I2RNTU at SemEval-2016 Task 4: Classifier Fusion for Polarity Classification in Twitter

Zhengchen Zhang[1], **Chen Zhang**[2], **Fuxiang Wu**[3],
Dong-Yan Huang[1], **Weisi Lin**[2], **Minghui Dong**[1]
[1]Human Language Technology Department, Institute for Infocomm Research (I2R),
A*STAR, Singapore, 138632
[2]School of Computer Engineering, Nanyang Technological University,
50 Nanyang Avenue Singapore, 639798
[3]BeiHang University, Beijing, China, 100191
huang@i2r.a-star.edu.sg

Abstract

In this work, we apply classifier fusion to tweet polarity identification problem. The task is to predict whether the emotion hidden in a tweet is positive, neutral, or negative. An asymmetric SIMPLS (ASIMPLS) based classifier, which was proved to be able to identify the minority class well in imbalanced classification problems, is implemented. Word embedding is also employed as a new feature. For each word, we obtain three word embedding vectors on positive, neutral, and negative tweet sets respectively. These vectors are used as features in the ASIMPLS classifier. Another three state-of-the-art systems are implemented also, and these four systems are fused together to further boost the performance. The fusion system achieved 59.63% accuracy on the 2016 test set of SemEval2016 Task 4, Subtask A.

1 Introduction

The I2RNTU system works on the Subtask A: Message Polarity Classification in Twitter of SemEval-2016 Task 4: the Sentiment Analysis in Twitter (Nakov et al., 2016). The task is to predict whether a tweet is of positive, neutral, or negative sentiment. This task can be formulated as a multi-class classification problem, i.e., to classify a tweet into one of the three classes. We use the one-vs-rest strategy to solve the three-class classification problem. Given a tweet, a classifier generates three confidence scores about the tweet belonging to the three classes respectively. The predicted label is chosen based on the highest confidence score. Four classifiers are implemented in our work, and classifier fusion is used to improve the system performance.

Classifier fusion has been proved to be very powerful in classification problems. In SemEval-2015, a system named Webis won the first place in the message polarity classification subtask, which is subtask B of SemEval-2015 task 10 "Sentiment Analysis in Twitter" (Hagen et al., 2015). The authors reproduced four state-of-the-art twitter polarity prediction algorithms. Each algorithm generates three confidence scores for a tweet. The fusion system averages the scores generated by the four classifiers, and then predicts a label according to the highest average score. In the Speaker State Challenge of INTERSPEECH 2011, the method of fusing Asymmetric SIMPLS (ASIMPLS) and Support Vector Machines (SVMs) won the sleepiness sub-challenge (Huang et al., 2011). The asymmetric SIMPLS based classifier is shown to be able to generate a higher prediction accuracy for the class with small number of instances in the imbalanced classification problem, while SVMs are strong at predicting the class with majority number of instances. The fusion of these two types of methods could achieve a balance between favouring the majority class and the minority class. In the Music Information Retrieval Evaluation eXchange (MIREX 2013), the method of fusing SIMPLS and SuperFlux won the 3rd place on Audio Onset Detection subtask (Zhang et al., 2013). In the Emotion Recognition in the Wild Challenge (EmotiW2014), which aims to automatically classify the emotions acted by human subjects in video clips under real-world environment, the fusion of kernel SVM, logistic regression, and partial least squares (PLS) with different Riemannian ker-

Proceedings of SemEval-2016, pages 71–78,
San Diego, California, June 16-17, 2016.

nels won the first place of the competition (Liu et al., 2014). In this paper, we will take advantage of the classifier fusion again. We introduce the asymmetric SIMPLS based classifier to the tweet polarity classification problem, and combine it with other three of state-of-the-art classifiers. The fusion method is same with (Hagen et al., 2015).

The most popular features used in the tweet polarity classification problem are derived from sentiment lexicons (Rosenthal et al., 2015). Word embedding represents a word using a low dimensional vector which contains the syntactic and semantic meaning of the word. If we can enhance the sentiment information hidden in the vector, it may be a good feature for the task. Word embedding has been used in tweet sentiment analysis in (Zhang et al., 2015). The authors used the vectors with dimensionality of 300 trained by word2vec, which is publicly available on-line [1]. However, the vectors are trained using Google News. No emotional info was considered during the training. Also, the news articles are written in formal language. Many words appearing frequently in tweets such as 'goood' may not be included in the data set. In this paper, we train the word embedding on downloaded tweet data sets. Furthermore, we separate the tweet data set into three subsets named positive, neutral, and negative subsets respectively. For each word, three vectors are obtained. These vectors are used as features for the ASIMPLS classifier.

The papers of task description of SemEval (Rosenthal et al., 2015; Rosenthal et al., 2014; Nakov et al., 2013) may be the best material of understanding the related work of sentiment analysis in twitter. The classifiers used include SVM, maximum entropy, Conditional Random Fields (CRFs), deep neural networks, and linear regression etc. The most popular features are derived from sentiment lexicons. Bag-of-words, hashtags, and punctuations etc. are also used widely.

We will introduce the asymmetric SIMPLS classifier and the proposed word embedding feature in Section 2. The three state-of-the-art systems and the fusion method are described in Section 3. The experimental results are shown in Section 4. We conclude our work in Section 5.

[1] https://code.google.com/archive/p/word2vec/

2 The Asymmetric SIMPLS (ASIMPLS) based classifier

2.1 The ASIMPLS algorithm

Partial Least Squares (PLS) has been introduced into classification problems in (Huang et al., 2011). SIMPLS is an efficient algorithm for PLS regression that calculates the PLS factors as linear combinations of the original variables (De Jong, 1993). Given two matrices $\mathbf{X} \in \mathbb{R}^{N\times M}$ with N samples and M dimensional features, and a label matrix $\mathbf{Y} \in \mathbb{R}^{N\times K}$. The SIMPLS algorithm aims to find a linear projection (De Jong, 1993)

$$\widehat{\mathbf{Y}} = \mathbf{XB} \tag{1}$$

For the bianry classification problems, $K = 1$. The solution is to extract the orthogonal factors of $\mathbf{X}$ and $\mathbf{Y}$ sequentially,

$$\mathbf{t}_a = \mathbf{X}_0\mathbf{r}_a \tag{2}$$

and

$$\mathbf{u}_a = \mathbf{Y}_0\mathbf{q}_a, a = 1, 2, ..., A \tag{3}$$

where $\mathbf{X}_0 = \mathbf{X} - \mathrm{mean}(\mathbf{X})$, $\mathbf{Y}_0 = \mathbf{Y} - \mathrm{mean}(\mathbf{Y})$, and $A \leq M$. The algorithm of extracting the parameters is shown in Algorithm 1 (De Jong, 1993; Huang et al., 2011; Zhang et al., 2013). It can be seen as the training algorithm of the ASIMPLS based classifier.

In the SIMPLS algorithm, a constrain is added that the scores $\mathbf{t}_i$ are orthogonal to each other, i.e., $\mathbf{t}_b'\mathbf{t}_a = 0$ for $a > b$. Also $\mathbf{t}_a$ is normalized by $\mathbf{t}_a = \mathbf{t}_a/\sqrt{\mathbf{t}_a' * \mathbf{t}_a}$. Then we have $\mathbf{T}'\mathbf{T} = \mathbf{I}$ where $\mathbf{T} = [\mathbf{t}_1, ..., \mathbf{t}_A]$. Hence,

$$\widehat{\mathbf{Y}}_0 = \mathbf{T}\mathbf{T}'\mathbf{Y}_0 = \mathbf{X}_0\mathbf{R}\mathbf{R}'\mathbf{X}_0'\mathbf{Y}_0 = \mathbf{X}_0\mathbf{R}\mathbf{R}'\mathbf{S}_0 \tag{4}$$

We can write $\mathbf{B}$ in (1) as:

$$\mathbf{B} = \mathbf{R}(\mathbf{R}'\mathbf{S}_0) = \mathbf{R}(\mathbf{T}'\mathbf{Y}_0) = \mathbf{R}\mathbf{Q}' \tag{5}$$

For a new feature matrix $\mathbf{X}^*$, the new projected matrix

$$\widehat{\mathbf{Y}}^* = \mathbf{X}_0^*\mathbf{B} \tag{6}$$

where $\mathbf{X}_0^* = \mathbf{X}^* - \mathrm{mean}(\mathbf{X}^*)$. The prediction can be written in another format (De Jong, 1993):

$$\mathbf{Y} = \sum_{i=1}^{A} \mathbf{b}_i\mathbf{t}_i^*\mathbf{q}_i' \tag{7}$$

Algorithm 1 Training procedure of SIMPLS

Input: Feature set $\mathbf{X}$, Label $\mathbf{y}$, and Number of components A

Variables: Projection matrix $\mathbf{R}$, score vectors $\mathbf{T}$ and $\mathbf{U}$, loading $\mathbf{P}$ and $\mathbf{Q}$

$\mathbf{R} = []; \mathbf{V} = []; \mathbf{Q} = []; \mathbf{T} = []; \mathbf{U} = [];$
$\mathbf{y} = [y_1, y_2, ..., y_N]'; \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]'$
$\mathbf{y}_0 = \mathbf{y} - \text{mean}(\mathbf{y}); \mathbf{X}_0 = \mathbf{X} - \text{mean}(\mathbf{X});$
$\mathbf{S} = \mathbf{X}_0' \mathbf{y}_0$
for $i = 1$ to A **do**
  $\mathbf{q}_i =$ dominant eigenvectors of $\mathbf{S}'\mathbf{S}$
  $\mathbf{r}_i = \mathbf{S} * \mathbf{q}_i$
  $\mathbf{t}_i = \mathbf{X}_0 * \mathbf{r}_i$
  $\text{normt}_i = \text{SQRT}(\mathbf{t}_i' \mathbf{t}_i)$
  $\mathbf{t}_i = \mathbf{t}_i / \text{normt}_i$
  $\mathbf{r}_i = \mathbf{r}_i / \text{normt}_i$
  $\mathbf{p}_i = \mathbf{X}_0' * \mathbf{t}_i$
  $\mathbf{q}_i = \mathbf{y}_0^T * \mathbf{t}_i$
  $\mathbf{u}_i = \mathbf{y}_0 * \mathbf{q}_i$
  $\mathbf{v}_i = \mathbf{p}_i$
  if $i > 1$ **then**
    $\mathbf{v}_i = \mathbf{v}_i - \mathbf{V} * (\mathbf{V}' * \mathbf{p}_i)$
    $\mathbf{u}_i = \mathbf{u}_i - \mathbf{T} * (\mathbf{T}' * \mathbf{u}_i)$
  end if
  $\mathbf{v}_i = \mathbf{v}_i / \mathbf{SQRT}(\mathbf{v}_i' * \mathbf{v}_i)$
  $\mathbf{S} = \mathbf{S} - \mathbf{v}_i * (\mathbf{v}_i' * \mathbf{S})$
  $\mathbf{r}_i, \mathbf{t}_i, \mathbf{p}_i, \mathbf{q}_i, \mathbf{u}_i,$ and $\mathbf{v}_i$ into $\mathbf{R}, \mathbf{T}, \mathbf{P}, \mathbf{Q}, \mathbf{U}$, and $\mathbf{V}$, respectively.
end for
$\mathbf{B} = \mathbf{R} * \mathbf{Q}'$

Algorithm 2 To obtain new $\mathbf{T}$ in testing

Input: New feature matrix $\mathbf{X}$; projection matrix $\mathbf{R}$.

Output: New $\mathbf{T} = [\mathbf{t}_1^*, \mathbf{t}_2^*, ..., \mathbf{t}_A^*]$

$\mathbf{X}_0 = \mathbf{X} - \text{mean}(\mathbf{X})$
for $i = 1$ to A **do**
  $\mathbf{t}_i^* = \mathbf{X}_{i-1} \mathbf{r}_i;$
  $\mathbf{X}_i = \mathbf{X}_{i-1} - \mathbf{t}_i^* (\mathbf{t}_i^{*'} \mathbf{X}_{i-1}) / (\mathbf{t}_i^{*'} \mathbf{t}_i^*);$
end for

where $\mathbf{b}_a = \mathbf{u}_a' \mathbf{t}_a / (\mathbf{t}_a' \mathbf{t}_a)$. The new $\mathbf{t}^*$ is calculated by Algorithm 2. For the classification problems, we need to predict the label of a sample

$$\hat{\mathbf{Y}} = \text{sign}(\sum_{i=1}^{A} \mathbf{b}_i \mathbf{t}_i^* \mathbf{q}_i') \quad (8)$$

$$= \text{sign}(\sum_{i=1}^{A} \mathbf{m}_i \mathbf{t}_i^*) \quad (9)$$

$$= \text{sign}(\mathbf{m} \cdot \mathbf{t}^*) \quad (10)$$

It can be observed that the label $\widehat{\mathbf{Y}}$ is a function of the score vectors $\mathbf{t}^*$. If $\hat{\mathbf{Y}}$ and $\mathbf{t}$ were on a plane, the boundary would be a line passes the original point $(0, 0)$. Suppose the original point is the center of the whole data set, which can be achieved by subtracting the mean of the data. When the number of instances of a class is much less than the other one, the original point will be far away from the center of the minority class, while near the center of the majority class. Hence, the line will pass cross the majority class. That is the reason that PLS based classifiers can detect the minority class well. However, the accuracy of majority class will decrease because the line cuts the majority class into two parts. The Asymmetric PLS classifier tries to move the line towards to the center of the minority class to make the boundary in the middle of the two classes (Huang et al., 2014). The distance moved is calculated on the first dimension of $\mathbf{T}$. The center points and the radii of positive and negative classes are estimated as in (11). Let the minority class be the positive class, and $index\_postive$ denotes the index of positive items in $\mathbf{Y}$. We use $\mathbf{t}_p = \mathbf{t}_1[index\_postive]$. Similarly, $\mathbf{t}_n = \mathbf{t}_1[index\_negative]$. The center points and the radii of the two classes are estimated by

$$\begin{aligned} rad_p &= std(\mathbf{t}_p) \\ rad_n &= std(\mathbf{t}_n) \\ cp_p &= mean(\mathbf{t}_p) \\ cp_n &= mean(\mathbf{t}_n) \end{aligned} \quad (11)$$

Then the distance should be moved is

$$distance = cp_p - (cp_p - cp_n) * \frac{rad_p}{rad_p + rad_n}$$

We move the line on the plane of $\mathbf{t}$

$$\hat{\mathbf{Y}} = \text{sign}(\sum_{i=1}^{A} \mathbf{m}_i \mathbf{t}_i^*) - \mathbf{m}_1 * distance \quad (12)$$

2.2 Features used in the ASIMPLS classifier

The features used in the ASIMPLS system are mainly adopted from the NRC-Canada (Mohammad et al., 2013) system. We will describe the features in detail in Section 3.1. In addition, we propose a type of word embedding based feature named emotional word embedding.

We assume that vectors obtained on pure positive/negative tweets are different from those obtained on the general text because the distribution of words are different. Hence, we collected three tweet sets on-line: positive, neutral, and negative. The twitter4j library [2] is used to collect the tweets. We collected a set of happy emoticons such as :), :D, :-), :o), :], :c), =], 8), =), :}, as well as a set of unhappy emoticons like :(, :-(, :c, :c, :[, :[, :{. We use them as keywords, and search tweets containing these emoticons. The tweets containing a happy emoticon are put into the positive set. Similarly, a tweet containing an unhappy emoticon is put into the negative set. For the neutral tweets, we search those containing keywords like sports, news, etc. This is a very simple rule to collect data sets. Noise will be introduced, and more filtering work shall be done in the future. Finally, we collect about 69 million positive tweets, 19 million neutral tweets, and 19 million negative tweets. The tweets are pre-processed by the CMU tweet NLP tools (Gimpel et al., 2011). We remove the @somebody tags and the hyper-links in the tweets. Three vectors are trained on the three data sets respectively for each word. The vectors are obtained using the open-source toolkit word2vec with negative sampler 10, window width 5, and vector dimension 100. The feature of a tweet is the mean value of the vectors of every word. These 300 dimensional features are put together with the other features to train the ASIMPLS classifier.

3 Reproduced Systems

With reference to the state-of-the-art system designed by team Webis (Hagen et al., 2015), we had also incorporated three models out of four (those developed by team NRC-Canada, team GU-MLT-LT and team KLUE) into our own system. As proved by the outstanding performance of team Webis in SemEval-2015, each of the three systems employed a unique set of features that could complement each other and would help enhance the performance in the three-point scale tweets sentiment classification. Thus, we decide to keep most of the same set of features for each reimplemented model and use the same classifier, L2-regularized logistic regression, for the three reimplemented systems. In detail, we use the weka.classifiers.functions.LibLINEAR class. The SVM Type is set to be 0, and the Cost is set to be 0.05. The other parameters use default values. The preprocessing steps for all three implementations are similar, which involves converting all letters to lower case and removing all the URLs and user names. Each system is described shortly in the following subsections.

3.1 NRC-Canada

In the reimplementation of the model developed by team NRC-Canada (Mohammad et al., 2013), all preprocessed tweets are tokenized and POS-tagged with the Twitter NLP tool developed by Carnegie Mellon University (Gimpel et al., 2011). The model leverages a rich set of features. We had kept all the features. Firstly, **word N-grams** and **character N-grams** are used. Word N-grams include the existence of one to four contiguous sequences of tokens and non-contiguous ones. Character N-grams include the existence of three, four and five consecutive sequences of characters. Secondly, the number of words with **all capitalized letters** and the number of **hashtags** are included in the feature set. Thirdly, the feature set contains the number of times each **part-of-speech tag** occurred. Next, **punctuation marks** and **emoticons** are also part of the features, because the number of consecutive sequences of exclamation/question marks and the polarity of an emoticon in the tweets would help determine the overall sentiment. In addition, the number of **elongated words**, such as "youuuuuu", and the number of **negations** are employed. As specified in (Pang et al., 2002), the negated context is part of a tweet that begins with a negation word, such as "not", and ends with a punctuation mark. With negation, the

[2] http://twitter4j.org/en/index.html

sentiment expressed by a token will be reversed. We attach each token in a negated context with a suffix "NEG". Furthermore, with the Brown clustering method (Brown et al., 1992), 56,345,753 tweets by Owoputi (Owoputi et al., 2013) have been clustered into 1000 clusters. If the tokens belonging to these clusters were present in the tweets, these clusters would be included in the feature set. Lastly, three manually crafted and two automatically generated polarity dictionaries are used, i.e., the **NRC Emotion Lexicon** (Mohammad and Turney, 2010; Mohammad and Turney, 2013), **the MPQA Lexicon** (Wilson et al., 2005), **the Bing Liu Lexicon** (Hu and Liu, 2004), **Hashtag Sentiment Lexicon** and **sentiment140 Lexicon**.

3.2 GU-MLT-LT

For the system designed by team GU-MLT-LT (Günther and Furrer, 2013), the tokenization process is slightly different from that of NRC-Canada system. Besides tokenizing the original raw tweets, the letters of tweets are lowercased and these normalized tweets are tokenized. In addition, for all the elongated words shown in the aforementioned normalized tweets, such as "youuuu", repetitions of letters after the first one are removed. A new version of further normalized tweets are obtained and then tokenized. Team GU-MLT-LT employed a smaller set of features as compared to that of team NRC-Canada. Our team keeps some of the features original used by GU-MLT-LT in our reimplementation. Firstly, **unigrams and bigrams** are used. For bigrams, stop word tokens, such as "the", and punctuation tokens, such as ".", are removed. In addition, following the Porter stemmer algorithm (Porter, 1980), **the word stems** of the normalized tokens are included in the feature set. Moreover, the polarity dictionary used is **the SentiWordNet** (Baccianella et al., 2010). Similar to that of team NRC-Canada, **clustering and negation** are also employed in the feature set.

3.3 KLUE

The tokenization process in the reimplementation of system designed by team KLUE (Proisl et al., 2013) is similar to that of team NRC-Canada. The features used by KLUE are quite different from those used by the other two teams. In order to complement the features of the other two systems, we keep most of the features used by team KLUE in our reimplementation. Firstly, **unigram and bigrams** are considered and their frequencies of occurrence serve as feature weights. In order to be counted as part of the feature set, the unigrams and bigrams should be present in at least five tweets. Secondly, **the total number of tokens per tweet** is also incorporated in the feature set. Furthermore, the polarity dictionary chosen is **AFINN-111 lexicon** (Nielsen, 2011) which contains a variety of words of degree from -5(very negative) to +5(very positive). The dictionary is used to extract features including **the frequencies of occurrence of positive and negative tokens**, **the total number of tokens that expressed one sentiment** as well as **the arithmetic mean of total sentiment scores per tweet**. Moreover, for **emoticons**, we adopt the manually crafted dictionary and its polarity scoring from Webis team and apply them in this reimplementation. Lastly, **negation** is considered for up to next three tokens or less for the case that the tweet ends within three tokens after the negated word and the sentiment scores for tokens up to a distance of at most 4 following the negated marker are reversed.

3.4 The fusion method

The four classifiers generate three confident scores $s^i_{pos}, s^i_{neu}, s^i_{neg}, i = 1, 2, 3, 4$ for each tweet respectively. We fuse the systems by summing these scores with a same weight. The final scores are $[\frac{1}{4}\sum_{i=1}^{4} s^i_{pos}, \frac{1}{4}\sum_{i=1}^{4} s^i_{neu}, \frac{1}{4}\sum_{i=1}^{4} s^i_{neg}]$. The predicted label is the index with the maximum score value.

We find that ASIMPLS does not perform well if the dimension of the features is much larger than the number of training samples. Unfortunately the features used in this task have more than 200 thousand dimensions, while we only have about 16 thousands training samples. Hence, the classifier fusion is used again. We split the features into 5 parts, e.g., the first 40 thousand dimensions of the features are used to train a ASIMPLS classifier. The second 40 thousand dimensions are used to train another one etc. The confidence scores of the 5 ASIMPLS classifiers are averaged to generate a new set of confidence scores. These scores are the output of the ASIMPLS system. We combine them with scores generated by other

	Training Set	Testing Set
2013	11338	3813
2014	0	1853
2016	5348	1704+1781
Total	16686	9151
After Preprocessing	16682	9146
Positive	6956	4153
Neutral	7166	3579
Negative	2560	1414

Table 1: Training and testing sets.

three systems as discussed above. For the ASIMPLS classifier, feature selection is an alternative way of improving system performance instead of classifier fusion, which will be tried in future work.

4 Experimental Results

Our training set is composed by the training and development set of SemEval-2013 as well as the training set of SemEval-2016. The testing set includes the development-test set of 2013 and 2014, as well as the development and development-test sets of 2016. The numbers of tweets in all data sets are shown in Table 1. We remove the tweets that only have @somebody or hyper-links. Hence, we have a total of 16682 tweets for training and 9146 tweets for testing after preprocessing. In both training and testing sets, there are much less number of negative tweets comparing to positive and neutral tweets. ASIMPLS may help improve the accuracy of negative tweets.

The experimental results are shown in Table 2. We list all the results of individual systems and the fusion results. In the table, the "Score" is the value obtained using the evaluation method of SemEval2016 Task 4 (Nakov et al., 2016). "Positive" means the accuracy of positive samples in the testing, i.e. $Positive = \frac{number\ of\ correct\ positive\ samples}{number\ of\ all\ positive\ samples}$. Similarly, "Negative" and "Neutral" denote the accuracies of negative as well as neutral samples respectively. "All Accuracy" means $\frac{number\ of\ all\ correct\ samples}{number\ of\ all\ testing\ samples}$. The results demonstrated that classifier fusion is able to generate better scores than individual classifiers. Fusing all systems obtained the best score 63.78. It is marginally higher than the score of fusing the first three systems 63.72. The score of the PLS system is worse than the other three systems. The reason may be that the dimension of the features (more than 200 thousands) is even bigger than the number of training samples (about 16 thousands). The ASIMPLS classifier is not good at handling this type of data even we have used 5 subsystems to alleviate the influence of high dimensional features. Nonetheless, it obtained the highest accuracy of negative samples and the second-best accuracy of neutral samples. The positive accuracy is much worse than the other three classifiers. It indicates that the bias should be further adjusted in the future. The results of our submission are shown in Table 3. We obtained a score of 59.63 on the 2016 Tweet submission.

5 Conclusion

In this paper, we have applied classifier fusion to the sentiment analysis in twitter task. An ASIMPLS based classifier has been implemented, and has been combined with other three state-of-the-art methods. A new feature named emotional word embedding has been introduced, and has been used in the ASIMPLS based method. Experimental results demonstrated that the fusion is able to improve the system performance because it can combine the strengths of different classifiers. The ASIMPLS obtained a good minority class accuracy and a bad accuracy for the majority class. How to adjust the bias to improve the balance between these two classes is the future work.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Sijmen De Jong. 1993. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. 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*

	NRC	GU-MLT-LT	KLUE	Fuse 3 methods	ASIMPLS	Fuse All
Score	61.48	62.30	61.55	63.72	57.49	**63.78**
Positive	71.18	**77.25**	70.29	74.57	56.85	74.52
Neutral	**53.93**	40.32	47.95	50.99	52.78	51.10
Negative	57.57	66.12	69.80	63.65	**75.60**	63.86
All Accuracy	62.32	61.08	61.47	63.66	58.16	**63.71**

Table 2: System performance.

2013		2014			2015	2016
Tweet	SMS	Tweet	Tweet sarcasm	Live Journal	Tweet	Tweet
69.27	59.66	68.02	46.93	69.63	63.80	59.63

Table 3: Submission results.

Technologies: short papers-Volume 2, pages 42–47. Association for Computational Linguistics.

Tobias Günther and Lenz Furrer. 2013. Gu-mlt-lt: Sentiment analysis of short messages using linguistic features and stochastic gradient descent.

Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. 2015. Webis: An ensemble for twitter sentiment detection. *SemEval-2015*, page 582.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Dong-yan Huang, Shuzhi Sam Ge, and Zhengchen Zhang. 2011. Speaker state classification based on fusion of asymmetric simpls and support vector machines. In *Twelfth Annual Conference of the International Speech Communication Association*.

Dong-Yan Huang, Zhengchen Zhang, and Shuzhi Sam Ge. 2014. Speaker state classification based on fusion of asymmetric simple partial least squares (simpls) and support vector machines. *Computer Speech & Language*, 28(2):392–419.

Mengyi Liu, Ruiping Wang, Shaoxin Li, Shiguang Shan, Zhiwu Huang, and Xilin Chen. 2014. Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 494–501. ACM.

Saif M Mohammad and Peter D Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics.

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

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.

Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter.

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 2016)*, San Diego, California, June. Association for Computational Linguistics.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Thomas Proisl, Paul Greiner, Stefan Evert, and Besim Kabashi. 2013. Klue: Simple and robust methods for polarity classification. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 395–401.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 73–80.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. *Proceedings of SemEval-2015*.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Zhengchen Zhang, Dong-yan Huang, Renbo Zhao, and Minghui Dong. 2013. Onset detection based on fusion of simpls and superflux. *Music Information Retrieval Evaluation eXchange (MIREX 2013)*.

Zhihua Zhang, Guoshun Wu, and Man Lan. 2015. Ecnu: Multi-level sentiment analysis on twitter using traditional linguistic features and word embedding features. *SemEval-2015*, page 561.