KELabTeam: A Statistical Approach on Figurative Language Sentiment Analysis in Twitter

Hoang Long Nguyen, Trung Duc Nguyen and Dosam Hwang
Department of Computer Engineering
Yeungnam University, Korea
{longnh238, duc.nguyentrung, dosamhwang}@gmail.com

Jason J. Jung*
Department of Computer Engineering
Chung-Ang University, Korea
j2jung@gmail.com

Abstract

In this paper, we propose a new statistical method for sentiment analysis of figurative language within short texts collected from Twitter (called tweets) as a part of SemEval-2015 Task 11. Particularly, the proposed model focuses on classifying the tweets into three categories (i.e., sarcastic, ironic, and metaphorical tweet) by extracting two main features (i.e., term features and emotion patterns). Our experiments have been conducted with two datasets, which are Trial set (1000 tweets) and Test set (4000 tweets). Performance is evaluated by cosine similarity to gold annotations. Using this evaluation methodology, the proposed method achieves 0.74 on the Trial set. On the Test set, we achieve 0.90 on sarcastic tweets and 0.89 on ironic tweets.

1 Introduction

Sentiment analysis in computer science is a difficult task which aims to identify the emotion from a given data source. The goal of sentiment analysis is to dissect a given document and determine whether its opinion represent positive, negative, or neutral. There have been many studies (which use lexicon-based methods and machine learning-based methods) to extract and identify the sentiment (Medhat et al., 2014). In case of figurative language, the task becomes more challenging because the document can have secondary or extended meanings. Hence, exactly finding the truth meaning of figurative language is an interesting problem for researchers due to its importance.

*Corresponding author

The first work that we want to mention here is contributed by Reyes and Rosso (2013a). The authors captured ironic sentences from low-level to high-level of irony according to three conceptual layers and their eight textual features. With customer reviews on Amazon, Reyes and Rosso (2012a) contributed an approach for distinguishing irony and non-irony based on six models. Also focusing on detecting irony, Hao and Veale (2010) classifies irony and non-irony by analyzing the large quantity of simile forms with 9-steps sequence. By considering short texts with case-study is Twitter, Reyes et al. (2013b) introduced a model to detect verbal irony by combining four types of conceptual features and their dimensions. Focusing on comprehending metaphor, Shutova et al. (2010) used unsupervised methods to find the associate from a small set of metaphorical expressions by verb and noun clustering processing to detect similarity structure of metaphor. Finally, Reyes et al. (2012b) analyzed humor and irony by adding more features to express the favorable and unfavorable ironic contexts using the theory of textual.

These above studies tried to solve the problem by focusing on lexical level. Therefore, the goal of our research is to find a new way to identify figurative meaning. In this work, we focus on analyzing three types of figurative languages (i.e., sarcasm, irony, and metaphor) on tweets collected from Twitter. With FLASA Model (Figurative Language Analysis using Statistical Approach) to detect multiple types of figurative language, we believe that this is a general model to solve the problem and easy-extending for characterizing other types.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 679–683,
Denver, Colorado, June 4-5, 2015.

2 System Description

The Training set includes 8000 tweets collected from Twitter. All the tweets are presented in English with three main types of tweets: sarcasm, irony, and metaphor with the respective ratio: 5000 sarcastic tweets, 1000 ironic tweets and 2000 metaphorical tweets.

$$Z = \{< t, s > \mid s \in [-5, 5]\} \quad (1)$$

where Z is a set of tweets in the Training set; t is a tweet, and s is the score of that tweet.

Tweets are extracted into the set of terms. All the tweets are pre-processed by: i) considering in lower-case mode, ii) removing unnecessary information such as: the tagged persons, pronouns, iii) formalizing words (e.g., remove redundancy characters which repeat more than three times, and correct the typos). The hash-tags and symbol in the tweets are kept because of the sentiment expressing property. The set of terms which is extracted from Z:

$$T_Z = \bigcup_{i=1}^{n} t_i = \bigcup_{i=1}^{n} \{w_j \mid w_j \in t_i\}_{j=1}^{m} \quad (2)$$

where T_Z is a set of terms that are extracted from Z; n is the number of tweets in the Training set; w_j is a term; and m is the number of terms that are extracted from Z.

2.1 FLASA Model

FLASA Model includes two main modules which are: i) Content-based Approach Module, and ii) Emotion Pattern-based Approach Module. The final score of a tweet is calculated by using the following formula:

$$S = \alpha \times SC + \beta \times SE \quad (3)$$

where S is the final score of a tweet; SC is the score that is calculated by Content-based Approach module; SE is the score that is calculated by Emotion Pattern-based Approach Module; and α, and β are coefficients identified based on the training error score of the classification model of each approach, with $\alpha + \beta = 1$.

2.1.1 Content-based Approach Module

Content-based approach module evaluate the sentiment of a tweet based on the co-occurrence of terms which are extracted from a tweet using the Training set. This method basically use statistics on the Training set to predict the score of a tweet.

With a tweet t_k that is needed to be annotated. First, it is extracted into set of terms:

$$T_k = \bigcup \{w_i \mid w_i \in t_k\}_{i=1}^{m_k} \quad (4)$$

where T_k is the set of terms extracted from tweet t_k; w_i is a term belongs to tweet t_k; and m_k is the number of terms which are extracted from tweet t_k.

From T_k, we build all the possible combinations from the set of terms to consider all the possible co-occurrence of terms because terms can express different meaning when they appear together. With this step, we can achieve all these aspects: i) all the meaning of the tweet t_k when terms co-exist, and ii) some main terms that affect the score of the tweet t_k. We can consider each of combination is a cluster which can respective as a feature vector:

$$C_k = \left\{ (\delta_k)_{i=1}^{\gamma_k} \middle| \gamma_k = \sum_{j=1}^{m_k} \binom{m_k}{j} \right\} \quad (5)$$

where C_k is the set of all possible clusters extract from the given tweet; δ_k is a cluster, each cluster can be represented as a feature vector; and γ_k is the number of all combinations which are created from terms in T_k.

Each cluster in C_k is represented as a feature vector, with the dimension equals with the number of terms in T_k. From the set of tweets Z in the Training set, we cluster every tweet into the set of cluster C_k. A tweet is assigned into a cluster in the case: i) the distance between a vector to a cluster is minimum comparing to its distance to other clusters, and ii) the distance has to smaller than a defined threshold. This has a significant meaning in expressing the co-occurrence of terms in a tweet. The distance between a tweet and a cluster is calculated by using the following formula:

$$dis(A, B) = 1 - \frac{A^T B}{|A||B|} \quad (6)$$

where $dis(A, B)$ is the distance between a term and a cluster.

Each cluster has a cluster coefficient which is calculated from the number of feature terms of a cluster. If a cluster has more terms, its coefficient will

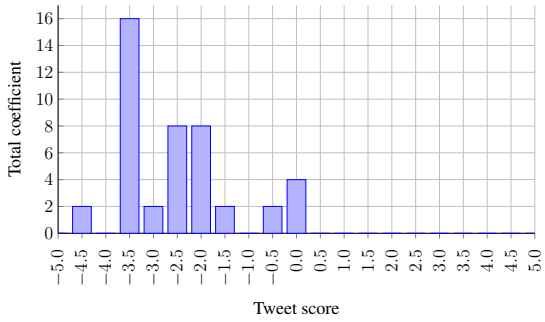

Figure 1: Histogram of score distribution.

be higher. The cluster coefficient can expresses how important it affects the final score of a tweet. Then, from tweets in clusters with their scores and coefficient, the histogram is built to represent the distribution of score in the Training set. Finally, the score of a tweet is annotated by selecting the peak of the histogram.

Example 1. We have 3 clusters: cluster $\{A, B, C\}$: includes 3 tweets ($< t_1, -2.5 >$; $< t_2, -3.5 >$; $< t_3, -3.5 >$); cluster $\{B, C\}$: includes 3 tweets ($< t_4, 0.0 >$; $< t_5, -2.0 >$; $< t_6, -2.0 >$); cluster $\{C\}$: includes 4 tweets ($< t_7, -4.5 >$; $< t_8, -3.0 >$; $< t_9, -0.5 >$; $< t_{10}, -1.5 >$). Figure 1 expresses the above data as histogram. In this case, the score of tweet which is calculated by Content-based Approach Module is -3.5.

2.1.2 Emotion Pattern-based Approach Module

The Emotion Pattern-based Approach Module determine the score of a given tweet based on the emotion change pattern in the content. This approach consists in calculating the sentiment score for each term, then construct the emotion distribution pattern using the termss score in the tweet corresponding to its occurrence positions.

Each term has a score which is calculated based on tweets in the Training set. By finding the score of term and the pattern of tweet, we can understand about how important a term contributes to the final score of a tweet, and about the sentiment degree of a term, whether it's positive, negative, or neutral. The score of a tweet is decided by the pattern of terms in a sentence. Our goal is try to find the real score of a term. In the Traning set, a term belongs to many tweets, and in each tweet, it represents a different score. Assuming that all the tweets have equatable meaning, the score of a term is calculated by the following formula:

$$S_w = \frac{\sum_{i=1}^{l} S_{w_i}}{l} \tag{7}$$

where S_w is the score of a term; and l is the number of tweets which contain this term.

From the set of tweets Z and the set of terms T, we can find the distribution of a term by using the score of tweets which contain it. The peak of histogram is the point at that a term has highest distribution with a score. At the beginning (i^0 step), each term has the score which is selected from the peak of its respective histogram. Then, the score in the step $i+1$ is calculated by using the formula:

$$S_w^i = \frac{S_w^{i-1} * P(S_t|w)}{\sum_{j=1}^{n}(S_{w_j}^{i-1} * P(S_t|w_j))} * S_t \tag{8}$$

where S_w^i is the score of a term at step i_{th}; S_t is the score of tweet that contains this term; and $P(S_t|w)$ is the probability that a term has the score with given tweet score.

This step is conducted repeatably until the score of term at step i^{th} greater than the score of term at step $(i-1)^{th}$ a value of defined epsilon, with epsilon is extremely small.

With each tweet in the Training set, it is extracted into the list of terms and then create a pattern based on its term scores as we mentioned above. Due to the different of the number of terms in a tweet, the signal of pattern is needed to be scaled by using an interpolation function. The pattern is scaled to the maximum possible terms that a tweet in the Training set contain in order to be able to map all the tweets into vectors with same dimension.

Example 2. We have a tweet: *@SamySamson wow you're soooo funny #sarcasm it actually hurts a bunch!*. From this tweet, we have list of terms and their scores: ($< wow, -0.2057831 >$; $< soo, -0.1552674 >$; $< funny, -0.19274 >$; $< \#sarcasm, -2.34994 >$; $< actually, -0.03287 >$; $< hurts, -0.16091 >$; $< bunch, -0.02096 >$). Figure 2 expresses the pattern of the above data after the term scores are scaled down by the size of largest terms in a tweet

found in the Training set. Here, the maximum number of terms that a tweet contains in the Training set is 24.

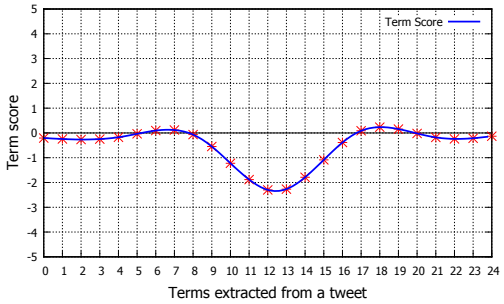

Figure 2: Sequential pattern of tweet term scores after length normalization.

Using the set of patterns from the Training set, we construct a vector space representation whereby each dimension signifies a match to one of the extracted patterns. We then train a decision tree based classifier to predict from these vectors the integer sentiment labels $[-5..5]$ of the corresponding tweets. And that is the score which is annotated by using Emotion Pattern-based Approach Module.

3 Experimental results

The test data comprises 4000 tweets with both figurative and non-figurative tweets with 70% of them are sarcasm, irony, or metaphor; and 30% of the data are other. We evaluate the test with: i) Content-based Approach Module, ii) Emotion Pattern-based Approach Module, and iii) Combined Module.

FLASA Model works well with figurative tweets. Using cosine similarity to gold annotations to evaluate the system, the highest performance that we got is 0.90 with irony type, and the next is sarcastic type with 0.89. With metaphor type, we achieve 0.34 with annotated tweets. About non-figurative tweets, the performance is still low due to the tweets in the Training set. The root cause is that there are no non-figurative tweets in the Training set. If we add more non-figurative tweets to the Training set in order to learn, the result will be improved. Fig. 3 shows the performance that we got from testing our approach on the Test set.

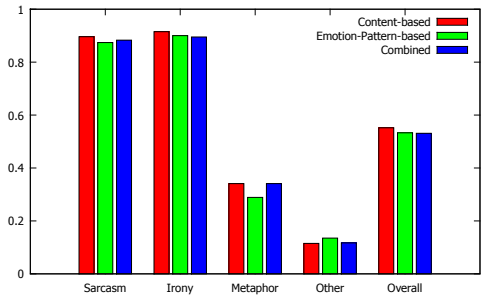

Figure 3: The performance of FLASA Model on Test set using cosine similarity.

4 Conclusion

In this paper, we proposed a new approach for analyzing the sentiment of figurative language based-on the statistics with two main approaches: content and emotional pattern. By combining all these features, we enhanced the performance of our algorithm. However, the result of FLASA Model is affected by these following reasons:

i) Almost all the tweets in the Training set are sarcastic tweets, and irony tweets. Due to this reason, the performance on metaphor tweets, and non-figurative tweets are still low.

ii) Is this work, we only consider unigram model when calculating the score for terms in Emotion Pattern-based Approach. This leads to the miss-expressing meaning of terms if they are co-showing an specific sense in a phrase.

iii) Our training data has a little noise because some tweets are written in an unstandardized way(e.g. abbreviation word, and repeated word).

In the next work, we will improve the performance by increasing the number of tweets in the Training set, especially the metaphor tweets, and non-figurative tweets. Bigram or trigram model will be used to clearly comprehend the sentiment of a tweet. Moreover, we will add more heuristic to completely formalize tweets. Finally, we will extend FLASA Model to analyze the data from the other social network, such as Facebook, Instagram, Flick, and Google Plus also.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant

funded by the Korea government (MSIP) (NRF-2014R1A2A2A05007154). Also, this research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Cetner) support program (NIPA-2014-H0301-14-1044) supervised by the NIPA (National ICT Industry Promotion Agency).

References

Hao, Y., & Veale, T. 2010. *An Ironic Fist in a Velvet Glove: Creative Mis-Representation in the Construction of Ironic Similes*. Minds and Machines, 20(4), 635-650.

Kaur, A., & Gupta, V. 2013. *A Survey on Sentiment Analysis and Opinion Mining Techniques*. Ain Journal of Emerging Technologies in Web Intelligence, 5(4), 367-371.

Kumon-Nakamura, S., Glucksberg, S., & Brown, M. 1995. *How about Another Piece of Pie: The Allusional Pretense Theory of Discourse Irony*. Journal of Experimental Psychology General, 124(1), 3-21.

Medhat, W., Hassan, A., & Korashy, H. 2014. *Sentiment Analysis Algorithms and Applications: A Survey*. Ain Shams Engineering Journal, 5(4), 1093-1113.

Reyes, A., & Rosso, P. 2013a. *On the Difficulty of Automatically Detecting Irony: Beyond a Simple Case of Negation*. Knowledge and Information Systems, 40(3), 595-614.

Reyes, A., Rosso, P., & Veale, T. 2013b. *A Multidimensional Approach for Detecting Irony in Twitter*. Languages Resources and Evaluation, 47(1), 239-268.

Reyes, A., & Rosso, P. 2012a. *Making Objective Decisions from Subjective Data: Detecting Irony in Customers Reviews*. Journal on Decision Support Systems, 53(4), 754-760.

Reyes, A., Rosso, P., & Buscaldi, D. 2012b. *From Humor Recognition to Irony Detection: The Figurative Language of Social Media*. Data & Knowledge Engineering, 74(0), 1-12.

Shutova, E., Sun, L., & Korhonen, A. 2010. *Metaphor Identification Using Verb and Noun Clustering*. Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Beijing, China, August 23-27, pp. 1002-1010.