Syntax-Aware Aspect Level Sentiment Classification with Graph Attention Networks

Binxuan Huang  
binxuanh@cs.cmu.edu

Kathleen M. Carley  
kathleen.carley@cs.cmu.edu
Outlines

- Introduction
- Previous work
- Method
- Experiments
- Conclusion
Introduction

- Sentence-level sentiment classification
  - Identify the overall sentiment polarity of a sentence as positive, negative, or neutral.
  - Many times a document is mixed with different aspects and opinions.
    - “The food in this restaurant is excellent, but the service is not good.”
  - Jiang et al. examined that 40% of sentiment classification errors come from not considering aspects[1].
Introduction

- Aspect level sentiment classification
  - Identify the sentiment polarity for each aspect in one document.

  - “The food in this restaurant is excellent, but the service is not good.”
Introduction

- Formal problem definition of Aspect level sentiment classification
  - Given a sentence \( s = [w_1, w_2, ..., w_i, ..., w_j, ..., w_n] \) and an aspect target \( t = [w_i, ..., w_j] \), the goal is to classify the sentiment as positive, negative, or neutral.

<table>
<thead>
<tr>
<th>Sentences</th>
<th>Aspects</th>
<th>Sentiment</th>
</tr>
</thead>
<tbody>
<tr>
<td>The food in this restaurant is excellent, but the service is not good.</td>
<td>food</td>
<td>+1</td>
</tr>
<tr>
<td>The food in this restaurant is excellent, but the service is not good.</td>
<td>service</td>
<td>-1</td>
</tr>
<tr>
<td>Boot time is super fast, around anywhere from 35 seconds to 1 minutes.</td>
<td>Boot</td>
<td>+1</td>
</tr>
</tbody>
</table>
• Motivation
  • Using syntactic structure of the sentence is helpful for identifying sentiment features directly related to the aspect target.
    – eg. “The food, though served with bad service, is actually great”
  • It is also helpful to resolve potential ambiguity in a word sequence.
    – eg. “Good food bad service”, “a bad sushi lover”

Most previous work treat a sentence as a word sequence and use LSTM or CNN to extract aspect related features.

- **Feature-based SVM** utilizes n-gram features, parse features and lexicon features for aspect-level sentiment classification. [1]

- **TD-LSTM** uses two LSTM networks to model the preceding and following contexts surrounding the aspect term. The last hidden states of these two LSTM networks are concatenated for predicting the sentiment polarity. [2]

- **AT-LSTM** first models the sentence via a LSTM model. Then it combines the hidden states from the LSTM with the aspect term embedding to generate the attention vector. [3]

- **AOA-LSTM** introduces an attention-over-attention (AOA) based network to model aspects and sentences in a joint way and explicitly capture the interaction between aspects and context sentences. [4]

- **PG-CNN** is a CNN based model where aspect features are used as gates to control the feature extraction on sentences. [5]
Method

- Model components
  - Text representation
  - Graph attention network
  - Target-dependent graph attention network
  - Classification
Method

- Text representation

Sentence:
  delivery was early too

Aspect:
  delivery

Each node $i$ in the dependency graph $A$ is associated with a GloVe word embedding vector or a contextual BERT representation, denoted as $x_i$
Method

• Graph attention network (GAT)
  - GAT is a variant of graph neural networks, which propagate features and learn node representations on a graph.
  - At each layer, GAT updates one node’s representation by aggregating its neighborhood’s representations using multihead attention.

\[
H_{l+1} = GAT(H_l, A)
\]

\[
h_{l+1}^i = \text{multihead}\left(\left\{h_j^i; j \in \text{nei}[i]\right\}\right)
\]

\[
= \left\|_{k=1}^K \sigma \left( \sum_{j \in \text{nei}[i]} \alpha_{lk}^i W_{lk}^i h_l^i \right) \right\|
\]

\[
\alpha_{lk}^i = \frac{\exp\left(\text{LeakyReLU}\left(a_{lk}^T W_{lk}^i h_l^i \| W_{lk}^u h_l^u\right)\right)}{\sum_{u \in \text{nei}[i]} \exp\left(\text{LeakyReLU}\left(a_{lk}^T W_{lk}^i h_l^i \| W_{lk}^u h_l^u\right)\right)}
\]
Method

- **Target-dependent graph attention network (TD-GAT)**
  - In a vanilla GAT network, the aspect information is not explicitly modeled.
  - We use an LSTM network to model the dependency for the aspect terms across layers.

\[
H_{l+1} = GAT(H_l, A)
\]

\[
H_{l+1}, C_{l+1} = LSTM(\overline{H_{l+1}}, (H_l, C_l))
\]

\[
H_0, C_0 = LSTM(XW + b, (0,0))
\]
Method

• Classification
  – The probability for each sentiment class is computed by a softmax function after a linear projection layer.
  \[
P(y = c) = \frac{\exp(W h_L^t + b)_c}{\sum_{i \in C} \exp(W h_L^t + b)_i}
\]
  – We minimize the cross entropy loss with L_2 regularization to train our model
  \[
  loss = - \sum_{c \in C} I(y = c) \cdot \log(P(y = c)) + \lambda ||\theta||^2
  \]
Experiments

- Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Positive</th>
<th>Neutral</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop-Train</td>
<td>767</td>
<td>373</td>
<td>673</td>
</tr>
<tr>
<td>Laptop-Dev</td>
<td>220</td>
<td>87</td>
<td>193</td>
</tr>
<tr>
<td>Laptop-Test</td>
<td>341</td>
<td>169</td>
<td>128</td>
</tr>
<tr>
<td>Restaurant-Train</td>
<td>1886</td>
<td>531</td>
<td>685</td>
</tr>
<tr>
<td>Restaurant-Dev</td>
<td>278</td>
<td>102</td>
<td>120</td>
</tr>
<tr>
<td>Restaurant-Test</td>
<td>728</td>
<td>196</td>
<td>196</td>
</tr>
</tbody>
</table>

Table 1: Statistics of the datasets.
Experiments

- Using BERT representation

Sentence: workload is heavy

Aspect: workload

BERT tokenizer

'work', '##load', 'is', 'heavy'

'Sentence: workload is heavy'

'work', '##load'

Aspect: workload
Experiments

- Baseline comparisons
  - We compare our method with various baselines.
  - We report the performance of our model with different number of layers.

<table>
<thead>
<tr>
<th></th>
<th>Laptop</th>
<th>Restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature+SVM</td>
<td>70.5</td>
<td>80.2</td>
</tr>
<tr>
<td>TD-LSTM</td>
<td>68.1</td>
<td>75.6</td>
</tr>
<tr>
<td>AT-LSTM</td>
<td>68.9</td>
<td>76.2</td>
</tr>
<tr>
<td>MemNet</td>
<td>72.4</td>
<td>80.3</td>
</tr>
<tr>
<td>IAN</td>
<td>72.1</td>
<td>78.6</td>
</tr>
<tr>
<td>PG-CNN</td>
<td>69.1</td>
<td>78.9</td>
</tr>
<tr>
<td>AOA-LSTM</td>
<td>72.6</td>
<td>79.7</td>
</tr>
<tr>
<td>TD-GAT-GloVe (3)</td>
<td>73.7</td>
<td>81.1</td>
</tr>
<tr>
<td>TD-GAT-GloVe (4)</td>
<td><strong>74.0</strong></td>
<td>80.6</td>
</tr>
<tr>
<td>TD-GAT-GloVe (5)</td>
<td>73.4</td>
<td><strong>81.2</strong></td>
</tr>
<tr>
<td>BERT-AVG</td>
<td>76.5</td>
<td>78.7</td>
</tr>
<tr>
<td>BERT-CLS</td>
<td>77.1</td>
<td>81.2</td>
</tr>
<tr>
<td>TD-GAT-BERT (3)</td>
<td>79.3</td>
<td>82.9</td>
</tr>
<tr>
<td>TD-GAT-BERT (4)</td>
<td>79.8</td>
<td><strong>83.0</strong></td>
</tr>
<tr>
<td>TD-GAT-BERT (5)</td>
<td><strong>80.1</strong></td>
<td>82.8</td>
</tr>
</tbody>
</table>

Table 2: Comparison results of different methods on laptop and restaurant datasets. Numbers in parentheses indicate number of layers in our model.
Experiments

- Effects of target information
  - To examine the effects of explicitly modeling target information, we remove the LSTM unit in our model and compare it with TD-GAT.
  - As shown in the table, explicitly capturing aspect target information consistently improves the performance of the TD-GAT-GloVe over the GAT-GloVe model.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Laptop</th>
<th>Restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>GAT-GloVe</td>
<td>73.0</td>
<td>72.1</td>
</tr>
<tr>
<td>TD-GAT-GloVe</td>
<td>73.7</td>
<td>74.0</td>
</tr>
<tr>
<td>GAT-BERT</td>
<td>78.1</td>
<td>78.5</td>
</tr>
<tr>
<td>TG-GAT-BERT</td>
<td>79.3</td>
<td>79.8</td>
</tr>
</tbody>
</table>

Table 3: An ablation study shows the effect of explicit target information.
Experiments

- Effects of model depth

![Graphs showing the effects of model depth for Laptop and Restaurant scenarios. The graphs compare TD-GAT-GloVe and TD-GAT-BERT models.](image-url)
Experiments

- **Model size**
  - Using the same dimension of hidden states, our TD-GAT-GloVe has a lower model size compared to these LSTM-based methods.
  - When we switch from GloVe embeddings to BERT representations, the training time for a three-layer TD-GAT model on the restaurant dataset only increases from 1.12 seconds/epoch to 1.15 seconds/epoch.

<table>
<thead>
<tr>
<th>Models</th>
<th>Model size $(\times 10^6)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD-LSTM</td>
<td>1.45</td>
</tr>
<tr>
<td>MemNet (3)</td>
<td>0.36</td>
</tr>
<tr>
<td>IAN</td>
<td>2.17</td>
</tr>
<tr>
<td>AOA-LSTM</td>
<td>2.89</td>
</tr>
<tr>
<td>TD-GAT-GloVe (3)</td>
<td>1.00</td>
</tr>
<tr>
<td>TD-GAT-GloVe (4)</td>
<td>1.09</td>
</tr>
<tr>
<td>TD-GAT-GloVe (5)</td>
<td>1.18</td>
</tr>
<tr>
<td>BERT-CLS</td>
<td>335.14</td>
</tr>
<tr>
<td>TD-GAT-BERT (3)</td>
<td>1.30</td>
</tr>
<tr>
<td>TD-GAT-BERT (4)</td>
<td>1.39</td>
</tr>
<tr>
<td>TD-GAT-BERT (5)</td>
<td>1.49</td>
</tr>
</tbody>
</table>

Table 4: The model size (number of parameters) of our model as well as baselines.
Conclusion

- In this paper, we propose a novel target-dependent graph attention neural network for aspect level sentiment classification.

- Using GloVe embeddings, our approach TD-GAT-GloVe outperforms various baseline models.

- After switching to BERT representations, we show that TD-GAT-BERT achieves much better performance.

- It is lightweight and requires fewer computational resources and less training time than fine-tuning the original BERT model.
Future Direction

- Future work could consider using an attention mechanism to focus on important words in the aspect.

- Since this work only uses the dependency graph and ignores various types of relations in the graph, we plan to incorporate dependency relation types into our model and take part-of-speech tagging into consideration as well in the future.

- We would also like to combine such a graph-based model with a sequence-based model to avoid potential noise from dependency parsing errors.


