Negation Effect on Emotion Recognition in Twitter

Zach Stallbohm[1], Yang Liu[2], and Zhongyu Wei[2]
University of Texas at Dallas
Richardson, Texas 75080, USA
zxs110030@utdallas.edu[1]
{yangl,zywei}@hlt.utdallas.edu[2]

Abstract

In this paper, we explore to model negation into the task of emotion recognition. Our system uses negation cues to detect the negation, then English grammar phrases and simple rules to dynamically detect negation scope. We use three different corpora for experiments, two existing ones and an additional one created by us from tweets. Our experiments demonstrate that our system outperforms the state-of-the-art system without negation, showing detecting negation improves emotion recognition. Further experiments show that our method for negation scope identification is also better than some existing popular ones.

1 Introduction

Automatic emotion recognition has a variety of useful applications. In social media domain, it can be used to determine how people feel about an emerging event, or a product they have bought. Emotion recognition can also be used to determine the emotional state of a customer when calling customer support, and help dialogue systems respond accordingly. Recent research shows it can even be used to determine the personality of people based on what they say in social media [Mohammad and Kiritchenko, 2015]. Besides, Ou *et al.* [2014] explores to use emotion information of an online community for domain specific event detection.

Emotion recognition has been a popular topic in NLP recently. Due to the lack of available language resources for emotion recognition, some researchers focus on emotion corpus creation [Aman and Szpakowicz, 2007; Strapparava and Mihalcea, 2008; Mohammad and Kiritchenko, 2015], while others work on emotion lexicon generation [Sintsova *et al.*, 2013; Mohammad and Turney, 2013; Staiano and Guerini, 2014]. Researchers also explore different methods for better emotion detection and classification [Bellegarda, 2010; Purver and Battersby, 2012; Wang, 2014; Jain and Sandu, 2015].

Negation plays an important role in emotion recognition. Most systems use emotion related lexical weights to help determine the emotion being portrayed. Without modeling negation, the system cannot recognize the real emotion expressed via pre-computed lexical weights. In this paper, we explore to model negation to help solve this issue for emotion recognition.

An important aspect of modeling negation is determining the negation scope. There are basically two types of methods, static methods and dynamic methods [Hogenboom *et al.*, 2011]. Window Size and Rest of the Sentence are static ones, while Next Non-Adverb and First Word with Lexical Weight are dynamic ones. These all use simple rules to determine the negation scope without considering semantic or syntactic information. Some researchers make use of dependency relation to determine the negation scope [Rosenberg and Bergler, 2012; Mohammad *et al.*, 2014; Reitan *et al.*, 2015]. Although this captures the syntactical relation between words, phrase information is ignored, which we believe is important for negation scope identification.

In this paper, we propose a negation scope detection method incorporating phrase relations and show that emotion recognition can be improved by considering negation information. This paper also contributes an emotion corpus and an emotion lexicon based on Twitter data. We consider six emotions, including joy, sadness, anger, surprise, fear, and disgust [Ekman, 1992]. Our method resembles work by Jia *et al.* [2009] and Council *et al.* [2010]. Council *et al.* [2010] manually create a corpus with negation information based on grammar phrases. They then explore to use conditional random field (CRF) for negation span detection. Jia *et al.* [2009] use a parse tree and get the least common ancestor following the negation cue. A manually constructed list of keywords is then used to determine the delimiter. Differently, we first use grammar rules to automatically detect the negation span and further determine the final scope by several simple phrase-based rules. We use tweet data for our experiments and show positive effect of modeling negation for emotion recognition. We also compare our generated emotion lexicon with an existing one for emotion classification, and experiment results show that using our lexicon improves system performance.

The rest of this paper is organized as follows. Section 2 introduces related work of emotion recognition and negation modeling for emotion recognition. Section 3 details on the data sets and lexicons used in this study. Section 4 presents our approach of modeling negation for emotion recognition in tweets. Experimental results are presented and analyzed in Section 5. Finally, Section 6 draws the conclusion.

2 Related Work

With emotion recognition becoming increasingly popular, there has been a high demand for emotion corpora and lexicons. Some researchers make use of the Mechanical Turk to create emotion annotated resources [Aman and Szpakowicz, 2007; Mohammad and Turney, 2013; Sintsova *et al.*, 2013]. Aman and Szpakowicz [2007] create an emotion corpus from web blogs. Mohammad and Turney [2013] create an emotion corpus from tweets. Sintsova *et al.* [2013] use the distribution of annotators' answers to compute weights for their emotion lexicon. Staiano and Guerini [2014] propose a method to create emotion lexicon in an automated way from the use of implicit crowd-sourcing with news articles. They obtain the weights of word-emotion pairs by multiplying a document-by-emotion matrix with a word-by-document matrix.

In this study we create our own emotion lexicon and corpus. Our work is most similar to Mohammad and Kiritchenko [2015]. They propose to use hashtags to generate the emotion corpus. They then create an emotion lexicon from this corpus using point-wise mutual information to calculate the strength of association of a word-emotion pair. Following their idea, we also create emotion corpus using hashtags. Compared to their corpus, ours is more balanced among different emotions (the class distributions are shown in later sections). For emotion lexicon generation, we use Latent Semantic Analysis (LSA) to calculate the weights of word-emotion pairs. Besides, our lexicon contains abbreviations and slang terms that are commonly used in social media.

Negation and its scope detection have been a popular topic in the area of NLP. Hogenboom *et al.* [2011] test four different existing methods for negation scope detection, including Window Size, Rest of Sentence, Next Non-Adverb and First Word with Lexical Weight. The first two are static methods, and the latter two are dynamic methods. Window Size specifies a fixed number of how many words to include in the negation scope following a negation cue. However, the negation scope is not always the same size in every context. Rest of Sentence assumes that all words following a negation cue until the end of the sentence are in the negation scope. This is a rather strong assumption and not always true. Next Non-Adverb determines that all words up to the next non-adverb following the negation cue are in the negation scope. First Word with Lexical Weight searches for the first word carrying a lexical weight to determine the scope. These four methods are not able to capture any relationship between words, which may cause incorrect negation scopes. Some researchers also use dependency relation to determine the negation scope [Rosenberg and Bergler, 2012; Mohammad *et al.*, 2014; Reitan *et al.*, 2015]. Mohammad *et al.* [2014] model negation by adding a feature determining how close an emotion word is to a negation cue based on the dependency relation. Reitan *et al.* [2015] use a CRF classifier to determine if the token is in the negation scope by using dependency relation as a feature as well. The limitation of using the dependency relation is that it does not capture the phrase relations between words which we believe is helpful for negation scope identification. We thus propose a method using the phrase relation between words to determine the scope of negation.

Emotion	Distribution %	Negation %
Joy	21.99	0.08
Anger	19.85	45.46
Sadness	16.39	41.04
Fear	10.63	25.02
Surprise	15.59	24.71
Disgust	15.52	26.67

Table 1: Distribution of Emotion Corpus

Joy	Loooove my life right now.
Anger	I can feel my blood boiling
Sadness	I haven't seen bae in a week :((
Fear	I am home alone, and I hear noises.
Surprise	It went from sunny to stormy real quick.
Disgust	Ew @ the girls not wearing any shoes in the gym

Table 2: Examples of tweets for each emotion

3 Creation of Emotion Corpus and Lexicon

3.1 Emotion Corpus

Although there is an existing emotion corpus from Mohammad *et al.* [2015], the samples among the six emotions are not quite balanced. We thus create our own corpus including tweets using the method described in Mohammad *et al.* [2015]. We polled the Twitter Search API[1] using emotion related hashtags, during May and August 2015. We collected 120K tweets. These are the hashtags used for each emotion:

- **Joy**: #joy, #happy, #pleasure, #amazing, #amuse, #delight
- **Anger**: #mad, #anger, #disappointed, #pissed, #irritated, #annoyed
- **Sadness**: #miserable, #unhappy, #despair, #sorrow, #sad, #depressed
- **Fear**: #terrified, #afraid, #frightened, #scary, #fear, #horror
- **Surprise**: #stunned, #amazed, #astonished, #surprise, #shock, #wow
- **Disgust**: #ew, #gross, #loathe, #nausea, #disgust, #distaste

Considering that hashtags in the middle of a tweet might not indicate its emotion, we only kept tweets with the hashtags of interest appearing at the end. We use the remaining 62,400 tweets to form our Emotion Corpus. Table 1 presents the basic statistics of our corpus. The negation percentage was calculated by the number of tweets with negation over the total number of tweets for that emotion. Observe that the distribution of tweets is quite balanced among the six emotions. We manually checked some of the tweets in the corpus, and felt that without the emotion related hashtags, we can still determine the emotion those tweets are conveying. This suggests that the reference labels in our corpus and the classification task are reasonable. Table 2 shows examples of tweets for each emotion from our Emotion Corpus.

[1] `http://www.tweepy.org/`

Joy	prizes, #summerday, #ootd
Anger	wrecked, yell, #bgc
Sadness	abort, loneliness, quizzes
Fear	change, hooks, bandits
Surprise	dm, #teenchoice, fannibals
Disgust	revolting, #ew, #nsfw

Table 3: Examples of lexicons for each emotion

3.2 Emotion Lexicon

Emotion lexicon is used for emotion recognition by providing the emotion related lexical weights for each word. There is an existing one for tweets, NRC Hashtag Emotion Lexicon [Mohammad and Kiritchenko, 2015]. However, that lexicon does not contain slang words or abbreviations. We believe such entries are important for emotion recognition in tweets because writing is always informal in social media. Hence, we create our own lexicon from a different and larger set of 300K tweets gathered between January and April of 2015. These tweets were gathered using the same method as for the Emotion Corpus construction. We calculate the Strength of Association (SoA) scores as the lexical weight for the unigrams and bigrams from the dataset with the six basic emotions. The SoA between an n-gram *w* and an emotion *e* is:

$$SoA(w, e) = LSA(w, e) - LSA(w, \neg e) \tag{1}$$

$$LSA(w, e) = \frac{\sum_{i=1}^{|ew|} cos\_sim(w, ew_i)}{|ew|} \tag{2}$$

$$LSA(w, \neg e) = \frac{\sum_{i=1}^{|\neg ew|} cos\_sim(w, \neg ew_i)}{|\neg ew|} \tag{3}$$

where LSA is Latent Semantic Analysis [Landauer *et al.*, 1998], *ew* is the list of six seed words for that emotion (refer to Section 3.1), and $\neg ew$ is the list of seed words for the other emotions. We compute the weight of an n-gram-emotion pair by computing the similarity between the n-gram and the seed words of the target emotion. For similarity computation, we create the n-gram-tweet matrix M (WxT), where W is the size of the vocabulary, T is the number of tweets, and the value in a cell (i,j) is the number of times n-gram *i* appears in tweet *j*. We apply singular value decomposition (SVD) on M to get its factorization with rank K approximation in the form UEV, where U is a WxK matrix, E is a KxK matrix and V is a KxT matrix. Then a n-gram with the original vector $\vec{w_i}$ can be represented as $\vec{w_i} * U$. Cosine similarity between the two vectors for the n-gram and the emotion seed word is then calculated. Note that we used the same number of seed words for all the emotions in order to avoid the domination of a specific emotion when calculating SoA for n-gram-emotion pairs. All the n-grams with a positive score are kept, which resulted in 32,677 unique n-grams for our emotion lexicon. Table 3 shows examples of lexicons for each emotion.

hardly	neither	nobody	not
cannot	lack	nor	havent
none	lacking	never	nothing
without	lacks	no	mustnt
nowhere	didnt	shouldnt	neednt
wasnt	darent	hadnt	doesnt
isnt	oughtnt	wouldnt	aint
dont	mightnt	shant	cant

Table 4: List of Negation words

4 Negation Detection and Emotion Recognition

4.1 Negation Detection

Since typically a negation cue negates the phrase following it, our method first identifies the negation cue and then determine the negation phrase. These are done using rules that we developed by inspecting the phrase structure of tweets and abstracting the general language patterns.

Given a tweet, we first obtain the POS tags by the CMU POS Tagger [Gimpel *et al.*, 2011], which has a reported accuracy of 90% on a Twitter corpus. Then we replace all the hashtags and emoticons with special symbols, so it will not confuse the parser. We detect a negation cue based on a manually generated list of negation words, as shown in Table 4, and a regular expression to catch words that end in *n't*. Once our system has detected a negation cue it will split the tweet into two parts, words before the negation cue and words after. We then pass the second part into the Standford Parser [Klein and Manning, 2003], which returns a set of phrases. We manually checked some parsed tweets. Although the parser is not perfect, we felt that the parsing results are reasonable to derive rules.

Our method for negation scope detection has two steps. First based on the POS tag of the word after the negation cue, we obtain phrase candidates by applying one of the following rules:

1) Noun/Determiner: The whole noun phrase is treated as candidate. For example, *That was not a sign*, the scope would be *a sign*.
2) Adjective: The adjective itself would be the candidate scope of negation. For example, *That was not amazing popcorn*, the scope would be *amazing*.
3) Adverb: There are two possible candidate scopes:
   (a) If it is an adverbial comparative, the adverbial phrase is the candidate negation scope. For example, *That movie was not very good*, the scope would be *very good*.
   (b) If it is not an adverbial comparative, then only the adverb itself is the candidate negation scope. For example, *The food was not quite as great as last time*, the scope would be *quite*.
4) Verb: The entire verb phrase is treated as a candidate. For example, *It did not run as long*, the scope would be *run as long*.

5) Preposition: The entire prepositional phrase is treated as a candidate. For example, *He is not at home*, the scope would be *at home*.

We observe that the candidate phrases generated by this first step are often larger than the ground-truth negation scope. Therefore in the second pass we apply the following rules to trim the phrase to obtain the final negation scope:

1) Find the first occurrence of a pronoun, preposition, proper noun and conjunction, if any, that exists in the negated phrase.
2) Segment the phrase based on what it contains:
   (a) If both a pro/proper noun and preposition occur, the preposition has more precedence so the phrase is cut at the preposition. For example, *dont think i can come back from this gaming experience #confused #farcry3 #sad*, the phrase is cut at *from*, even though *i* comes before *from*.
   (b) If a conjunction happens along with a pro/proper noun, preposition or both, the phrase is cut at the conjunction if it occurs first. For example, *don't eat out much but when i do its Chipotle*, the phrase is cut at *but* since it occurs before the pronoun *i*. If a pro/proper noun and preposition both occur before the conjunction, then the rule of precedence applies. For example, *never buy from over priced and it breaks only after a couple uses*, the phrase is cut at *from* instead of *and* since it occurs before the conjunction.
   (c) If the phrase is entirely pro/proper nouns and prepositions then nothing needs to be cut. For example, *not of christ* nothing is cut.
   (d) Remove all trailing hashtags that appear at the end since they are tags and not part of the phrase itself. For example, *can't go #sad #whyme*, the hashtags *#sad* and *#whyme* are removed.

The negation scope is used in the emotion recognition phase of the system.

4.2 Emotion Recognition Features and Model

We use SVM[2] for emotion recognition tasks. Some widely used features are implemented.

- **Word N-grams**: Presence of unigrams and bigrams.
- **POS**: The number of occurrences of each POS tag.
- **Elongated Words**: Number of words with one character repeating over two times.
- **Emoticons**: The number of emoticons that occur for each emotion.
- **Words per Emotion**: The number of words that occur per emotion. For each word, its emotion is the one with the highest lexical weight in (Eq. 1).
- **Emotion Score**: The weight of each emotion in the tweet. This is computed as the sum of the lexical weight of all the words in the tweet per emotion (Eq. 1).

[2] `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

Corpus	Label	Distribution %	Negation %
HEC	Joy	39.1	16.18
	Anger	7.4	34.47
	Sadness	18.2	38.04
	Fear	13.4	39.91
	Surprise	18.3	21.49
	Disgust	3.6	27.99
Blog Corpus	Emotion	34.3	23.12
	No Emotion	65.7	

Table 6: Distribution of Hashtag Emotion Corpus (HEC) and Blog Corpus

Once the negation scope is found, all of the lexical weights for each word in the scope are multiplied by negative one. A word will not be counted in the *Words per Emotion* feature if all the six lexical weights for the word are negative after the process of weight negation, but it will still be used to compute the *Emotion Score* feature.

5 Experiments and Results

In the experiment part, we evaluate the effectiveness of incorporating negation information for emotion recognition tasks. Besides, we also present experiments to compare the quality of different emotion lexicons. Finally, we compare our negation identification approach with others.

5.1 Experiments Setup

In our experiments we use three different data sets:

- *Our Emotion Corpus*: this is described in Section 3.1.
- *Hashtag Emotion Corpus* [Mohammad and Kiritchenko, 2015]: they create an emotion corpus from tweets based on emotion related hashtags. The corpus consists of 21,000 emotion tweets.
- *Blog Corpus* [Aman and Szpakowicz, 2007]: they collected web blog posts from the Web and used Mechanical Turk service to annotate each sentence based on whether the sentence contained emotion or not. The corpus has 4,266 sentences.

Table 6 presents the distribution of the other two corpora. As we can see, the distribution of the Hashtag Emotion Corpus is very skewed, whereas our Emotion Corpus has a much closer distribution among the six emotions. To explore the effectiveness of considering negation information on emotion recognition, we design these three tasks.

- *Task I Binary Emotion Classification*: We use one-vs-all classification for each emotion.
- *Task II Multi-class Emotion Classification*: We consider all the emotion classes during classification.
- *Task III Emotion Detection*: We detect if a sentence has emotion or not.

Task I and II are performed on our Emotion Corpus and the Hashtag Emotion Corpus, while task III is conducted on the Blog Corpus. 10-fold cross-validation is used. We report F1 score as the evaluation metric. For all the three tasks, we use a combined lexicon including both NRC lexicon and ours. If a

Task	Emotion	Our Emotion Corpus		Hashtag Emotion Corpus	
		Without Negation	Our Method	Without Negation	Our Method
I	Joy	87.84	**88.66**	71.47	**74.27**
	Anger	52.61	**57.51**	37.87	**41.67**
	Sadness	48.28	**52.84**	48.00	**54.54**
	Fear	60.04	**63.54**	59.82	**63.70**
	Surprise	62.41	**67.22**	53.24	**56.76**
	Disgust	54.73	**57.88**	30.48	**34.55**
II	All	63.76	**68.00**	61.63	**64.85**

Table 5: Scores of Emotion Classification using our Emotion Corpus and Hashtag Emotion Corpus (Task I. Binary Classification; Task II. Multi-Class Emotion Classification. **Bold**: The best performance per row for each corpus)

Task	Emotion	Without Negation	Our Method
III	Binary	70.97	**76.58**

Table 7: F1 Scores of Emotion Detection using the Blog Corpus (Task III. Emotion Detection. **Bold**: The best performance)

word is in both lexicons, we use the weights from our lexicon as its weights.

5.2 Results

For all the three tasks, we use the same set of features described in Section 4.2 for both our system and the one without negation. The only difference between the two systems is the value for *Emotion Score* and *Words per Emotion* features, which are affected by negation.

Table 5 shows the performance of different systems for task I and II using our Emotion Corpus and Hashtag Emotion Corpus. For our Emotion Corpus, we observe a significant improvement by our system over the one without negation. The improvement is the lowest for class joy because it has very few samples with negation (negation percentage is 0.08%). In general, the more samples with negation an emotion class contains, the higher the improvement made by our system over the one without negation. This indicates the effectiveness of modeling negation for emotion classification. Regarding the binary classification results, we can see that joy has the highest F1 score. This is possibly because words expressing joy overlap little with words related to other emotions. In contrast, the other emotions have less unique word choices for emotion expression, which affects the performance of classification. For example, *I wish I could've gotten into MIT on scholarship #jealous*, *jealous* could be both anger and sadness. We can see the same significant improvement over the system without negation for different emotions on the Hashtag Emotion Corpus. Similarly the improvement is higher for those emotions containing more negations. The F1 score for joy is the highest because it has most samples in the corpus. The F1 scores for anger and disgust are the lowest since the corpus contains the least amount of tweets that belong to these two emotions.

Table 7 shows the performance of different systems for task III using the Blog Corpus. A significant improvement is achieved by our system over the system without negation. This shows that our method can also work for web blogs and for the binary emotion detection task.

Task	Emotion	NRC-lex	our-lex	both-lex
I	Joy	82.86	85.64	**87.84**
	Anger	50.71	51.97	**52.61**
	Sadness	46.58	47.88	**48.28**
	Fear	56.93	58.97	**60.04**
	Surprise	58.59	60.30	**62.41**
	Disgust	51.29	52.12	**54.73**
II	All	59.43	61.06	**63.76**
III	Binary	68.89	69.54	**70.97**

Table 8: F1 Scores of Emotion Recognition based on different lexicons (Task I. Binary Emotion Classification; Task II. Multi-class Emotion Classification; Taks III. Emotion Detection. **Bold**: The best performance per row.)

Table 8 shows experiments comparing three lexicons for lexical weight computing for emotion recognition, including *NRC lex* (using NRC lexicon only), *our lex* (using our lexicon only) and *both lex* (using combined lexicon). The experiments were performed using the baseline system without modeling negation, in order to focus on the impact of three lexicons. For all the three tasks, *our-lex* performs better than *NRC-lex*, showing the effectiveness of our emotion lexicon. The best results are achieved by using the combined version of the lexicons.

Finally, we performed an experiment to see how our negation scope detection method compares to the existing methods, including Window Size, Rest of Sentence (ROS), Next Lexical Weight (NLW), Next Non-Adverb (NNA), End of Phrase (EOP), and dependency parse (DP). To test DP, we used negation features similar to Mohammad *et al.* [2014] and used TweeboParser [Kong *et al.*, 2014] to get the dependency relations of tweets. Table 10 shows the F1 scores for Task II and III when using different methods for negation scope detection. As expected, our method outperforms the static methods, Window Size and ROS, because our method is able to dynamically capture the negation scope. Our method per-

Method	Tweet
Window=3	I wish I could do the things I want to do and not [**what everyone else wants**] me to do.
Window=5	I can't [**believe**] they are giving him the next arthur ashe award #sad
ROS	man today is not [**a good day**] for me i dunno why my year keeps letting me down #upset #sad
NLW	Not [**sure**] who won the debate tonight, but somebody just ordered a closet full of new Pantsuits!
NNA	If you weren't [a **germaphobe**] before taking microbiology, you definitely are by the first test!
EOP	don't [**eat out much**] but when I do its Chipotle

Table 9: Examples of identified negation scope by our method and others. Underlined is what the method chooses as the negation scope. In bold is what our method chooses as the negation scope. Words within [] is the true negation scope

Method	Task II	Task III
Window=3	65.84	74.54
Window=5	65.91	73.95
ROS	56.78	73.27
NLW	61.25	73.89
NNA	60.85	71.86
EOP	66.03	74.88
DP	66.68	75.48
Our Method	**68.00**	**76.58**

Table 10: F1 Scores of Our Method vs Other Negation Scope Methods for emotion classification (Task II) and emotion detection (Task III).

forms better than NLW and NNA, since our method is able to determine the relationship between words. Our method outperforms the EOP method since the phrase can be longer than the negation scope and using grammar rules we are able to shorten it to the correct scope. Our method outperforms the DP method, since it uses the phrase relation between words which is important in determining the negation scope. Table 9 shows some examples of our method and others. It does not contain DP because it is usually used to create features instead of detecting scope directly. Our method is able to correctly identify the scope of negation over other methods, which either encapsulate too much or too little for the negation scope.

5.3 Error Analysis

We examined some misclassified tweets of our system and identified several causes for mistakes.

- When tweets are sarcastic or passive aggressive. For example, *Working late on a Monday #joy*, this tweet contains sarcasm as the tweeter is not actually happy.
- When a tweet is grammatically incorrect regardless of slang, abbreviations, emoticons or hashtags. This disrupts the phrase and negation scope identification aspect of our method. For example, *Without to play him? Are you joking me!*
- Our system considers emotions to be independent and only provide one prediction, yet tweets could portray more than one emotion. For example, *Why is life so awful* could portray both sadness and anger.
- When a tweet requires more information. For example, when one tweeter is responding to a question or statement from another tweeter, *@User, I agree that was poorly handled*. That tweet could portray anger, sadness, or disgust, and it is hard for the system to classify it without the previous knowledge of what the tweeter is referring to.

A common error in some systems is they do not account for when the negation cue *no* is the answer to a question. When this happens, *no* should be treated as a regular word and not a negation cue. However our system is able to identify when the negation cue *no* is the answer to a previous tweet. This is simply done by checking if the tweet starts with an *@User* followed by the negation cue *no*. We do not consider other negation cues like this as they are not commonly used as one word answers.

6 Conclusion

In this paper, we propose the use of grammar phrases and simple grammar rules as a method to determine the scope of negation and contribute a new emotion lexicon and corpus. We show that our method outperforms the system without negation for emotion classification and emotion detection. Our Emotion Lexicon outperforms an existing emotion lexicon, NRC-Lex. Our method is able to better dynamically encapsulate the negation with grammar phrases over existent static and dynamic models used for negation scope detection. Future research will include incorporating the detection of sarcasm and passive aggressiveness to determine their influence on emotion recognition.

Acknowledgments

We thank the anonymous reviewers for their detailed and insightful comments on this paper. The work is partially supported by DARPA Contract No. FA8750-13-2-0041 and AFOSR award No. FA9550-15-1-0346. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[Aman and Szpakowicz, 2007] Saima Aman and Stan Szpakowicz. Identifying expressions of emotion in text. In *Text, Speech and Dialogue*, pages 196–205. Springer, 2007.

[Bellegarda, 2010] Jerome R Bellegarda. Emotion analysis using latent affective folding and embedding. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 1–9. Association for Computational Linguistics, 2010.

[Councill *et al.*, 2010] Isaac G Councill, Ryan McDonald, and Leonid Velikovich. What's great and what's not: learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 51–59. Association for Computational Linguistics, 2010.

[Ekman, 1992] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[Gimpel *et al.*, 2011] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics, 2011.

[Hogenboom *et al.*, 2011] Alexander Hogenboom, Paul Van Iterson, Bas Heerschop, Flavius Frasincar, and Uzay Kaymak. Determining negation scope and strength in sentiment analysis. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 2589–2594. IEEE, 2011.

[Jain and Sandu, 2015] Er Ubeeka Jain and Amandeep Sandu. Emotion detection from punjabi text using hybrid support vector machine and maximum entropy algorithm. *Emotion*, 4(11), 2015.

[Jia *et al.*, 2009] Lifeng Jia, Clement Yu, and Weiyi Meng. The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1827–1830. ACM, 2009.

[Klein and Manning, 2003] Dan Klein and Christopher D Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

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

[Landauer *et al.*, 1998] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[Mohammad and Kiritchenko, 2015] Saif M Mohammad and Svetlana Kiritchenko. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326, 2015.

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

[Mohammad *et al.*, 2014] Saif Mohammad, Xiaodan Zhu, and Joel Martin. Semantic role labeling of emotions in tweets. *Proceedings of WASSA*, pages 32–41, 2014.

[Ou *et al.*, 2014] Gaoyan Ou, Wei Chen, Tengjiao Wang, Zhongyu Wei, Binyang LI, Dongqing Yang, and Kam-Fai Wong. Exploiting community emotion for microblog event detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1159–1168. Association for Computational Linguistics, 2014.

[Purver and Battersby, 2012] Matthew Purver and Stuart Battersby. Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics, 2012.

[Reitan *et al.*, 2015] Johan Reitan, Jørgen Faret, Björn Gambäck, and Lars Bungum. Negation scope detection for twitter sentiment analysis. In *6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis WASSA 2015*, page 99, 2015.

[Rosenberg and Bergler, 2012] Sabine Rosenberg and Sabine Bergler. Uconcordia: Clac negation focus detection at* sem 2012. 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 294–300. Association for Computational Linguistics, 2012.

[Sintsova *et al.*, 2013] Valentina Sintsova, Claudiu-Cristian Musat, and Pearl Pu Faltings. Fine-grained emotion recognition in olympic tweets based on human computation. In *4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2013.

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

[Strapparava and Mihalcea, 2008] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560. ACM, 2008.

[Wang, 2014] Zengfu Wang. Segment-based fine-grained emotion detection for Chinese text. *CLP 2014*, page 52, 2014.