structures in addition to linguistic patterns. Our method depends only on raw texts and thus has wider applicability.

3 Proposed Method

3.1 Polarity Function

Our goal is to learn the polarity function \( p(x) \), which predicts the sentiment polarity score of an event \( x \). We approximate \( p(x) \) by a neural network with the following form:

\[
p(x) = \tanh(\text{Linear}(\text{Encoder}(x))).
\]  

Encoder outputs a vector representation of the event \( x \). Linear is a fully-connected layer and transforms the representation into a scalar. \( \tanh \) is the hyperbolic tangent and transforms the scalar into a score ranging from \(-1\) to \(1\). In Section 4.2, we consider two specific implementations of Encoder.

3.2 Discourse Relation-Based Event Pairs

Our method requires a very small seed lexicon and a large raw corpus. We assume that we can automatically extract discourse-tagged event pairs, \((x_{i1}, x_{i2}) (i = 1, \ldots)\) from the raw corpus. We refer to \(x_{i1}\) and \(x_{i2}\) as former and latter events, respectively. As shown in Figure 1, we limit our scope to two discourse relations: CAUSE and CONCESSION.

The seed lexicon consists of positive and negative predicates. If the predicate of an extracted
event is in the seed lexicon and does not involve complex phenomena like negation, we assign the corresponding polarity score (+1 for positive events and −1 for negative events) to the event. We expect the model to automatically learn complex phenomena through label propagation. Based on the availability of scores and the types of discourse relations, we classify the extracted event pairs into the following three types.

**AL (Automatically Labeled Pairs)** The seed lexicon matches (1) the latter event but not the former event, and (3) their discourse relation type is CAUSE or CONCESSION. If the discourse relation type is CAUSE, the former event is given the same score as the latter. Likewise, if the discourse relation type is CONCESSION, the former event is given the opposite of the latter’s score. They are used as reference scores during training.

**CA (Cause Pairs)** The seed lexicon matches neither the former nor the latter event, and their discourse relation type is CAUSE. We assume the two events have the same polarities.

**CO (Concession Pairs)** The seed lexicon matches neither the former nor the latter event, and their discourse relation type is CONCESSION. We assume the two events have the reversed polarities.

### 3.3 Loss Functions

Using AL, CA, and CO data, we optimize the parameters of the polarity function \( p(x) \). We define a loss function for each of the three types of event pairs and sum up the multiple loss functions.

We use mean squared error to construct loss functions. For the AL data, the loss function is defined as:

\[
\mathcal{L}_{\text{AL}} = \frac{1}{N_{\text{AL}}} \sum_{i=1}^{N_{\text{AL}}} (r_{i2} - p(x_{i2}))^2 + \lambda_{\text{AL}} \frac{1}{N_{\text{AL}}} \sum_{i=1}^{N_{\text{AL}}} (r_{i1} - p(x_{i1}))^2,
\]

where \( x_{i1} \) and \( x_{i2} \) are the \( i \)-th pair of the AL data. \( r_{i1} \) and \( r_{i2} \) are the automatically-assigned scores of \( x_{i1} \) and \( x_{i2} \), respectively. \( N_{\text{AL}} \) is the total number of AL pairs, and \( \lambda_{\text{AL}} \) is a hyperparameter.

For the CA data, the loss function is defined as:

\[
\mathcal{L}_{\text{CA}} = \lambda_{\text{CA}} \frac{1}{N_{\text{CA}}} \sum_{i=1}^{N_{\text{CA}}} (p(y_{i1}) - p(y_{i2}))^2 + \mu \frac{1}{N_{\text{CA}}} \sum_{i=1}^{N_{\text{CA}}} \sum_{u \in \{y_{i1}, y_{i2}\}} (1 - p(u))^2.
\]

\( y_{i1} \) and \( y_{i2} \) are the \( i \)-th pair of the CA pairs. \( N_{\text{CA}} \) is the total number of CA pairs. \( \lambda_{\text{CA}} \) and \( \mu \) are hyperparameters. The first term makes the scores of the two events closer while the second term prevents the scores from shrinking to zero.

The loss function for the CO data is defined analogously:

\[
\mathcal{L}_{\text{CO}} = \lambda_{\text{CO}} \frac{1}{N_{\text{CO}}} \sum_{i=1}^{N_{\text{CO}}} (p(z_{i1}) + p(z_{i2}))^2 + \mu \frac{1}{N_{\text{CO}}} \sum_{i=1}^{N_{\text{CO}}} \sum_{u \in \{z_{i1}, z_{i2}\}} (1 - p(u))^2.
\]

The difference is that the first term makes the scores of the two events distant from each other.

### 4 Experiments

#### 4.1 Dataset

**4.1.1 AL, CA, and CO**

As a raw corpus, we used a Japanese web corpus that was compiled through the procedures proposed by Kawahara and Kurohashi (2006). To extract event pairs tagged with discourse relations, we used the Japanese dependency parser KNPl and in-house postprocessing scripts (Saito et al., 2018). KNP used hand-written rules to segment each sentence into what we conventionally called clauses (mostly consecutive text chunks), each of which contained one main predicate. KNP also identified the discourse relations of event pairs if explicit discourse connectives were present (Prasad et al., 2008) such as “のため” (because) and “ので” (in spite of) were present. We treated Cause/Reason and Condition in the original tagset (Kawahara et al., 2014) as CAUSE and Concession, respectively. Here is an example of event pair extraction.

---

1 http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP
2 To be precise, this discourse type is semantically broader than Concession and extends to the area of Contrast.
<table>
<thead>
<tr>
<th>Type of pairs</th>
<th># of pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL (Automatically Labeled Pairs)</td>
<td>1,000,000</td>
</tr>
<tr>
<td>CA (CAUSE Pairs)</td>
<td>5,000,000</td>
</tr>
<tr>
<td>CO (CONCESSION Pairs)</td>
<td>5,000,000</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the AL, CA, and CO datasets.

Because I made a serious mistake, I got fired.

From this sentence, we extracted the event pair of "重大な失敗を犯す" (I make a serious mistake) and "仕事をクビになる" (I get fired), and tagged it with CAUSE.

We constructed our seed lexicon consisting of 15 positive words and 15 negative words, as shown in Section A.1. From the corpus of about 100 million sentences, we obtained 1.4 millions event pairs for AL, 41 millions for CA, and 6 millions for CO. We randomly selected subsets of AL event pairs such that positive and negative latter events were equal in size. We also sampled event pairs for each of CA and CO such that it was five times larger than AL. The results are shown in Table 1.

4.1.2 ACP (ACP Corpus)

We used the latest version\(^3\) of the ACP Corpus (Kaji and Kitsuregawa, 2006) for evaluation. It was used for (semi-)supervised training as well. Extracted from Japanese websites using HTML layouts and linguistic patterns, the dataset covered various genres. For example, the following two sentences were labeled positive and negative, respectively:

(2) 作業が楽だ。

The work is easy.

(3) 駐車場がない。

There is no parking lot.

Although the ACP corpus was originally constructed in the context of sentiment analysis, we found that it could roughly be regarded as a collection of affective events. We parsed each sentence and extracted the last clause in it. The train/dev/test split of the data is shown in Table 2. The objective function for supervised training is:

\[
\mathcal{L}_{ACP} = \frac{1}{N_{ACP}} \sum_{i=1}^{N_{ACP}} (R_i - p(v_i))^2, \quad (5)
\]

where \(v_i\) is the \(i\)-th event, \(R_i\) is the reference score of \(v_i\), and \(N_{ACP}\) is the number of the events of the ACP Corpus.

To optimize the hyperparameters, we used the dev set of the ACP Corpus. For the evaluation, we used the test set of the ACP Corpus. The model output was classified as positive if \(p(x) > 0\) and negative if \(p(x) \leq 0\).

4.2 Model Configurations

As for Encoder, we compared two types of neural networks: BiGRU and BERT. GRU (Cho et al., 2014) is a recurrent neural network sequence encoder. BiGRU reads an input sequence forward and backward and the output is the concatenation of the final forward and backward hidden states. BERT (Devlin et al., 2019) is a pre-trained multi-layer bidirectional Transformer (Vaswani et al., 2017) encoder. Its output is the final hidden state corresponding to the special classification tag ([CLS]). For the details of Encoder, see Sections A.2.

We trained the model with the following four combinations of the datasets: AL, AL+CA+CO (two proposed models), ACP (supervised), and ACP+AL+CA+CO (semi-supervised). The corresponding objective functions were: \(\mathcal{L}_{AL} + \mathcal{L}_{CA} + \mathcal{L}_{CO} + \mathcal{L}_{ACP}\), and \(\mathcal{L}_{ACP} + \mathcal{L}_{AL} + \mathcal{L}_{CA} + \mathcal{L}_{CO}\).

4.3 Results and Discussion

Table 3 shows accuracy. As the Random baseline suggests, positive and negative labels were distributed evenly. The Random+Seed baseline made use of the seed lexicon and output the corresponding label (or the reverse of it for negation) if the event’s predicate is in the seed lexicon. We can see that the seed lexicon itself had practically no impact on prediction.

The models in the top block performed considerably better than the random baselines. The performance gaps with their (semi-)supervised counterparts, shown in the middle block, were less than

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Event polarity</th>
<th># of events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Positive</td>
<td>299,834</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>300,164</td>
</tr>
<tr>
<td>Dev</td>
<td>Positive</td>
<td>50,118</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>49,882</td>
</tr>
<tr>
<td>Test</td>
<td>Positive</td>
<td>50,046</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>49,954</td>
</tr>
</tbody>
</table>

Table 2: Details of the ACP dataset.

---

\(^3\)The dataset was obtained from Nobuhiro Kaji via personal communication.
<table>
<thead>
<tr>
<th>Training dataset</th>
<th>Encoder</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>BiGRU</td>
<td>0.843</td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td>0.863</td>
</tr>
<tr>
<td>AL+CA+CO</td>
<td>BiGRU</td>
<td><strong>0.866</strong></td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td>0.835</td>
</tr>
<tr>
<td>ACP</td>
<td>BiGRU</td>
<td>0.919</td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td><strong>0.933</strong></td>
</tr>
<tr>
<td>ACP+AL+CA+CO</td>
<td>BiGRU</td>
<td>0.917</td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td>0.913</td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td>0.500</td>
</tr>
<tr>
<td>Random+Seed</td>
<td></td>
<td>0.503</td>
</tr>
</tbody>
</table>

Table 3: Performance of various models on the ACP test set.

<table>
<thead>
<tr>
<th>Training dataset</th>
<th>Encoder</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACP (6K)</td>
<td>BERT</td>
<td>0.876</td>
</tr>
<tr>
<td></td>
<td>+AL</td>
<td><strong>0.886</strong></td>
</tr>
<tr>
<td>ACP (6K)</td>
<td>BiGRU</td>
<td>0.830</td>
</tr>
<tr>
<td></td>
<td>+AL+CA+CO</td>
<td><strong>0.879</strong></td>
</tr>
</tbody>
</table>

Table 4: Results for small labeled training data. Given the performance with the full dataset, we show BERT trained only with the AL data.

7%. This demonstrates the effectiveness of discourse relation-based label propagation.

Comparing the model variants, we obtained the highest score with the BiGRU encoder trained with the AL+CA+CO dataset. BERT was competitive but its performance went down if CA and CO were used in addition to AL. We conjecture that BERT was more sensitive to noises found more frequently in CA and CO.

Contrary to our expectations, supervised models (ACP) outperformed semi-supervised models (ACP+AL+CA+CO). This suggests that the training set of 0.6 million events is sufficiently large for training the models. For comparison, we trained the models with a subset (6,000 events) of the ACP dataset. As the results shown in Table 4 demonstrate, our method is effective when labeled data are small.

The result of hyperparameter optimization for the BiGRU encoder was as follows:

\[ \lambda_{AL} = 1, \lambda_{CA} = 0.35, \lambda_{CO} = 1, \mu = 0.5. \]

As the CA and CO pairs were equal in size (Table 1), \(\lambda_{CA}\) and \(\lambda_{CO}\) were comparable values. \(\lambda_{CA}\) was about one-third of \(\lambda_{CO}\), and this indicated that the CA pairs were noisier than the CO pairs. A major type of CA pairs that violates our assumption was in the form of “problem_negative causes solution_positive”:

(4) (悪いところがある、よくなるように努力する)

(there is a bad point, [I] try to improve [it])

The polarities of the two events were reversed in spite of the CAUSE relation, and this lowered the value of \(\lambda_{CA}\).

Some examples of model outputs are shown in Table 5. The first two examples suggest that our model successfully learned negation without explicit supervision. Similarly, the next two examples differ only in voice but the model correctly recognized that they had opposite polarities. The last two examples share the predicate “落とす” (drop) and only the objects are different. The second event “肩を落とす” (lit. drop one’s shoulders) is an idiom that expresses a disappointed feeling. The examples demonstrate that our model correctly learned non-compositional expressions.

5 Conclusion

In this paper, we proposed to use discourse relations to effectively propagate polarities of affective events from seeds. Experiments show that, even with a minimal amount of supervision, the proposed method performed well.

Although event pairs linked by discourse analysis are shown to be useful, they nevertheless contain noises. Adding linguistically-motivated filtering rules would help improve the performance.

Acknowledgments

We thank Nobuhiro Kaji for providing the ACP Corpus and Hirokazu Kiyomaru and Yudai Kishimoto for their help in extracting event pairs. This work was partially supported by Yahoo! Japan Corporation.
References


Johan Reitan, Jørgen Faret, Björn Gambåck, and Lars Bungum. 2015. Negation scope detection for twitter sentiment analysis. In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis.


Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words us-


A Appendices

A.1 Seed Lexicon

Positive Words 喜ぶ (rejoice), 嬉しい (be glad), 楽しい (be pleasant), 幸せ (be happy), 感動 (be impressed), 興奮 (be excited), 懐かしい (feel nostalgic), 好き (like), 尊敬 (respect), 安心 (be relieved), 感心 (admire), 落ち着く (be calm), 満足 (be satisfied), 療される (be healed), and スッキリ (be refreshed).

Negative Words 怒る (get angry), 悲しい (be sad), 寂しい (be lonely), 怖い (be scared), 不安 (feel anxious), 耻ずかしい (be embarrassed), 嫌 (hate), 落ち込む (feel down), 退屈 (be bored), 絶望 (feel hopeless), 辛い (have a hard time), 困る (have trouble), 憂鬱 (be depressed), 心配 (be worried), and 情けない (be sorry).

A.2 Settings of Encoder

BiGRU The dimension of the embedding layer was 256. The embedding layer was initialized with the word embeddings pretrained using the Web corpus. The input sentences were segmented into words by the morphological analyzer Juman++. The vocabulary size was 100,000. The number of hidden layers was 2. The dimension of hidden units was 256. The optimizer was Momentum SGD (Sutskever et al., 2013). The mini-batch size was 1024. We ran 100 epochs and selected the snapshot that achieved the highest score for the dev set.

BERT We used a Japanese BERT model pretrained with Japanese Wikipedia. The input sentences were segmented into words by Juman++, and words were broken into subwords by applying BPE (Sennrich et al., 2016). The vocabulary size was 32,000. The maximum length of an input sequence was 128. The number of hidden layers was 12. The dimension of hidden units was 768. The number of self-attention heads was 12. The optimizer was Adam (Kingma and Ba, 2014). The mini-batch size was 32. We ran 1 epoch.