Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification

Duyu Tang\textsuperscript{1}, Furu Wei\textsuperscript{2}, Nan Yang\textsuperscript{2}, Ming Zhou\textsuperscript{2}, Ting Liu\textsuperscript{1}, Bing Qin\textsuperscript{1}

\textsuperscript{1} Harbin Institute of Technology
\textsuperscript{2} Microsoft Research
Roadmap

• Motivation

• The Proposed Method

• Experiments

• Conclusion
Twitter sentiment classification

- Input: A tweet
- Output: Sentiment polarity of the tweet
  - Positive / Negative / Neutral
Top-system in SemEval-2013 Task 2(B)

• *NRC-Canada* [Mohammad 2013]
  – Feature engineering
    • Hand-crafted features
    • Sentiment lexicons

  – How about learning feature automatically from data for Twitter sentiment classification?
Word Representation (Embedding)

• Word embedding is important
  – Compositionality
  – [Yessenalina11; Socher13]

• Word Embedding

\[ \text{linguistic} = \begin{pmatrix} 1.045 \\ 0.912 \\ -0.894 \\ -1.053 \\ 0.459 \end{pmatrix} \]
Is It Enough for Sentiment Analysis?

- Existing embedding learning algorithms typically use the syntactic contexts of words.

```
he formed the **good** habit of ...
he formed the **bad** habit of ...
```

---

The words with similar contexts but *opposite* sentiment polarity are mapped into close vectors.
Roadmap

• Motivation

• The Proposed Method

• Experiments

• Conclusion
Twitter Sentiment Classification

Input

Embedding Layer

Layer $f$

Layer $f'$

Training Data

Learning Algorithm

Sentiment Classifier

average

max

min

concatenate

i don’t wanna miss it :)

Collobert and Weston (C&W) 2011

\[
\text{Loss Function} \quad \max(0, 1 - f^c_{cw}(t) + f^c_{cw}(t^r))
\]

\[
\text{HardTanh} \quad y = \begin{cases} 
-1 & \text{if } x < -1 \\
 x & \text{if } -1 \leq x \leq 1 \\
1 & \text{if } x > 1 
\end{cases}
\]

\[
Y_j = W_{ij} \times X_i + b_j
\]

Text: it is so coooll :)
Collobert and Weston (C&W) 2011

**Loss Function**

\[ \max(0, 1 - f^{cw}(t) + f^{cw}(t^r)) \]

**HardTanh**

\[ \frac{\partial y}{\partial x} = \begin{cases} 
0 & \text{if } x < -1 \\
1 & \text{if } -1 \leq x \leq 1 \\
0 & \text{if } x > 1 
\end{cases} \]

**Linear**

\[ \frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial Y_j} \times \frac{\partial Y_j}{\partial W_{ij}} = \frac{\partial L}{\partial Y_j} \times X_i \]

\[ \frac{\partial L}{\partial X_i} = \sum_j \frac{\partial L}{\partial Y_j} \times \frac{\partial Y_j}{\partial X_i} = \sum_j \frac{\partial L}{\partial Y_j} \times W_{ij} \]

\[ \frac{\partial L}{\partial b_j} = \frac{\partial L}{\partial Y_j} \times \frac{\partial Y_j}{\partial b_j} = \frac{\partial L}{\partial Y_j} \times 1 \]
Model 1: SSWE Hard

• Intuition
  – Use the sentiment polarity of sentences (e.g. tweets) to learn the sentiment-specific word embedding (SSWE)

  – Solution
    • Predict the sentiment polarity of text

\[
\begin{align*}
\text{Positive} & \quad \rightarrow \quad \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
\text{Negative} & \quad \rightarrow \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\end{align*}
\]
Loss Function

\[- \sum_{k=\{0,1\}} f^g_k(t) \cdot \log(f^h_k(t))\]

Gold Distribution

Predicted Distribution

Softmax

\[Y_i = \frac{\exp(X_i)}{Z} \quad Z = \sum_{i'} \exp(X_{i'})\]
Model 2: SSWE Soft

• Intuition
  – Use the **sentiment polarity of sentences** to learn the sentiment-specific word embedding

  – Solution
    • Soften the hard constrains of Model 1

![Positive](image1.png)  ➔  ![Negative](image2.png)

|      | Model 1 | Model 2 |
|------|---------|---------|
| **P** | [1, 0]  | [1.7, 0.2] |
| **N** | [0, 1]  | [0.3, 3.8] |

\[ P > N \]
\[ P < N \]
Input Window

Text: *it is so coool :)*

Linear

\[ M^2 \times \circ \]

HardTanh

\[ \int \]

Linear

\[ M^1 \times \circ \]

Lookup Table

\[ LT_w \]

concatenate

\[ c = 2 \]

\[ H \]

\[ H \]

\[ D \]

Loss Function

\[
\max(0, 1 - \delta_s(t) f^r_0(t) + \delta_s(t) f^r_1(t))
\]

Positive Score

Negative Score

Indicator Function

\[
\delta_s(t) = \begin{cases} 
1 & \text{if } f^g(t) = [1, 0] \\
-1 & \text{if } f^g(t) = [0, 1]
\end{cases}
\]
Model 3: SSWE Unified

• Intuition
  – Use both the syntactic contexts of words and the sentiment polarity of sentences to learn the sentiment-specific word embedding

– Solution
  • A hybrid approach by capturing both information

he formed the good habit of
Input Window

Text \textit{it is so coooll :)}

Linear

$M^2 \times \odot$

HardTanh

$\int$

Linear

$M^1 \times \odot$

Lookup Table

$LT_w$

concatenate

$c = 2$

$H$

$D$

Loss Function

\[
\alpha \cdot \text{loss}_{cw}(t, t^r) + (1 - \alpha) \cdot \text{loss}_{us}(t, t^r)
\]

Syntactic Loss

Sentiment Loss

Sentiment Loss

\[
\max(0, 1 - \delta_s(t) f_1(t) + \delta_s(t) f_1(t^r))
\]
Embedding Training

• Data
  – Tweets contains positive/negative emoticons
    
    | Positive | :| ) | :- | :D | = |
    | Negative | :( | : ( | :- ( | |

  – 5M positive, 5M negative tweets from April, 2013

• Back-propagation + AdaGrad [Duchi 2011]
  – Embedding length = 50
  – Window size = 3
  – Learning rate = 0.1

Hu et al., 2013
Roadmap

• Motivation

• The Proposed Method

• Experiments
  – Twitter Sentiment Classification
  – Word Similarity of Sentiment Lexicons

• Conclusion
Twitter Sentiment Classification

• Setting
  – Data
    • Twitter Sentiment Classification Track in Semantic Evaluation 2013 (message-level)
    • **Positive VS negative** classification

  – Evaluation metric
    • Macro-F1 of positive VS negative
Results

• Comparison with Different Embeddings
## Results

- **Comparison with Twitter Sentiment Classification Algorithms**

| Method                                      | Macro-F1 |
|---------------------------------------------|----------|
| DistSuper + unigram                         | 61.74    |
| DistSuper + uni/bi/tri-gram                 | 63.84    |
| SVM + unigram                               | 74.50    |
| SVM + uni/bi/tri-gram                        | 75.06    |
| NBSVM                                        | 75.28    |
| RAE                                          | 75.12    |
| NRC (Top System in SemEval)                 | **84.73**|
| NRC - ngram                                  | 84.17    |
| SSWE$_u$                                      | **84.98**|
| SSWE$_u$+NRC                                 | **86.58**|
| SSWE$_u$+NRC-ngram                           | **86.48**|
SemEval 2014 Task 9 (b)

- **Coooollll**: A deep learning system for Twitter sentiment classification

![Diagram showing the process of training data, feature representation, learning algorithm, and sentiment classifier.](Image)
SemEval 2014 Task 9 (b)

• Results
  – Our system **Coooolll** is ranked 2\textsuperscript{nd} among 45 systems on Twitter2014 test set.
SemEval 2014 Task 9 (b)

- Results
  - Our system **Coooolll** is ranked 2\textsuperscript{nd} among 45 systems on Twitter2014 test set.

| Method | Positive/Negative/Neutral | | | | Positive/Negative | | | | | |
|--------|---------------------------|---|---|---|---|---|---|---|---|---|---|---|
| SSWE   | 70.49                     | 64.29 | 68.69 | 66.86 | 50.00 | 84.51 | 85.19 | 85.06 | 86.14 | 62.02 |
| Coooolll | 72.90                     | 67.68 | **70.40** | **70.14** | 46.66 | 86.46 | 85.32 | **86.01** | **87.61** | 56.55 |
| STATE  | 71.48                     | 65.43 | 66.18 | 67.07 | 44.89 | 83.96 | 82.82 | 84.39 | 86.16 | 58.27 |
| W2V    | 55.19                     | 52.98 | 52.33 | 50.58 | 49.63 | 68.87 | 71.89 | 74.50 | 71.52 | 61.60 |
| Top    | 74.84                     | 70.28 | 72.12 | **70.96** | 58.16 | - -   | - -   | - -   | - -   | - -   |
| Average| 63.52                     | 55.63 | 59.78 | 60.41 | 45.44 | - -   | - -   | - -   | - -   | - -   |
Roadmap

• Motivation
• The Proposed Method
• Experiments
  – Twitter Sentiment Classification
  – Word Similarity of Sentiment Lexicons
• Conclusion
Experiment Settings

• Setting
  – Evaluation metric
    \[
    \text{Accuracy} = \frac{\sum_{i=1}^{\#Lex} \sum_{j=1}^{N} \beta(w_i, c_{ij})}{\#Lex \times N}
    \]
  – Data

| Lexicon | Positive | Negative | Total |
|---------|----------|----------|-------|
| HL      | 1,331    | 2,647    | 3,978 |
| MPQA    | 1,932    | 2,817    | 4,749 |
| Joint   | 1,051    | 2,024    | 3,075 |

Accuracy = 8/10 = 80%
Results

- Evaluation
  - Word similarity of sentiment lexicons

| Embedding      | HL   | MPQA | Joint |
|----------------|------|------|-------|
| Random         | 50.00| 50.00| 50.00 |
| C&W            | 63.10| 58.13| 62.58 |
| Word2vec       | 66.22| 60.72| 65.59 |
| ReEmb(C&W)     | 64.81| 59.76| 64.09 |
| ReEmb(w2v)     | 67.16| 61.81| 66.39 |
| WVSA           | 68.14| 64.07| 67.12 |
| SSWE\(h\)      | 74.17| 68.36| 74.03 |
| SSWE\(r\)      | 73.65| 68.02| 73.14 |
| SSWE\(u\)      | 77.30| 71.74| 77.33 |
Conclusion

• Learn continuous representation of words for Twitter sentiment classification.

• Develop three neural networks for learning sentiment-specific word embedding (SSWE) from massive tweets without manual annotation.

• The effectiveness of SSWE has been verified in Twitter sentiment classification and word similarity judgement.
Thanks