HashCount at SemEval-2018 Task 3: Concatenative Featurization of Tweet and Hashtags for Irony Detection

Won Ik Cho, Woo Hyun Kang, Nam Soo Kim
Department of Electrical and Computer Engineering and INMC,
Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, Korea, 08826
{wicho, whkang}@hi.snu.ac.kr, nkim@snu.ac.kr

Abstract

This paper proposes a novel feature extraction process for SemEval task 3: *Irony detection in English tweets*. The proposed system incorporates a concatenative featurization of tweet and hashtags, which helps distinguishing between the irony-related and the other components. The system embeds tweets into a vector sequence with widely used pretrained word vectors, partially using a character embedding for the words that are out of vocabulary. Identification was performed with BiLSTM and CNN classifiers, achieving F1 score of 0.5939 (23/42) and 0.3925 (10/28) each for the binary and the multi-class case, respectively. The reliability of the proposed scheme was verified by analyzing the Gold test data, which demonstrates how hashtags can be taken into account when identifying various types of irony.

1 Introduction

Nowadays, opinion mining from social media has become an important issue in natural language processing (NLP). Since tweets are globally used social media text that can influence the worldwide readers with just a short arrangement of words, the analysis on tweets has been widely studied in the semantic aspects such as sentiment classification (Rosenthal et al., 2017), hate speech detection (Waseem and Hovy, 2016), and irony detection (Karoui et al., 2015; Liebrecht et al., 2013). Especially, the automatic detection of ironic tweets can help the readers who are having difficulty recognizing sarcasm to notice such figurative instances from excessive amount of text data.

Despite the potential usage of the tasks, irony and sarcasm are difficult to grasp simply by analyzing word distribution. They require understandings on the language and social context, which are dependent on time and space; this implies that the study should accompany constant updates on the database used for the analysis. Also, it is important to construct a concrete criteria set to distinguish between ironic and non-ironic tweets.

In this paper, we incorporate a classification on manually labeled irony tweet corpus. The corpus contains 2,396 English tweets for ironic/non-ironic each (4,792 total), which were annotated under the scheme suggested in Van Hee et al. (2016). For the competition, the corpus was split into a training (80%, or 3,834 instances) and test (20%, or 958 instances) set. A training procedure for the system was all performed with the former (constrained).

2 Task Description

Two tasks were proposed for ironic English tweet analysis. Task A deals with the binary classification involving ironic tweets and non-ironic ones, and in Task B the ironic ones are categorized into three types (i.e. *verbal irony by means of a polarity contrast*, *other types of verbal irony*, and *situational irony*) (Van Hee et al., 2018). The original corpus contains 1,728/267/401 instances for each type.

3 System Description

3.1 Feature Engineering

For feature extraction, significant user behaviors were taken into account. We observed several tendencies in tweets: a large portion of irony-relevant information is conveyed by hashtags, and the hashtagged words are usually out-of-vocabulary (OOV) or non-segmented; at the same time drawing attention of the readers and emphasizing the user's point.

While extracting the features, the importance of hashtagged information was reflected in the coextensive placement of the original tweet and hashtags. This connotes the idea that the hashtagged

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 546–552
New Orleans, Louisiana, June 5–6, 2018.

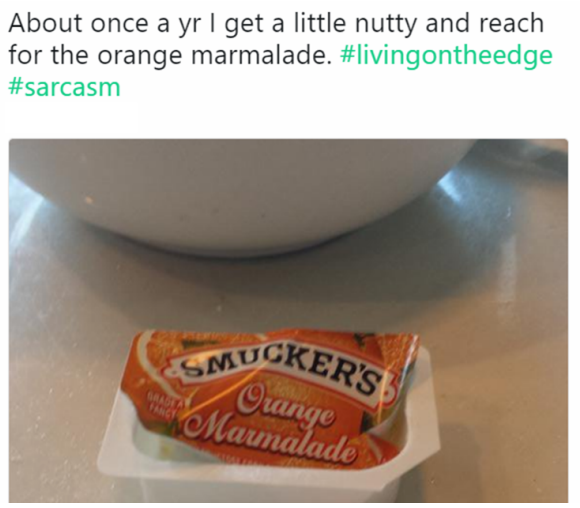

Figure 1: The capture of tweet number 63 in training set: *About once a yr I get a little nutty and reach for the orange marmalade. #livingontheedge http://t.co/sF9o6OWE1v* (ironic)

component is the metadata which needs highlighting.

3.1.1 Metadata handling

There are mainly three kinds of metadata observed in the tweets: ID tags (e.g. @someone), uniform resource locators (URLs, e.g. http://hyperlink), and hashtags (e.g. #something). The example sentences in this section are from the training set, not the Gold test data, thus they don't contain any irony hashtags, namely *#not*, *#irony*, and *#sarcasm*.

ID tags: In an empirical point of view, the ID tag of a tweet mainly delivered information on whether the user intends to notify someone the content (1a). This was not considered supportive information for irony detection, since sarcasm was observed to be conveyed in both a human-directive and non-human-directive way (1b,c) and that the presence of an ID tag ('@') could not be a crucial evidence in detecting irony.

(1) a. *@mrjamieeast I think it was the hotel owners...* (non-ironic)

b. *@LifeCheating doesn't lucky and fortunate mean the same thing?* (ironic)

c. *3 hours sleep yay loving life* (ironic)

URLs: The URLs are usually used as a hyperlink of the photo (Figure 1). However, as shown in the example, presence of photo doesn't necessarily affect the ironic nature of a tweet; we cannot infer the semantics of a photo just given an URL. Based on this observation, the URLs were not specially addressed; they were omitted in the word embedding process.

Hashtags: We paid attention to hashtagged words which were expected to contain information that can actually influence the nuance of the tweet (Chang, 2010). Thus, we created a placeholder for the vector embeddings of the hashtags (hashtag vector placeholder, HVP) and augmented it to the word vector sequence of the original tweet (Figure 2). This may cause a repetition on words in both the dictionary and the hashtags (e.g. *keep* of Figure 2), but it was assumed that the repetition would not degrade the performance. The reason for this assumption is because such words were expected to have a strong influence on the content, in that the user would have left it as a single word to notify its significance.

Firstly, a tweet was investigated whether it contains a hashtags. If not, the tweet would be compared to others based on the semantics of the original message. If any hashtag exists, the array of hashtagged words is embedded to the HVP to provide an useful evidence for the comparison with other tweets with hashtags. The numericalization of hashtags is demonstrated in the following passage.

3.1.2 Hashtag embedding

Given the nature of hashtags that does not allow spaces, most of the hashtagged words came under a category of either a single word (2a), concatenation of words with/without capital letters (2b,c), or an acronym (2d).

(2) a. *It will be impossible for me to be late if I start to dress up right now. #studing #university #lazy* (ironic)

b. *I picked a great week to start a new show on Netflix. #HellOnWheels* (ironic)

c. *Casper the friendly ghost on 2 levels! #thingsmichelleobamathinksareracist* (non-ironic)

d. *Another great day on the #TTC.* (ironic)

Instead of following the previous studies on hashtag segmentation (Bansal et al., 2015) that requires additional training processes, a simple guideline for hashtag embedding is proposed for such possible cases. Let *hash* be any hashtagged word, *dict* be a dictionary, $lower(w)$ be the character lowering process, and $D(w)$ be the vector

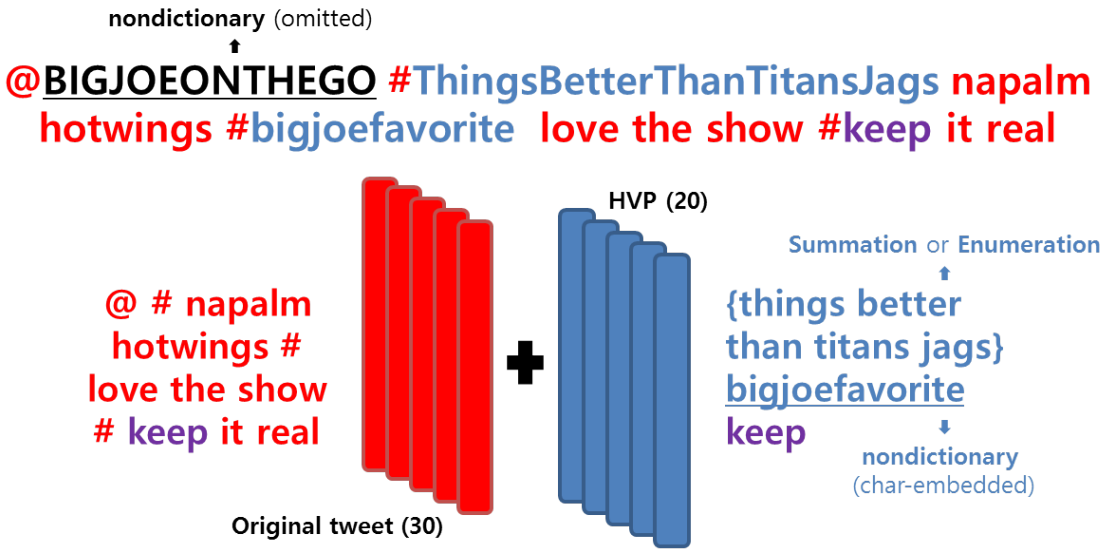

Figure 2: Feature extraction procedure for tweet number 1494. Word vector (or character-embedded vector) sequences are obtained for the original tweet and hashtags, denoted with length of 30 and 20 respectively. *keep* was colored purple since it belongs to both categories.

embedding for a word w.

- If $lower(hash)$ in *dict*: Trivial case. Compute output $O = D(lower(hash))$ and append it to the HVP.

- If $lower(hash)$ not in *dict* (OOV) and *hash* is a concatenation of words that each has a capitalized first letter:
  1) Split *hash* by the capital letters.
  2) One of the following strategies is chosen:
  a) **Summation** Compute the average $\sum_i D(lower(w_i))/W$ where w_i denotes each segmented word and W denotes the number of w_is. Assign it as an output O and append it to the HVP.
  b) **Enumeration** Append $D(lower(w_i))$ to the HVP one by one, for all w_is.

- Otherwise (character embedding): Compute $\sum_i D(lower(c_i))/C$ where c_i denotes each character in the *hash* and C denotes the number of c_is. Assign it as an output O and append it to the HVP.

The three dotted cases above can be respectively applied to the examples in Figure 2, namely *keep*, *ThingsBetterThanTitansJags*, and *bigjoefavorite*. To make it clear for the second case, for *ThingsBetterThanTitansJags*, the number of vectors appended to the HVP is 1 and 5 for the case of **Summation** and **Enumeration** respectively. Note again that only one heuristic method is chosen between those two - not being used simultaneously - in the aggregation of the word vectors from tweets.

The total size of the input feature is (*wdim*, 50), where *wdim* denotes the word vector size and 50 denotes the length of the vector sequence after the augmentation. The size was considered sufficient to cover the tokens of each tweet.

3.2 Classification

Classification was performed in a quite standard manner, utilizing both baseline sparse features and the proposed dense features. Although there have been some approaches that take into account the whole semantic relationship between terms (Kim et al., 2016; Cho et al., 2017), in this task we incorporated only the augmented feature constructed in the last section[1].

Since the sequential vectors were employed as features, descriptive models such as convolutional neural network (CNN) (Kim, 2014) and the bidirectional long short-term memory (BiLSTM) (Schuster and Paliwal, 1997) were adopted. In BiLSTM with an input size of 50 units, hidden layers had an output size of 32*2 = 64 units. In CNN, two 32*32-dim convolutional layers (for single channel) were used with a max-pooling layer between, summarized by a window of size 3. The output layer of the classifiers was fully connected

[1]Finding only the semantic contrasts could not cover the cases regarding situational irony.

F1 score	Task A	Task A (emoji)	Task B	Task B (emoji)
Baseline	0.6263	0.6200	0.3469	0.3455
CNN-text	0.6651	0.6425	0.3580	0.3647
CNN-enum	**0.6723**	0.6607	0.3785	0.4009
CNN-sum	0.6418	0.6543	0.4084	0.3517
BiLSTM-text	0.6568	0.6776	0.4022	0.3890
BiLSTM-enum	0.6701	0.6681	**0.4091**	**0.4378**
BiLSTM-sum	0.6616	**0.6852**	0.3843	0.3928

Table 1: 10-fold cross validation on the constrained training dataset. In task B, F1 score was obtained by macro averaging used in the scoring of competition. Bolded cases denote the best performing system for each labeled corpus. Underlined ones denote the systems that were originally submitted to the competition for each task. The score was updated according to an additional experiment.

	Accuracy	Precision	Recall	**F1 score**
Task A	0.6441	0.5426	0.6559	**0.5939**
(Best)	0.7972	0.7879	0.6688	0.7235
Task B	0.5982	0.4117	0.4096	**0.3925**
(Best)	0.7321	0.5768	0.5044	0.5074

Table 2: The submitted systems evaluated with the measurements, compared with the best scoring ones.

to the layer of 32 and then to decision layer of size 1 or 4, with either a sigmoid or a softmax activation, depending on the task.

4 Evaluation

The preprocessing was performed based on the example code[2], where the corpus was tokenized with NLTK (Bird et al., 2009). The 100dim GloVe (Pennington et al., 2014) pretrained with 27B token Twitter data[3] was employed as a dictionary for word embedding. All the training and validation procedure were carried out with Keras (TensorFlow backend) (Chollet et al., 2015). The code is provided online[4].

In this part, two main results are investigated. One deals with the comparison of performance according to classifiers and features. This is based on 10-fold cross validation using training data. In other words, the systems are trained using 3,450 instances and validated on 384 instances. In the other result, the competition score acquired using the submitted systems is inspected. Afterwards, we analyze the Gold test data, finding supportive evidences for the validity of the proposed approach regarding hashtags.

[2]https://github.com/Cyvhee/SemEval2018-Task3
[3]https://nlp.stanford.edu/projects/glove/
[4]https://github.com/warnikchow/HashCount

4.1 Result

The overall result tells that using the proposed features outperforms the baseline - term frequency-inverse document frequency (TF-IDF) on support vector machine (SVM), which was the system presented along with the train data. Table 1 shows the performance for each proposed system (*Classifier-Feature*) in each task. Here, *text* denotes the feature that includes only the original tweet (i.e. *Original tweet (30)* in Figure 2), and *enum/sum* denotes the feature that involves **Enumeration** and **Summation**, respectively.

The result suggests that employing *enum* and *sum* outperforms the vanilla case of *text* in general (Table 1). This supports our assumption in Section 3.1.1 and will be verified in Section 4.2. Although there were exceptions, employing BiLSTM outperformed the CNN-based method. This implies that the detection of irony is heavily influenced by specific terms (e.g. polarity items (Krifka, 1995)) or word sequence of the tweets. It is likely that the summarizing property of CNNs weaken the influence of such information. Also, unlike the intuition that more information will induce higher accuracy, there was no noticeable tendency regarding emoji.

4.2 Competition

The submitted systems showed F1 score of 0.5939 (ranked 23/42) and 0.3925 (ranked 10/28) in the competition, for Task A and B respectively.

Task A: The system submitted for Task A equals to the best performer of Table 1, *BiLSTM-sum* with emoji. However, it showed little an improvement compared to the CNN-based method (*CNN-enum* without emoji);%. it implies that the binary classification is less influenced by specific

Instances	Total	With @	With #	Correct	With # (>1)	Correct with # (>1)
Non-ironic	473	217 (45.8%)	255	301 (63.6%)	164 (34.6%)	125 (76.2% given #>1)
Ironic	311	84 (27.0%)	311	204 (65.5%)	139 (44.6%)	97 (69.7% given #>1)

Table 3: Analysis with the Gold test data of Task A. @ and # each denotes ID tag and hashtag.

Instances	Total	Correct	With # (>1)	Correct with # (>1)
Non-ironic	473	339 (71.6%)	164	134 (81.7%)
Polar	164	108 (65.8%)	74	51 (68.9%)
Situational	85	21 (24.7%)	40	10 (25.0%)
Other	62	1 (1.65%)	25	0 (0.0%)

Table 4: Analysis with the Gold test data of Task B.

terms or word sequence. From this, it could be inferred that the type classification of irony itself can be more difficult than just a detection.

Task B: The system submitted for task B, namely *CNN-sum* without emoji, differs from the lately-achieved best performer (*BiLSTM-enum* with emoji) which relatively outperforms the submitted model by 7.19%. The initial choice is based on overlooking the superiority of using BiLSTM, where the influence of the specific terms on nuance of the tweet has not been recognized.

There was a significant decrease in recall, compared with Task A. Since the feature is fixed, there may exist possible issues of classifier and emojis. Considering the analysis in the last subsection, it is presumed that the classifier issue is dominant over other problems. CNN seems to be a more cautious model than BiLSTM, but this is not considered a good property for the detector. In the aspect of the nature of sarcasm that at least 'sensitive' false alarms are better than generous ignorance, it is considered a reasonable decision to use a recurrent classifier model with high recall.

4.3 Gold Test Data

Further investigation regarding the Gold test data was performed for both tasks. We examined the composition of correct predictions, in terms of how the presence of hashtags positively affected the outcome.

Task A: We first figured out how ID tags and hashtags are distributed over the whole test corpus. It was revealed that non-ironic tweets more frequently accompany ID tags than the ironic ones (Table 3), but we did not conclude that ID tags convey a non-ironicalness for the reasons given in 3.1.1.

For hashtags, there was an important thing to consider in counting; all ironic tweets included at least one irony hashtag, which led to 100% presence of '#' for ironic instances. Thus, only the 'effective' hashtags were taken into account in a way of identifying tweets with more than one '#'; in other words, a single default hashtag was ignored[5].

In terms of tweet ratios with effective hashtags, ironic tweets outperformed the non-ironic ones. In addition, the rate of correct prediction in instances with #>1 was observed to be higher than the ratio in the whole instance, for both cases ($63.6\% < 76.2\%$ and $65.5\% < 69.7\%$ in Table 3). These validates the utility of the proposed modeling which emphasizes hashtags as indispensable metadata.

We also observed that 79, 41, and 19 instances of ironic tweets possessed one, two, and three or more effective hashtags, respectively. Except for 172 tweets with a single default hashtag, the most common case is to convey sarcasm in one word as in (3), inducing a contradiction between the literal evaluation and the intended one (Van Hee et al., 2016).

(3) *Just a @ScienceDaily article re: a robot arm you can control with your mind. Meh. Nothing huge. #sarcasm #science http://t.co/AK4bAorhBc*

Task B: We further investigated the detailed types of ironic tweets. Except for the case of *Other* where the accuracy was particularly low, hashtags performed as supportive information for correct prediction (Table 4).

Of the three irony types, *Polar* recorded the highest accuracy of correct answer prediction, also showing the most amount of enhancement given a condition of at least one effective hashtag. To be more specific than what was discussed above,

[5]This was also applied to non-ironic cases to match the proportion of instances.

Number of effective #s	0	1	2	>2
Polar	57/90 (63.3%)	27/42 (64.2%)	18/21 (85.7%)	6/11 (54.5%)
Situational	11/45 (24.4%)	6/26 (23.1%)	2/9 (22.2%)	2/5 (40.0%)
Other	1/37 (2.7%)	0/11 (0.0%)	0/11 (0.0%)	0/3 (0.0%)

Table 5: The number of effective instances for each irony types (correct/total) regarding the number of hashtags. The irony hashtags were not counted.

the number of effective hashtags was found to be most supportive in one or two cases (Table 5). This property can be effectively utilized, e.g. varying the weight given to the hashtag vector, depending on the number.

For *Situational*, the percentage was not reliable due to the shortage of the number of instances. We concluded that the case regarding the tweets only with default irony tags (11/45) is rather valid, in that the tone of the *Situational* ironic tweets appeared to be descrptive than provocative (4a). It was even more difficult to identify tendencies for the case of *Other* (4b); thus, it was assumed to resemble the case of *situational*.

(4) a. *The thirstiest of thirst buckets calling other people thirsty #irony*

b. *@BarryBlackNE I don't think the Hereditary Baronet wants to encourage a something-for-nothing culture :-$ #irony*

5 Discussion

There are a few more schemes to consider in the future implementation. First, Additional normalization of words can be done. The proposed scheme focused on the featurization of the original tweet and hashtags, and no lemmatization or stemming was carried out. This was mainly because normalization can inadvertently erase important information; but it would be tolerable if carried out just on the non-hashtagged lexical words.

It would also be effective to apply the advanced segmentation algorithm to hashtags, taking into account the improved performance of proposed systems. This does not necessarily involve an algorithm that requires additional training and a huge database. Lighter and fancy segmentation techniques that fully utilize the dictionary are expected to be introduced.

Finally, fusion of classifiers can be undertaken. This was not considered in the proposed system, since in mixed networks, it is difficult to recognize the influence of using each feature and classifier. Nonetheless, such networks can be chosen in terms of boosting performance for real-life applications.

6 Conclusion

In this paper, the feature engineering based on the coextensive placement of tweet and hashtags was presented. Two embedding schemes for hashtag vector placeholder (HVP) were employed, and concatenation of HVP and original word vector sequences was used as inputs to CNN and BiLSTM classifiers. The implementation verified that the proposed systems outperform the baseline, and the system's reliability was supported by analyzing the correlation of hashtags and prediction accuracy with the Gold test data. Future works include lemmatization that do not affect content, development of an efficient hashtag segmentation, and fusion of classifiers.

Acknowledgments

This work was supported by the Technology Innovation Program (10076583, Development of free-running speech recognition technologies for embedded robot system) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea).

References

Piyush Bansal, Romil Bansal, and Vasudeva Varma. 2015. Towards deep semantic analysis of hashtags. In *European Conference on Information Retrieval*. Springer, pages 453–464.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Hsia-Ching Chang. 2010. A new perspective on twitter hashtag use: Diffusion of innovation theory. *Proceedings of the Association for Information Science and Technology* 47(1):1–4.

Won Ik Cho, Woo Hyun Kang, Hyun Seung Lee, and Nam Soo Kim. 2017. Detecting oxymoron in a single statement. In *Proceedings of Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*. pages 48–52.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Jihen Karoui, Farah Benamara, Véronique Moriceau, Nathalie Aussenac-Gilles, and Lamia Hadrich Belguith. 2015. Towards a contextual pragmatic model to detect irony in tweets. In *53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. pages PP–644.

Joo-Kyung Kim, Marie-Catherine de Marneffe, and Eric Fosler-Lussier. 2016. Adjusting word embeddings with semantic intensity orders. In *Proceedings of the 1st Workshop on Representation Learning for NLP*. pages 62–69.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* https://arxiv.org/abs/1408.5882.

Manfred Krifka. 1995. The semantics and pragmatics of polarity items. *Linguistic analysis* 25(3-4):209–257.

Christine Liebrecht, Florian Kunneman, and Antal van Den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not pages 29–37.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pages 502–518.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2016. Guidelines for annotating irony in social media text, ver. 2.0. *LT3 Technical Report Series* .

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, New Orleans, LA, USA, SemEval-2018.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*. pages 88–93.