Twitter Emotion Cause Extraction with BECR

Julia McAnallen Zoe Winkworth
University of Washington, Seattle
{jmcanal, zoew2}@uw.edu

Abstract

Identifying emotion-cause relations is a challenging problem in information extraction, with applications in a variety of commercial and academic domains. This paper describes the design and implementation of two emotion cause extraction systems: a rule-based baseline and a semi-supervised algorithm we call BECR modeled after BREDS and Snowball (Batista et al., 2015; Agichtein and Gravano, 2000). It also outlines the process of collecting and curating a dataset of tweets to use for the task of emotion cause extraction. When run on manually annotated evaluation data, BECR has a recall of 0.36, precision at 25 of 0.6 and an F score of 0.45. We also discuss other approaches to the problem and future research directions for this important, yet nascent, area of relation extraction.

1 Introduction

We present BECR – Bootstrapping Emotion Cause Relations – a semi-supervised algorithm for identifying emotions and their causes in Tweets. BECR combines linguistic patterns with neural embeddings to isolate emotions and their causes using a small number seed examples that learn to find new emotion-cause contexts in a corpus of unlabeled Twitter data. We compare recall and precision scores on hand annotated evaluation data to results from our baseline system and other related systems and discuss the effect of these results on future work.

2 Background

A sizable amount of research has focused on sentiment analysis in social media, but less work has moved beyond polarity – positive, negative or neutral sentiment – and looked to identify specific emotions expressed. Even less research has focused on linking emotions to their cause, despite the fact that the source of the emotion is often the most relevant piece of information for the groups doing the analysis, e.g. companies seeking to find out whether negative or positive emotions expressed by customers on social media are caused by the product, technical support, advertising, or something else (Lee et al., 2010; Ghazi et al., 2015; Gui et al., 2017; Li and Xu, 2014).

More concretely, extracting emotion-cause relations is the task of identifying a specific emotion expressed and pairing that emotion with its cause, where available. An example is in (1), where the emotion "hate" is in boldface type and its cause is italicized.[1] (The convention of boldfacing emotions and italicizing causes is used throughout this paper.)

1. "I **love** *this time of the year*! #giving #family"
   - emotion: love
   - cause: this time of the year

Emotions can consist of single words, e.g. "love," "hate," or multiple words, e.g. "so excited," "not afraid," etc. Causes can be single words, noun phrases, verb phrases, clauses, or entire sentences.

2. "I'm **not afraid** of *tomorrow*"
   - emotion: not afraid
   - cause: tomorrow

3. "I **hope** *they fix my car ASAP....*"
   - emotion: hope
   - cause: they fix my car ASAP

In many examples, the cause of an emotion is not explicit; it is either implied in the Tweet, or left unexpressed.

[1] Examples in (1)-(5) are from the Twitter Emotion Corpus from Mohammad (2012)

4. "school was **pretty damn good** today"
   - emotion: pretty damn good
   - cause: ?

In still other cases, a Tweet has a sentiment, and perhaps even a cause associated with the sentiment, but the emotion is implicit.

5. "*Frosteas are back at Argo*. My favorite part of winter!!! #argotea"
   - emotion: excited (implicit)
   - cause: Frosteas are back at Argo

Examples such as (4) and (5) are outside the scope of our study.

The primary published studies on emotion-cause extraction used blogs written in Chinese as their data source. They had annotators label emotions from microblog text and used a pattern based approach to extract emotion causes. In a second phase they trained a classifier to classify emotions identified using the emotion corpus based on the extracted cause (Li and Xu, 2014).

This is similar to our task in that the data used is from a casual register and therefore requires some unique handling. We also used a pattern based approach, but our approach differs in that we are extracting both emotions and causes and only use our emotion corpus to preprocess our data, filtering to only include Tweets that are likely to be expressing emotion explicitly.

To our knowledge, there are no published studies on extracting emotion-cause relations from unlabeled data using unsupervised or semi-supervised approaches. In §3.4 and §4.4 we discuss semi-supervised approaches to information extraction that we leveraged to build a new semi-supervised system to solve the unique challenges presented in identifying emotions and their causes.

3 Approach

3.1 Data

While there are many labeled datasets for sentiment analysis on Twitter data, we were unable to find publicly available labeled data sets for emotion-cause relations appropriate for our task. The labeled data that we did find was labeled for implicit emotion, meaning the gold emotion was not always mentioned explicitly in the Tweet, such as in example (4) above. This made the data set unusable for our task, since extracting implied emotions is outside the scope of this study.

3.2 Datasets

In order to handle the issue of not having data available for our task, we aggregated three different datasets and filtered them using emotion keyword lists and lexicons in order to curate a dataset that contained as many explicit emotion-cause relations as possible.

3.2.1 Semeval 2016

The data set from Semeval 2016 Task 4, subtasks B,C,D and E includes Tweets that express sentiment about popular topics, grouped by topic and labeled for sentiment on either a 2 or 5 point polarity scale (Nakov et al., 2016). There are 200 meaningful topics represented with at least 100 Tweets each. The set labeled with 2 way polarity contains a total of 17,639 Tweets, while the 5 way polarity data includes a total of 30,632 Tweets. The SemEval task this data was collected for uses both English and Arabic Tweets, but since our approach required manual creation of surface-level rules, we were only able to use the English data.

3.2.2 Semeval 2018

The data set from Semeval 2018 Task 1, subtask 'E-c' contains 10,983 total Tweets labeled for 11 different emotion categories (Mohammad et al., 2018). This corpus of tweets was collected by polling the Twitter API for tweets that included emotion-related words such as *angry*, *annoyed*, *panic*, *happy*, *elated*, *surprised*, etc. The SemEval task this data was collected for included English, Spanish and Arabic Tweets, but since our approach required manual creation of surface-level rules, we were only able to use the English data.

3.2.3 Twitter Emotion Corpus

The Twitter Emotion Corpus dataset is an automatically created large dataset of 21,051 emotion-labeled tweets using hashtags for the 6 basic Ekman emotions: joy, sadness, anger, fear, disgust, and surprise (Mohammad, 2012). Tweets were discarded that do not contain at least three valid English words, but this did not completely eliminate non-english Tweets in all instances.

3.3 Emotion Filters

We experimented with several different strategies for filtering our Tweets that did not express explicit emotions, including two different emotion lexicons and a manually curated emotion keyword

list, with and without additional synonyms from WordNet.

After some experimentation, we found the most effective approach was to use the manually curated emotion keyword list without additional information from WordNet due to issues with polysemy.

3.3.1 EmoLex

EmoLex is an Emotion Lexicon from the National Research Council Canada and is a list of 14,183 English words and their associations with eight basic emotions and two sentiments (negative and positive) manually annotated by crowdsourcing (Mohammad and Turney, 2013).

3.3.2 DepecheMood

DepecheMood is a high coverage and high-precision lexicon of 37,772 terms annotated with emotion scores for eight emotions from crowd-sourced affective annotation implicitly provided by readers of news articles (Staiano and Guerini, 2014).

3.3.3 Keyword List

Our other approach was to manually collect a list of emotion keywords, drawing from similar lists found online (Straker, 2019; DeRose, 2005). We also experimented with expanding this list by including synsets from WordNet.

3.4 Relation Extraction

Since publicly available labeled data sets for emotion-cause relations are unavailable, we attempted two small-scale experiments with neural methods using a sample set of hand-labeled data. This allowed us to explore which approaches might be fruitful given a much larger corpus of labeled data. The results of these experiments are reported in §4.3.

Manually labeling a large data set was outside the scope of this project, which made both traditional and neural machine learning approaches to this task untenable. Given time and resource constraints, we ultimately decided on a semi-supervised approach to extracting the emotion and cause relations.

There is a genealogy of semi-supervised approaches for information extraction, originating in DIPRE (Brin, 1999). This was followed by Snowball (Agichtein and Gravano, 2000), which was recently updated and modernized in BREDS (Batista et al., 2015; Batista, 2019). DIPRE is a semi-supervised system for extracting author-publication relations; Snowball and BREDS are semi-supervised systems for extracting is-a relationships, e.g. *company - location of headquarters* and *CEO - company*. For all three systems – DIPRE, Snowball and BREDS – relations are unique named entities, e.g. ***Soundhound*** is headquartered in ***Berlin***, ***Satya Nadella*** is the CEO of ***Microsoft***.

Our final system – BECR – shares the same general approach of these previous semi-supervised systems by starting with a small set of hand-labeled seeds that are used to loop through unlabeled data to find similar examples that are added into a gradually expanding pool of relation contexts. However, BECR departs from these systems in both the type of relation extracted and in key components of its expansion algorithm.

4 Methodology

4.1 Baseline 1: OpenIE

The first baseline system we created for extracting emotion-cause relations in Tweets is a pattern-based system based on Allen AI's OpenIE tool. OpenIE "runs over sentences and creates extractions that represent relations in text" (Schmitz et al., 2017). Using the OpenIE relation extraction triples, we developed four broad rules to identify contexts that contain emotion-cause relations. These four contexts are in (6)-(9) below.

6. simple verb construction, e.g. "I **love** *Bernie Sanders*"

7. make construction, e.g. "*Seeing videos of them performing at digi* has made me **excited** to see the jacks in November"

8. modal verb construction, e.g. "*The results* may **surprise** you."

9. light verb construction with emotion adjective, e.g. "I'm **so excited** for *the new episode of Hannibal* tomorrow *cries*"

In this first baseline, as well as in our subsequent systems, we found verbs and adjectives, not nouns, to be the best indicator of an emotion expression in Tweets. This is consistent with research by Lee et al. (2010) who also found that "emotion cause events tend to be expressed by verbal events than nominal events." As a result, we

restrict our extractions to emotions expressed as verbs or adjectives.

OpenIE was not intended for relation extraction from Twitter, thus its performance is unreliable for our task. A large number of potential emotion-cause relations were not identified by OpenIE, which means that they never entered the pattern extraction algorithm as candidates. (10) is an example where OpenIE does not extract a triple that can be used in our system, and entirely bypasses the part of the Tweet that has an emotion-cause relation. Examples such as this are fairly common in the OpenIE outputs.

10. (a) ***Original Tweet:***
"I'm so excited for tomorrow's stream! I say we play Batman all day then do some viewer games after :D should be a juicy one."
(b) ***Desired triple:***
(I'm; so excited for; tomorrow's stream)
(c) ***OpenIE triple:***
(:D; should be; a juicy one)

Given the limitations of OpenIE, we developed a second baseline more suited to Twitter data.

4.2 Baseline 2: TweeboParser

In order to better capture emotion-cause candidates, we created a second baseline using the TweeboParser dependency parser (Kong et al., 2014). TweeboParser was trained on a labeled corpus of 929 Tweets, consisting of 12,318 tokens, and designed expressly for analysis of Twitter data. The strategy for the second baseline was similar to the first: extract emotion-cause relations based on predefined patterns.

Since TweeboParser is not an out-of-the-box dependency parser, we wrote functions to trace dependencies, creating parent and child dependency relationship nodes, which we leveraged in the extraction rules. In some cases, the cause is a dependent of the emotion word, e.g. when the emotion word is an inflected verb, but in other cases the cause is a dependent of a shared parent, such as a modal verb. Rules (6), (8) and (9) were re-created for the TweeboParser outputs, as well as the new rule in (11). (The "make" rule in (7) was not used because of inconsistent outputs for this construction by the dependency parser.)

11. modal verb + emotion adjective construction, e.g. "You may be **interested** in *this evening's BBC documentary*"

Overall, TweeboParser accurately identifies dependencies. Its biggest shortcoming is identifying partial dependencies, which can lead to misleading, and sometimes humorous, results, such as in example (12).

12. (a) ***Original Tweet:***
"@therealgokwan what i hate is when the sun brings out the men who think they look like david beckham but dont .. put it away !!"
(b) ***Full emotion-cause extraction:***
Emotion: hate
Cause: the sun brings out the men who think they look like david beckham but dont
(c) ***TweeboParser emotion-cause extraction:***
Emotion: hate
Cause: the sun brings out the men who think

4.3 Experiments with Neural Methods

Due to the open nature of many of the emotion-cause relations we are attempting to extract, we suspect that neural-based methods would be a powerful and effective tool for this task. However, due to the lack of labeled data for this task and time constraints preventing us from obtaining the appropriate data, we were unable to fully implement any neural approaches. Instead, we tried some small-scale tests to see if framing the task in a new way would yield promising results.

In order to try these neural methods, we hand annotated 250 Tweets, using 200 for training and 50 for testing. To annotate we used a modified BIOS tagging schema, marking each token as either the beginning of an emotion or cause, or an internal token for either emotion or cause.

4.3.1 Seq2Seq for Machine Translation

First, we tried to approach the emotion-cause extraction task as a machine translation problem where the source language was a series of tokenized, POS-tagged tweets and the target language was a series of BIOS tags corresponding to the emotion-cause relations. To implement this we used Seq2Seq as configured for a machine translation task using an RNN (Britz et al., 2017).

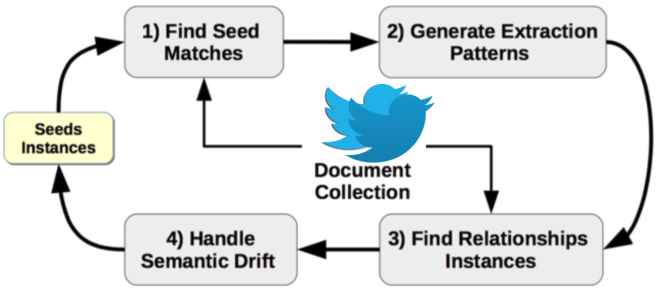

Figure 1: BECR Architecture (modified from Batista (2019))

Unsurprisingly, since the majority of tokens in the training data are neither emotion nor cause entities, the algorithm does not have a lot of positive training instances to learn from and the results were inconclusive. While it obtained a BLEU score of 66.36, it could not consistently predict the end of the sequence. However, it did get some cause tags correct, and the predicted sequences were always the correct length, which is promising and indicates that this approach might be fruitful if given an appropriate amount of training data.

4.3.2 LSTM for POS Tagging

Secondly, we tried to approach the emotion-cause extraction task as a POS-tagging problem where the input text was a series of tokenized, POS-tagged tweets and the output tag sequence was a series of BIOS tags corresponding to the emotion-cause relations. To implement this we used a LSTM model using Keras as configured for a POS-tagging task (Ivanov, 2018).

Similar to the Seq2Seq approach, the results were unimpressive since the algorithm did not have many positive instances to learn from. It resulted in a high accuracy of 87.35 due to the large number of tokens that were neither emotions nor causes, but the predictions were much worse than the Seq2Seq system. No emotion nor cause tags were predicted by the algorithm and often the length of the predictions did not match the gold labels.

It may seem more intuitive to frame emotion-cause extraction as a tagging problem than a translation task, but judging by our pilot experiments the translation task is a more promising strategy.

4.4 Final system: BECR

Our final system, BECR (Bootstrapping Emotion Cause Relations), is a semi-supervised system inspired by BREDS (Batista et al., 2015; Batista, 2019) and Snowball (Agichtein and Gravano, 2000).

4.4.1 System overview

BECR starts with a small number of emotion-cause relation seeds, which are used to find seeds with similar contexts in a process that iterates repeatedly through the set of Tweets. The seeds look for new seeds that exceed a minimum degree of similarity, measured by cosine similarity to previously identified seeds. The key components of BECR are diagrammed in Figure 1, which is based on BREDS (Batista et al., 2015; Batista, 2019). BECR diverges from BREDS and Snowball in a couple of ways, which are outlined in Table 1.

The similarity measures are summations of word embeddings for five separate contexts: 1) the **emotion** word/phrase context, 2) the **cause** context, 3) the **before** context: all words from the beginning of the Tweet up to the first relation (either emotion or cause), 4) the **between** context: all words between the emotion and cause words/phrases and 5) the **after** cause: all words following the last relation (emotion or cause) up to the last word of the Tweet. The before, between and after causes can contain zero or more words (below we discuss how we handle zero-word contexts in the calculations). (13) shows these five contexts for one Tweet.

13. "I'm **so excited** for *the new episode of Hannibal tomorrow* *cries*"
    (a) before: I'm
    (b) emotion: so excited
    (c) between: for
    (d) cause: the new episode of Hannibal
    (e) after: tomorrow *cries*

	Snowball	BREDS	BECR
context representation	TF*IDF vectors for contexts, but not for entity relations	Word2Vev word embedding vectors for contexts, but not for entity relations	GloVe (Twitter) word embedding vectors for contexts *and* emotion and cause relations
relation type	Unique *is-a* relationship, e.g. Soundcloud-Berlin	same as Snowball	Emotion-cause pairs, which can be infinite in number
number of relation types	Multiple relationship types, e.g. *HQ-ed in*, *CEO-of*	same as Snowball	One relationship type: *emotion-cause*
entity identification	Isolate entities (e.g. locations and companies) with NER entity tagger	same as Snowball	Isolate emotions with a curated emotion keyword list and find entity pair candidates (emotion, cause) through dependencies using TweeboParser dependency parser

Table 1: BECR vs. Snowball (Agichtein and Gravano, 2000) and BREDS (Batista et al., 2015)

Also note that Tweets can contain more than one emotion-cause relation, so the correspondence between Tweets and seeds is not always 1:1. Example (14) is one Tweet with two emotion-cause relations.

14. "*Hot yoga* was **intense** tonight. **Hope** *I can handle the power vinyasa class on Friday*. Eeek"
    emotion 1: intense
    cause 1: hot yoga
    emotion 2: hope
    cause 2: I can handle the power vinyasa class on Friday

In (14) the after context for the first emotion-cause relation includes the second emotion-cause relation; likewise, the before context of the second emotion-cause relation contains the first emotion-cause relation.

4.4.2 Preprocessing

Snowball and BREDS preprocess their inputs by running "a named-entity recognizer over the data to identify all location and organization entities" (Bach and Badaskar, 2007). Since emotions are not named entities and causes are often not named entities, our system cannot use the same preprocessing method.

Instead, BECR uses TweeboParser dependency parses and the TweetNLP part-of-speech "twokenizer" and part-of-speech tagger as its primary preprocessing step (Kong et al., 2014; Owoputi et al., 2013). First, Tweets that contain explicit emotion verbs or adjectives that are in the curated emotion keyword list discussed in §3.3 are separated from the corpus. Next, TweeboParser is used to produce dependency parses. Rules are applied to the DP outputs that are looser than the rules applied to the second baseline. At this stage, the goal is to identify any potential causes associated with an extracted emotion. The potential, or candidate, causes can be left-hand or right-hand dependents of either the emotion word itself, or a parent verb of the emotion word.

We also accommodate multi-word emotion phrases including negated emotions (e.g. *not happy*, *don't really like*) and emphatic emotions (e.g. *so excited*, *so very happy*). To do this, we explicitly check up to two of the words preceding the emotion word to identify modifiers tagged as negation words or adverbs and include those modifiers in the context for that emotion word.

4.4.3 Seed Matches

The first step of learning, or training, the BECR algorithm is to convert Tweets labeled for the experimental neural methods (cf. §4.3) into a dictionary of seeds that are input in BECR, in order to find the Tweets matching the seeds. Those Seeds are labeled as True matches with a confidence value of 1.0 (on a scale of 0.0 to 1.0).

In this first loop through Tweets in search of

seed contexts, the algorithm also calculates values for the embedding contexts for all True seed matches, as well as all potential seeds, for all five contexts: before, between and after contexts and emotion and cause contexts. Each of the five seed contexts is stored separately in a seed object. The embeddings are simply element-by-element summations of the GloVe embeddings for each of the words in the context, as shown in the equation below, where each *word* represents the GloVe vector for a given word:

$$context = word_1 + word_2 + ... + word_i$$

Zero-length contexts occur frequently in Tweets, e.g. example (1) above has no between context and example (2) above lacks an after context. To avoid errors when calculating context vector similarities (cf. §4.4.4), we initialize all embeddings to a very small number: 1E-28. We compared similarity calculations with and without this initialization and found the results to be the same to at least ten decimal places. This step allows us to avoid errors and also captures what we know intuitively: zero-word contexts are meaningful and should be leveraged in the similarity calculations.

4.4.4 Learning New Seed Contexts

The next step in the algorithm is to loop through Tweets one or more times, comparing the contexts of candidate seeds with matched seeds. The five similarity scores considered are the context before either the emotion or cause be_{sim}, the context between the emotion and cause bt_{sim}, the context after both the emotion and cause a_{sim}, the emotion phrase e_{sim} and the cause phrase c_{sim}. This is done by adding the five similarity scores between the seeds and the candidate seeds, as shown in the equation below. Each similarity score is multiplied by a weight. The default weights are listed below the equation.

$$\begin{aligned} total_{sim} = & \\ & w_{be}be_{sim} + w_{bt}bt_{sim} + w_a a_{sim} \\ & + w_e e_{sim} + w_c c_{sim} \end{aligned}$$

$$\begin{aligned} w_{be} = 0.2, w_{bt} = 0.5, w_a = 0.2, \\ w_e = 0.1, w_c = 0.0 \end{aligned}$$

If cosine similarity between an established seed and any of the candidate seeds surpasses a predefined Tau similarity threshold (default = 0.8) it is added to the seed matches list with a confidence score corresponding to its similarity score. (Recall that in the first iteration the seeds matching the input labels are given a confidence of 1.0.) Currently the system defaults to ten iterations through the full set of Tweets.

We also experimented with teaching the system to identify "bad" seeds by feeding it a set of prelabeled negative seeds and searching for new contexts that fall within a certain similarity threshold of those negative seeds. We call this similarity threshold negative Tau. However, the system did not behave as expected, expanding too rapidly and misappropriating true emotion-cause contexts as negative seeds, instead of positive seeds. This rapid semantic drift occurred even when just 5-10 negative seeds were fed into the system and the similarity threshold was set as high as 0.95. As a result, we do not use negative seeds in our final system configuration.

At the end of the iteration cycles, the algorithm finishes and and outputs the list of emotion-cause seed objects alongside their confidence scores. The confidence score is the argmax of each seed's similarity scores for the seeds exceeding the Tau threshold.

4.4.5 Applying Seeds to Unseen Tweets

We reserved a test set of Tweets to see how well the seeds the system learned during the learning phase in §4.4.4 above could identify emotion-cause relations in a set of unlabeled Tweets.

The test set underwent the same preprocessing as the training set: first Tweets were filtered for only those containing words in the emotion keyword list, then dependency parses were created for those Tweets with the TweeboParser.

The test phase uses the output of the algorithm in §4.4.4 as its set of seed matches. Since there are no labeled seeds in the test phase, the first step of matching tweets to labeled seeds is bypassed; i.e. the process of matching labels to Tweets to find true matches described in §.4.4.3 is skipped. The testing phase gets context scores for the five contexts: before, between, after, emotion and cause. Then it loops through all the tweets in one iteration comparing the candidate seed contexts with the contexts in the input seed matches. When a candidate seed exceeds the pre-set similarity threshold,

Emotion	Cause	
	Correct extractions	Incorrect (red) or Partial (blue) extractions
not afraid *afraid*	tomorrow it, I might miss something	 start (“Be afraid not to start)
love *seriously loving*	it when mum has a week off, that song, you and niall get away with murder	you
hate	winter, basyar assad for what he did, the rain	hate hate hate hate
so excited *super excited*	new thor on wednesday, the vamps new album on friday my naruto cards coming tomorrow	 this
hope	he has a fantastic day, i can handle the power vinyasa class on friday	that, I
sure	that function is even worse before	you know all your movie options (cf. “Sure, ANT-MAN is a cool film, but make sure you know all your movie options...”)

Table 2: A sample of BECR Extracted Relations.

it is added as an emotion-cause pair to the output list.

The testing phase is also capable of looping through the test Tweets more than once and expanding its contexts, but for our experiments we performed a single loop and did not add to the seed contexts.

5 Experiments

5.1 Outputs

BECR performs well with more straightforward emotion words and patterns, as exemplified in Table 2. BECR struggles with cases of negated causes as in “Be afraid not to start,” and cases where coreference resolution is needed as in “I hope that ...” BECR also sometimes confuses the experiencer of an emotion and a cause as in “You love ...” or “I hope ...” Furthermore, BECR has issues distinguishing polysemous words like “sure,” which can be used to express a feeling of confidence, but which can also be used in imperative constructions to encourage another person to do something.

5.2 Evaluation

Since we did not have labeled data, we had to resort to manual methods of evaluation for both our basline system and BECR. To obtain our recall results, we randomly sampled 25 Tweets from the unprocessed input data and hand annotated the sample. We then compared the sample with our outputs. Because our input data was tokenized dependency parser outputs that did not exactly match the original Tweets (mostly due to minor spacing changes), we used the SequenceMatcher package in Python to determine how similar the outputs were to the randomly sampled relations. In the case of strict match, the Tweet, emotion phrase and cause all had to match within 99%, and for relaxed match that was reduced to 70% to allow partial matches in the case of emotions and causes (e.g. ‘very excited’ and ‘excited’).

For precision scores, we used the P@K method. For BECR, we selected the results with the top 10 or 25 confidence scores and hand annotated them. Since our baseline did not have confidence scores associated with outputs, we randomly sampled 25 Tweets from the output to annotate. For strict matches, the emotion and cause were required to match exactly, while the relaxed match case allowed partial matches.

5.3 Results

Comparing BECR to our baseline system, it appears that BECR is an improvement over the baseline system for recall, P@25 and F-score.

The P@10 values do appear to be lower than the

Strict match				
System	**Precision (P@10)**	**Precision (P@25)**	**Recall-25**	**F-score-25**
BECR	0.4	0.6	0.36	0.45
Baseline 2 (TweeboParser)		0.56	0.32	0.407
Relaxed match				
System	**Precision (P@10)**	**Precision (P@25)**	**Recall-25**	**F-score-25**
BECR	0.5	0.72	0.36	0.48
Baseline 2 (TweeboParser)		0.68	0.32	0.435
Lee, et al. (2010) Results				
System	**Precision**		**Recall**	**F-score**
Redesigned	0.76		0.52	0.61
Original	0.72		0.42	0.53
Baseline	0.45		0.51	0.48

Table 3: Evaluation of BECR and Baseline System: Precision, Recall and F1 Scores

precision scores for the baseline system, but it is hard to compare those numbers since the baseline system precision is based off of a random sample while the BECR precision is not.

Inspecting the outputs, it looks like the lower P@K numbers are due to BECR learning a specific incorrect pattern and finding the same incorrect emotion-cause relation with high confidence multiple times in a single, misleading Tweet shown in (15).

15. "Hate hate hate hate hate hate hate"
    emotion: hate
    cause: hate / hate hate / hate hate hate

Results like this could probably be avoided with the addition of negative seed examples in our algorithm as long as semantic drift is avoided.

Comparing BECR to the previous work of Lee, et al. (2010) is difficult because they had labeled data, allowing for more thorough evaluation. Their task also diverged from ours slightly since their aim was to extract only emotion causes, finding emotion words using a keyword list before classifying them using the cause.

Nevertheless, our BECR results fall within range of the baseline results in Lee et al. (2010). This is promising and suggests that if we were able to provide a more extensive evaluation of our results, they might be comparable to that of an algorithm leveraging labeled data.

6 Conclusion

BECR is successful as an initial attempt at applying a semi-supervised approach to the challenging problem of emotion-cause extraction from Twitter. The system leveraged core components of previous semi-supervised systems for relation extraction, including Snowball (Agichtein and Gravano, 2000) and BREDS (Batista et al., 2015), while making necessary modifications for differences in extracting *emotion-cause* relations, versus *is-a* relationships. BECR could be improved by systematically testing different parameters to find the optimal configuration of weights for its five contexts: before, between, after, emotion, and cause. Other parameters that could be adjusted include the size of the Twitter GloVe embeddings, since our tests only used the smaller 25-dimensional GloVe embeddings. We were limited in the number of parameters we could experiment with because our evaluation is a largely manual process.

The algorithm is currently designed to incorporate negative or "bad" seeds, i.e. seeds which contain a candidate emotion and cause which are not a true emotion-cause relation. However, incorporating negative seeds worsened the performance of the algorithm. Adding even 5-10 negative seeds led to dramatic semantic drift, quickly generalizing and incorrectly marking genuine emotion-cause relation examples as false. In future work, adjustments could be made to the way negative seeds expand and find contexts to avoid the semantic drift that occurs in the current implementation.

In addition to modifying hyperparamters of the system, BECR could be improved through adjustments to its architecture. Instead of extracting candidate emotion-cause relations using an emotion keyword list alongside a pattern-based system from dependency parser outputs, Twitter GloVe embeddings could be used to expand the scope of the emotions and causes extracted from Tweets. By not restricting emotion words to a fixed list, the system could potentially capture metaphorical expressions of emotion, e.g. "fuming" for "angry." However, this modification could also introduce more pronounced semantic drift, necessitating adjustments in other hyperparameters or in the algorithm's method for expanding emotion-cause contexts.

In updates to the system we also propose to down-weight candidate seeds by the confidence scores of the seed matches, so as not to give too much weight to lower-confidence seed matches. For example, if a new seed candidate has a similarity score of 0.85, and it achieves this similarity score with a seed match that has a confidence score of 0.9, the new seed would have a confidence score of: 0.85 * 0.9 = 0.765.

In the long run, the most impactful direction for future work would be to label much more data for emotion-cause relations, thus making neural machine learning methods viable options.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Nguyen Bach and Sameer Badaskar. 2007. A review of relation extraction. *Literature review for Language and Statistics II*.

David S Batista, Bruno Martins, and Mário J Silva. 2015. Semi-supervised bootstrapping of relationship extractors with distributional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 499–504.

David Batista. 2019. Breds. `https://github.com/davidsbatista/BREDS`.

Sergey Brin. 1999. Extracting patterns and relations from the world wide web. In Alberto Mendelzon Paolo Atzeni and Giansalvatore Mecca, editors, *The World Wide Web and Databases*, page 17283, Berlin, Heidelberg. Springer.

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. 2017. Massive Exploration of Neural Machine Translation Architectures. *ArXiv e-prints*, March.

Steven J. DeRose. 2005. Emotion words. `http://www.derose.net/steve/resources/emotionwords/ewords.html`.

Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 152–165. Springer.

Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. A question answering approach to emotion cause extraction. *arXiv preprint arXiv:1708.05482*.

George-Bogdan Ivanov. 2018. Build a pos tagger with an lstm using keras. `https://nlpforhackers.io/lstm-pos-tagger-keras/`.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1001–1012.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53. Association for Computational Linguistics.

Weiyuan Li and Hua Xu. 2014. Text-based emotion classification using emotion cause extraction. *Expert Systems with Applications*, 41(4):1742–1749.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Saif M Mohammad. 2012. # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, SemEval '16, San Diego, California, June. Association for Computational Linguistics.

Olutobi Owoputi, Brendan OConnor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.

Michael Schmitz, Harinder Pal, Bhadra Mani, and Michal Guerquin. 2017. Open ie. `https://github.com/allenai/openie-standalone`.

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

David Straker. 2019. Basic emotions. `http://changingminds.org/explanations/emotions/basicemotions.htm`.