Aspect Based Sentiment Analysis

...

Priyansh Agrawal, Yash Goyal

International Institute of Information Technology, Hyderabad
Team - Linguasouls

Reference Paper

- Attention-based LSTM for Aspect-level Sentiment Classification.

  → Yequan Wang, Minlie Huang, Li Zhao, Xiaoyan Zhu

Motivation

- Aspect-level sentiment analysis has received much attention these years. It is a fine-grained task of sentiment analysis since it provides more complete and in-depth results.
- In our work, we reveal that the sentiment polarity of a sentence is not only determined by the content but is also highly related to the concerned aspect and can improvise with the attention mechanism.
- It is worthwhile to explore the connection between an aspect and the content of a sentence.

Introduction

- Deal with aspect level sentiment classification and find that the sentiment polarity of a sentence is highly dependent on both content and aspect.
- For example, the sentiment polarity of "*Staffs are not that friendly, but the taste covers all*" will be positive if the aspect is *food* and negative when considering the aspect *service*.
- Propose an attention mechanism to attend to the important part of the sentence in response to a specific aspect.
- Design an aspect to sentence attention mechanism that can concentrate on the key part of the sentence given the aspect.
- Explore the potential correlation of aspect and sentiment polarity in aspect-level sentiment classification. In order to capture important information in response to a given aspect, we design an attention-based LSTM with some variants.

Dataset

We used the dataset of SemEval 2014 (Task 4). The dataset consists of customers reviews of restaurant domain. Each review contains a list of aspects and corresponding polarities. Our aim is to identify the aspect polarity of a sentence with the corresponding aspect. The statistics of the data is presented in Table 1.

Sentiment	Training Data (sent.)	Testing Data (sent.)
Positive	2164	728
Negative	898	210
Neutral	637	196

Baseline

- For each aspect term in a test sentence, check if the aspect term had been encountered in training data.
- If it had been encountered, then assign to the aspect term (in test sentence) the most frequent polarity (positive, negative or neutral) that the aspect term had in the k-most similar training sentences that contained the aspect term.
- If it had not been encountered in the training sentences, assign to the aspect term (in the test sentence) the most frequent polarity of all aspect terms in training data.
- Top k most similar training sentences for target sentence (test sentence) is retrieved using cardinal-based similarity methods:- “Jaccard similarity” and “Dice similarity”. Optimal value of k comes out to be 12.

Baseline (cont..)

- DSC (Dice similarity coefficient) is given by :-

$$\text{similarity} = \frac{2 * |X \cap Y|}{|X| + |Y|}$$

- JSC (Jaccard similarity coefficient) is given by:-

$$jac(x, y) = \frac{|x \cap y|}{|X| + |y| - |x \cap y|}$$

Baseline (cont.)

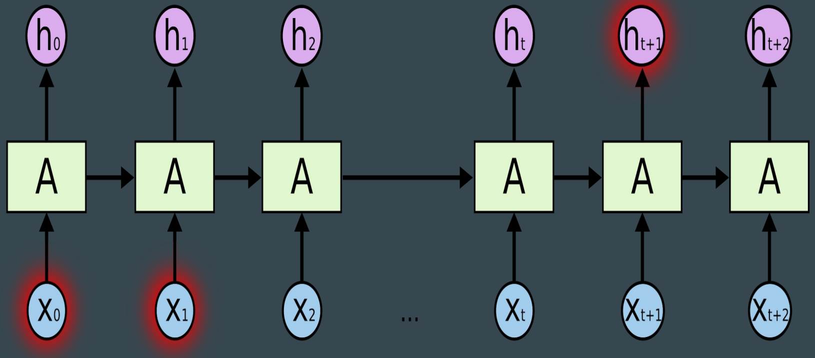
- We also experimented with similarity measure using word-embeddings and cosine-similarity in which we take the embeddings of the words in the sentence and compute their average weighted by the frequency of each word (also known as the term frequency - tf). After getting the sentence vectors we use the Cosine similarity.
- Cosine similarity between two embeddings is given by:-

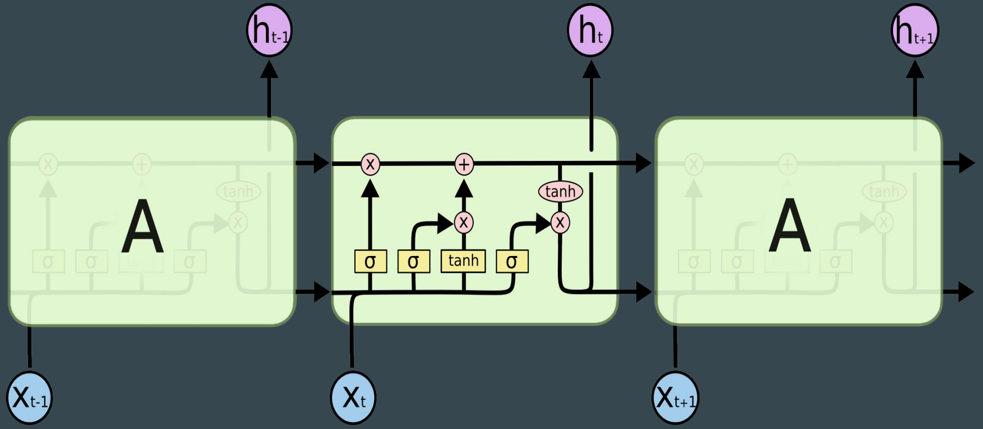
$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

LSTM

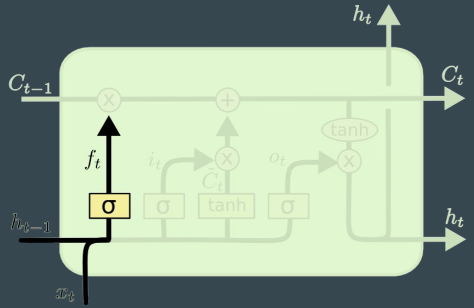
Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN(Recurrent Neural Networks), capable of learning long-term dependencies.

- Explicitly designed to avoid the long-term dependency problem.

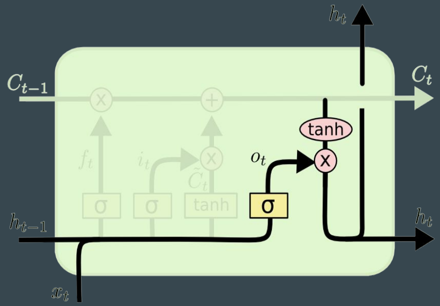




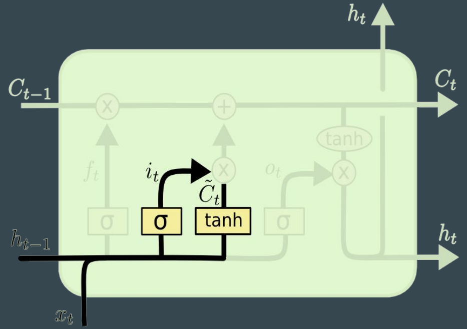
LSTM (cont..)



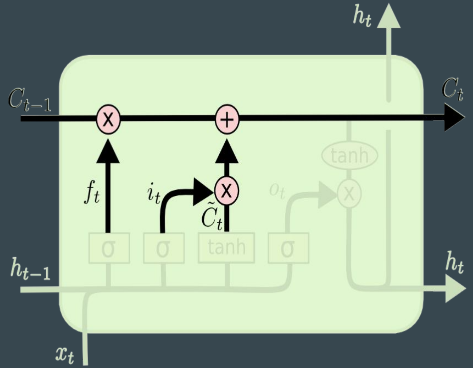
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$



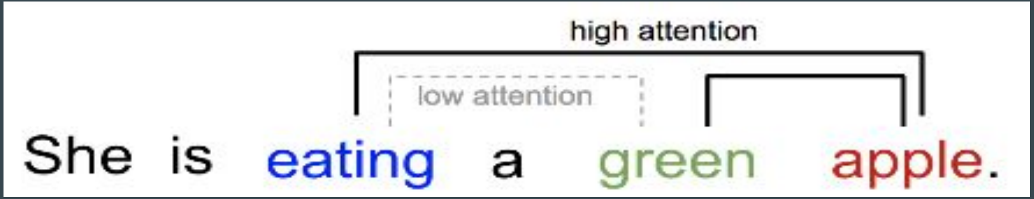
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Attention Mechanism

- Attention is, to some extent, motivated by how we pay visual attention to different regions of an image or correlate words in one sentence.
- Attention in deep learning can be broadly interpreted as a vector of importance weights: in order to predict or infer one element, such as a pixel in an image or a word in a sentence.
- We estimate using the attention vector how strongly it is correlated with (or "*attends to*") other elements and take the sum of their values weighted by the attention vector as the approximation of the target.

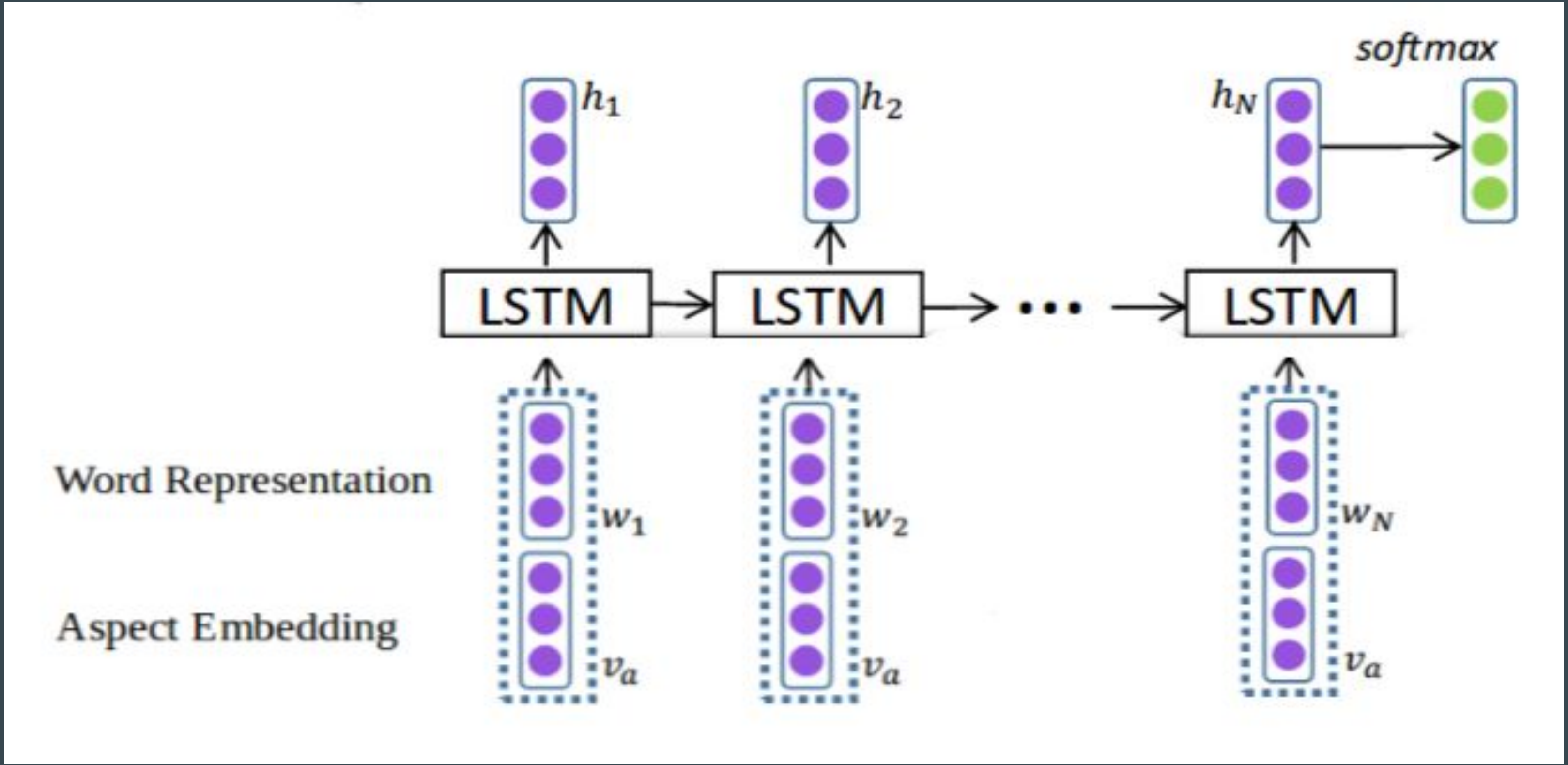


When we see "eating", we expect to encounter a food word very soon. The color term describes the food, but probably not so much with "eating" directly.

Approach

- AE-LSTM (LSTM with Aspect Embedding)
    - In this model, we have concatenated aspect term embedding of a sentence with word vectors and give this as an input to our LSTM.
    - An aspect term can have multiple words in it (example , red chilly soup , interior decoration). The embedding for such an aspect with more than 1 word is calculated by averaging its constituents word vectors..
    - The last hidden vector represents our sentence which is put into a softmax layer after linearizing it into a vector of length 3.
    - This three length vector is corresponds to our final output or label vector for labels :- positive, negative and neutral.

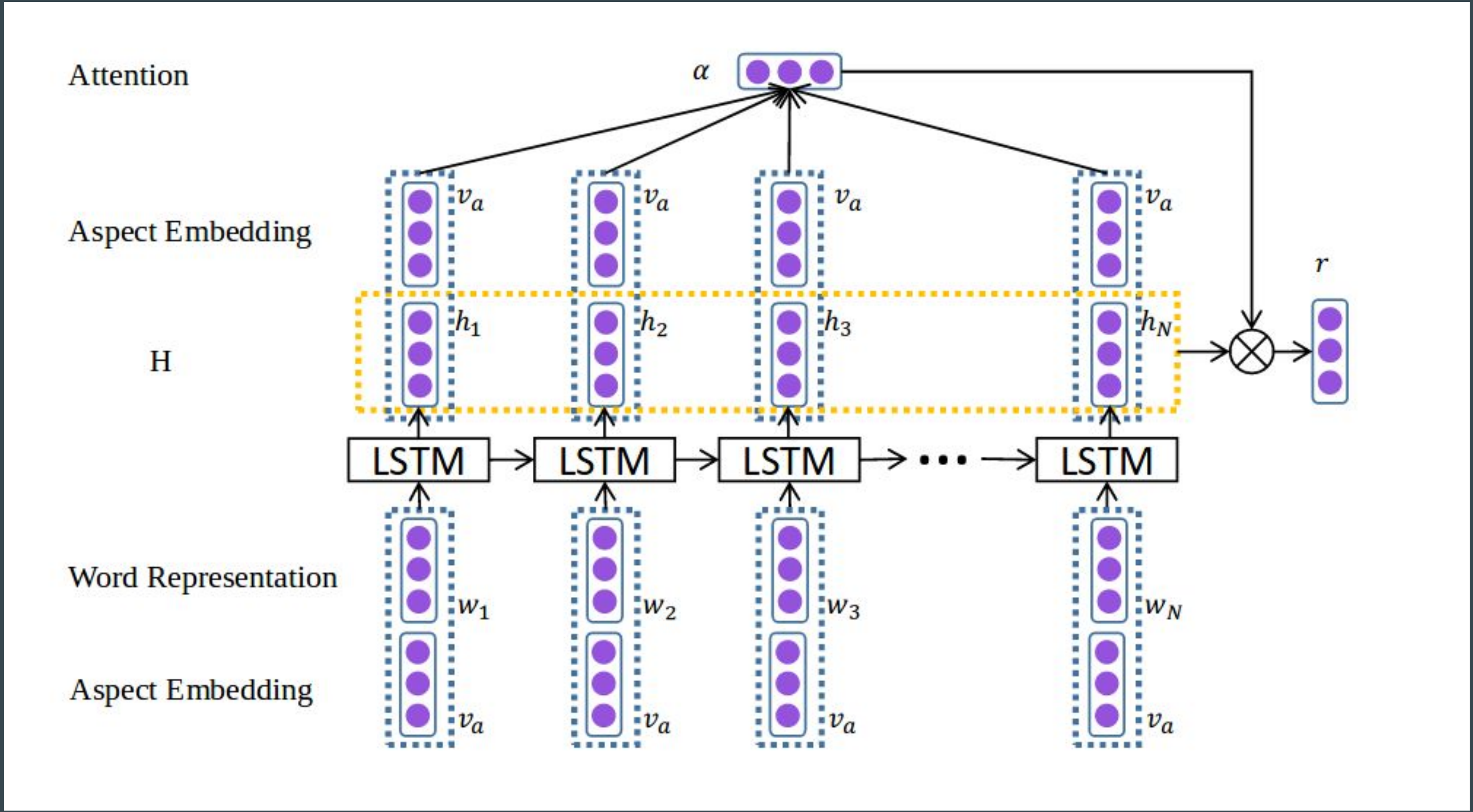
Approach (cont..)



Approach (cont..)

- ATAE-LSTM (Attention-based LSTM with Aspect Embedding)
  - The standard LSTM cannot detect which is the important part for aspect-level sentiment classification.
  - Incorporating attention mechanism can capture the key part of sentence in response to a given aspect.
  - In this model in order to take better advantage of aspect information we append the aspect embedding into each hidden vector which plays a key role in computing the attention weights.
  - The following picture shows our ATAE-model.

Approach (cont..)



Approach (cont.)

- $H \in \mathbb{R}^{d \times N}$ is a matrix consisting of hidden vectors $[h_1, h_2, \ldots h_N]$
- $v_a \in \mathbb{R}^{da}$ is the aspect embedding.
- $e_N \in \mathbb{R}^{N}$ is a vector of 1's.
- α is the attention weight vector and r is the weighted representation of sentence given aspect.
- $M = \tanh(W[H; v_a \otimes e_N])$ where $M \in \mathbb{R}^{(d+da) \times N}$, $W \in \mathbb{R}^{(d+da)x(d+da)}$.
  Here $v_a \otimes e_N = [v_a; v_a; \ldots; v_a] \in \mathbb{R}^{da \times N}$
- $\alpha = \mathrm{softmax}(w^T M)$, $r = H\alpha^T$ where $\alpha \in \mathbb{R}^N$ and $r \in \mathbb{R}^d$.
- The final sentence representation is given by:
  $h^* = \tanh(W_p r + W_x h_N)$ where $h^* \in \mathbb{R}^d$, W_p and W_x are the projection matrix.

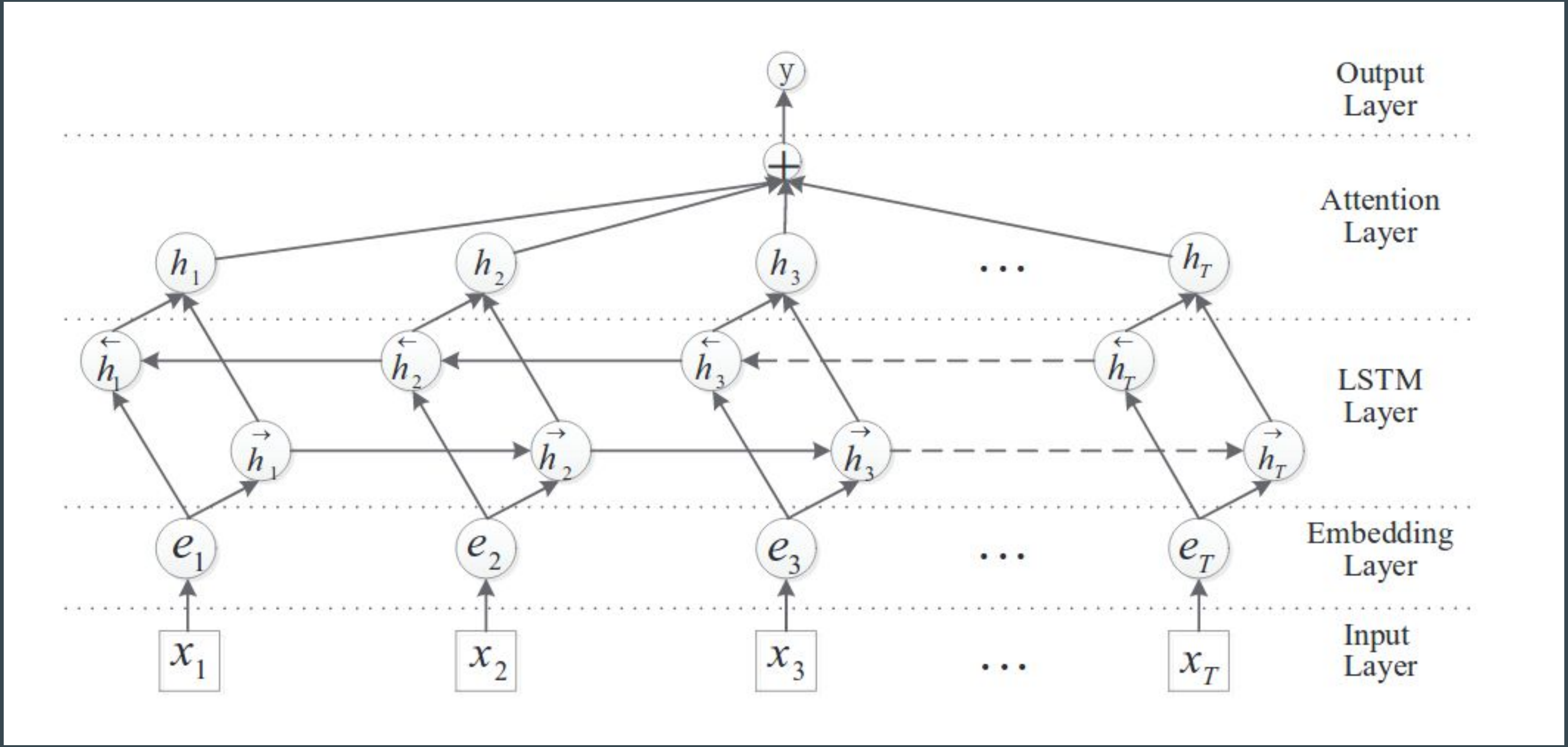
Approach (cont..)

- $y = \text{softmax}(W_s h^* + b_s)$ is the final label vector, $y \in \mathbb{R}^{3x1}$.
- Here, $W_s \in \mathbb{R}^{3xd}$ and $b_s \in \mathbb{R}^{3x1}$ are the parameters for softmax layer which are learned during training just like W_p and W_x .

Approach (cont..)

- Experiment using Bi-LSTM
    - LSTMs are limited in the sense that at any particular node, it can have access to only the past information and hence the output can only be generated based on what the network has seen (learned information).
    - Bi-LSTMs can overcome this. These are preferred in the applications where getting the past and future information can improve the performance.
    - It has two networks, one access information in forward direction and another access in the reverse direction.
    - We used it in our case to see if they can make use of past and future information given a word sequence and its corresponding aspect.

Approach (cont..)


y
Output Layer
Attention Layer
h_1 h_2 h_3 … h_T
LSTM Layer
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ $\overleftarrow{h}_3$ $\overleftarrow{h}_T$
$\overrightarrow{h}_1$ $\overrightarrow{h}_2$ $\overrightarrow{h}_3$ $\overrightarrow{h}_T$
Embedding Layer
e_1 e_2 e_3 … e_T
Input Layer
x_1 x_2 x_3 … x_T

Experiment

- As a preprocessing, we restructured our dataset as follows :
  - Removed unwanted content like <xml></xml> tags and some unwanted punctuation marks along with the sentences which are not in correct encoding which gives encoding error while implementation.
  - Then we structured the data in format shown below.
    - <sentence> -- sentence
    - <aspect-term> -- aspect term of above sentence
    - <polarity> -- 1 or -1 or 0 each corresponds to sentiment (positive, negative , neutral)

- We have used the pretrained glove vectors for word-embeddings.

Experiment (cont..)

- The weight matrices and the OOV word vectors are initialised with normal distribution U(-0.01,0.01) as suggested in the paper.
- The dimension of word vectors, aspect embeddings and the size of hidden layer are 300.
- The length of attention weights is the same as the length of sentence.
- We trained all models with a batch size of 25 examples, L2-regularization weight of 0.001 and initial learning rate of 0.05 for AdaGrad (adaptive gradient descent algorithm).
- The number of epochs or training iteration is set to 25.
- All the above hyperparameters are not exhaustively searched but these were the optimal for us.

Results

Models	Three-way Accuracy
BaseLine	65.52
AE-LSTM	64.19
ATAE-LSTM	75.04
AE-Bi-LSTM	74.86
ATAE-Bi-LSTM	75.48

Results (cont..)

Baseline	F1-score	Precision	Recall
Positive	78.39	69.61	89.69
Negative	33.12	48.18	25.23
Neutral	26.24	43.02	18.87

ATAE-Bi-LSTM	F1-score	Precision	Recall
Positive	85.01	79.92	90.79
Negative	61.46	61.03	61.90
Neutral	44.82	69.14	33.16

Conclusion

- Implemented attention-based LSTMs for aspect-level sentiment classification.
- Proposed models can concentrate on different parts of sentence when different aspects are given which is done via attention mechanism.
- Experiments show that ATAE-LSTM and Bi-LSTM obtain nearly same but superior performance over the baseline and other models.

THANK YOU