Carnegie Mellon

isr institute for SOFTWARE RESEARCH

Aspect Level Sentiment Classification with Attention-over-Attention Neural Networks

Binxuan Huang

Yanglan Ou

Kathleen M. Carley

Introduction

- Document-level sentiment classification
  - Identify the overall sentiment polarity of a document 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

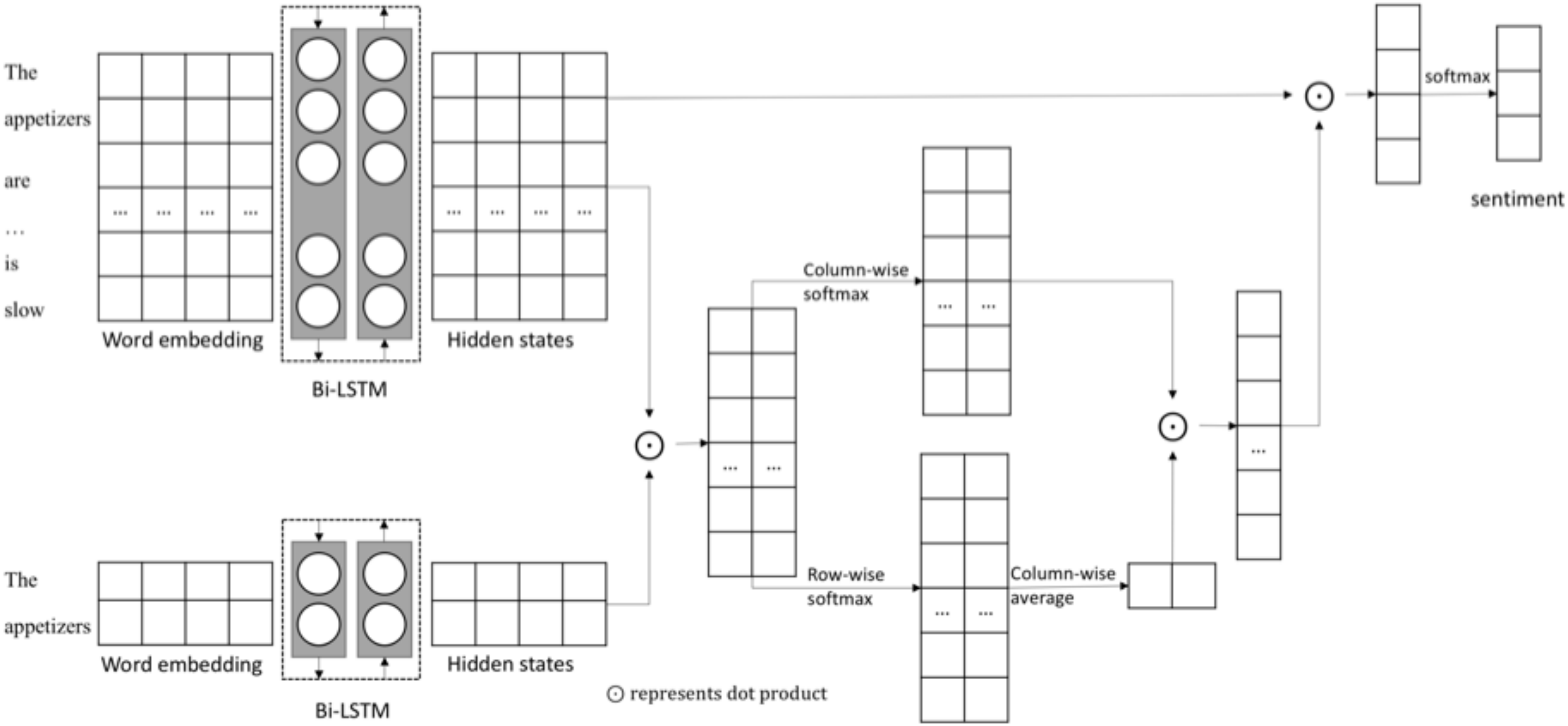
- Formal problem definition of Aspect level sentiment classification
  - Given a sentence $s = [w_1, w_2, \ldots, w_i \ldots, w_j, \ldots w_n]$ and an aspect target $t = [w_i, \ldots, w_j]$, the goal is to classify the sentiment as positive, negative, or neutral.

Sentences	Aspects	Sentiment
The food in this restaurant is excellent, but the service is not good.	food	+1
The food in this restaurant is excellent, but the service is not good.	service	-1
Boot time is super fast, around anywhere from 35 seconds to 1 minutes.	Boot time	+1

Method

- Components in the model
  - Word embedding layer
  - LSTM
  - Attention over Attention(AOA)
  - Final classification layer

Method

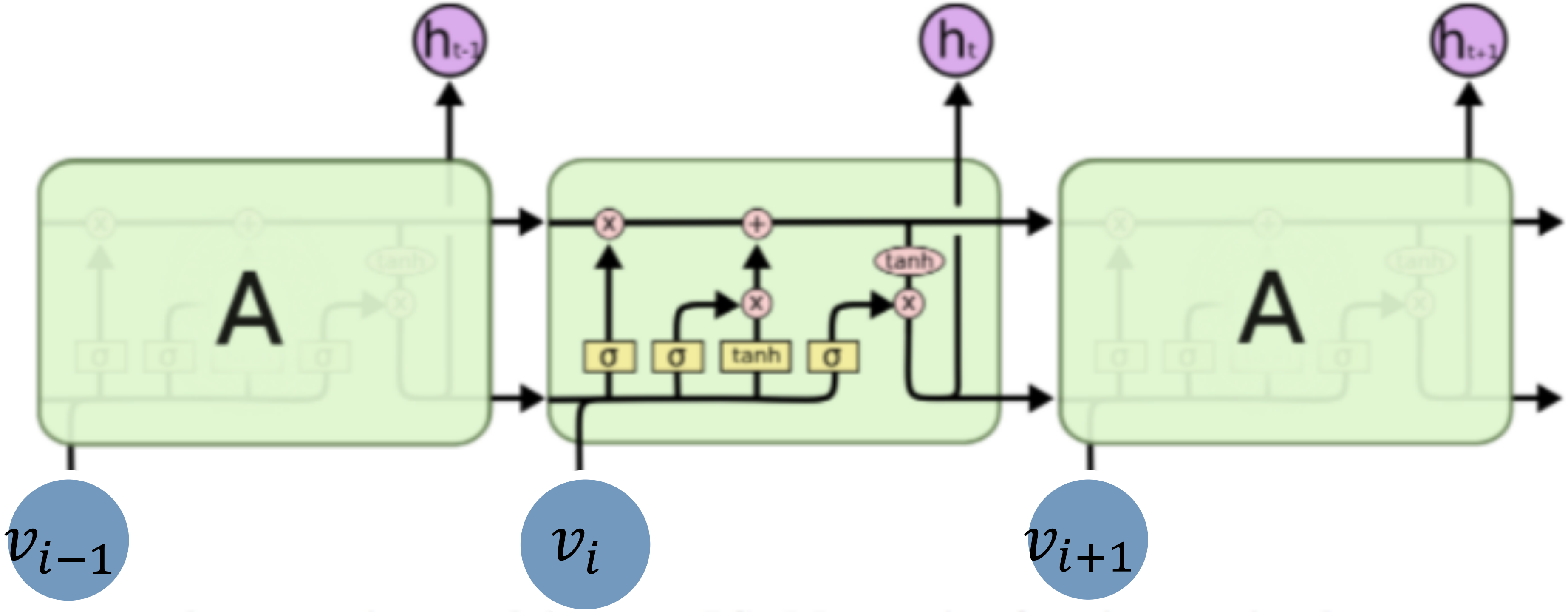

The
appetizers
are
...
is
slow
Word embedding
Bi-LSTM
Hidden states
The
appetizers
Word embedding
Bi-LSTM
Hidden states
Column-wise
softmax
Row-wise
softmax
Column-wise
average
softmax
sentiment
⊙ represents dot product

Method

- Components in the model
  - Word embedding layer[2]
    - Treat each word as a low-dimensional real-value vector R^d.
    - Similar words have similar vectors.
    - A sentence of length n can be represented as a sequence of vectors $[v_1, \dots, v_n] \in R^{n\times d}$
    - Similarly, an aspect of length m can be represented as $[v_i, \dots, v_j] \in R^{m\times d}$

Method

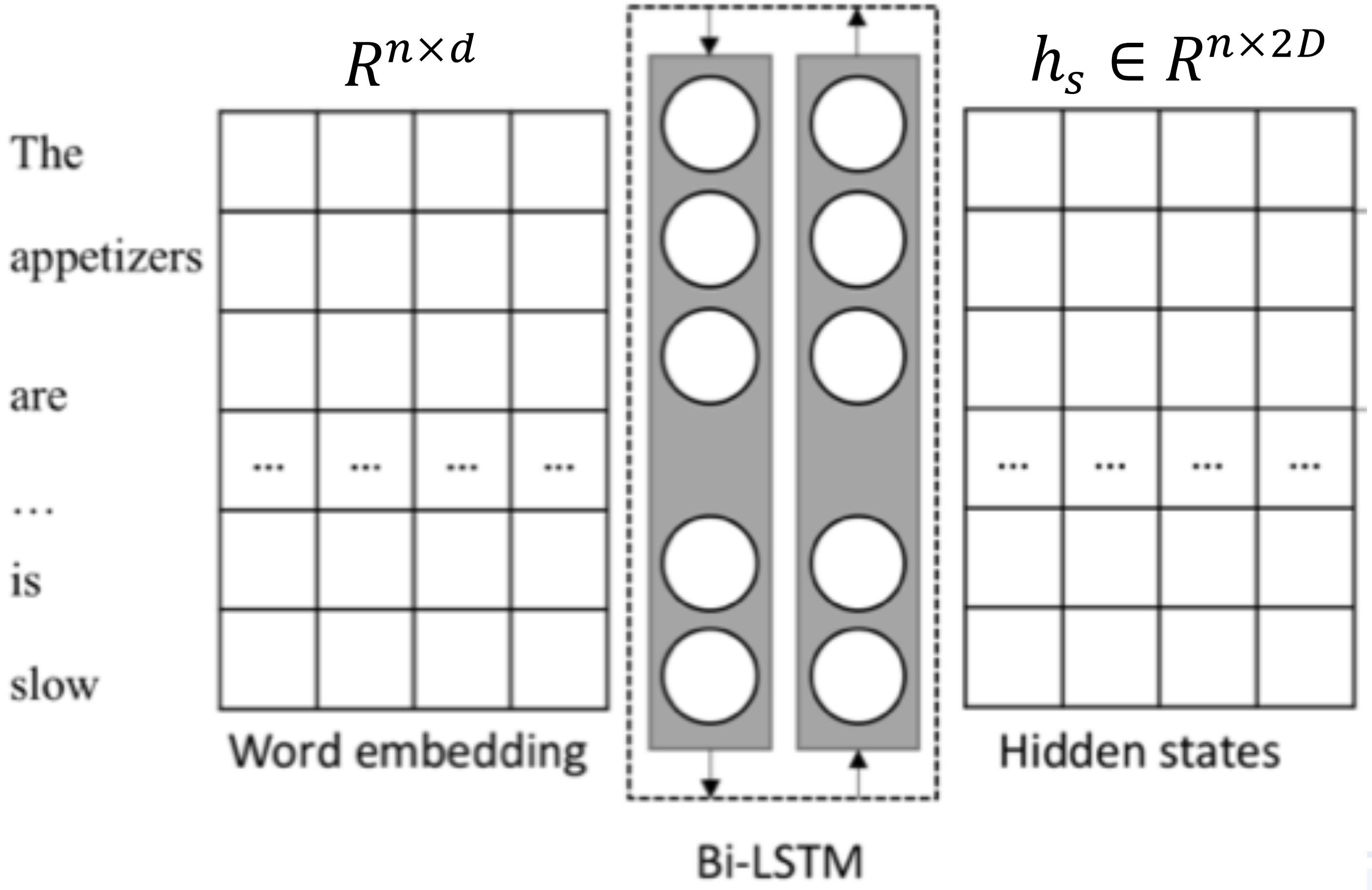
- Components in the model
  - LSTM layer[3]: we use LSTM to get the semantic meaning of texts.



The repeating module in an LSTM contains four interacting layers.

Method

- Components in the model
  - LSTM layer: we use LSTM to get the semantic meaning of texts.

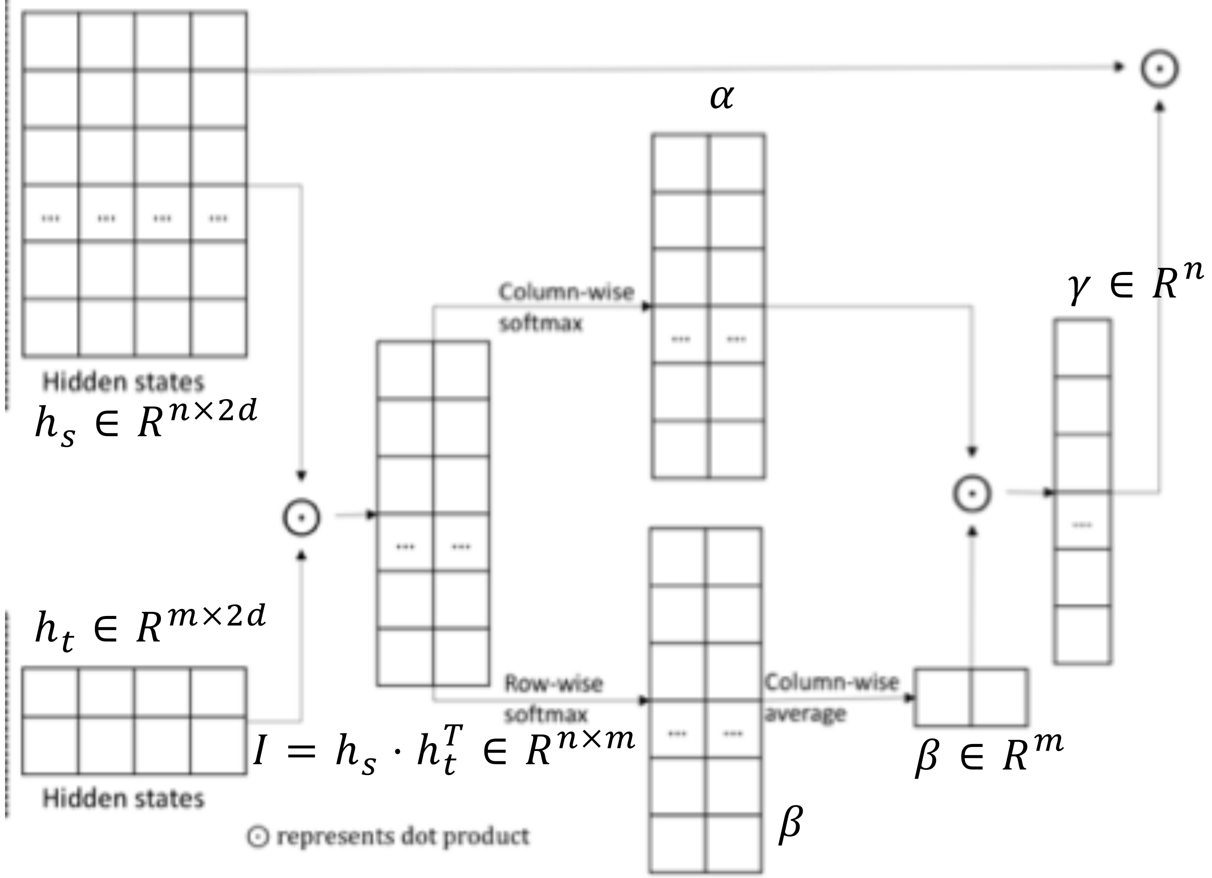


Method

- AOA layer[4]: generate an attention vector from aspect hidden states to indicate important words in the sentence.
  - From previous LSTM layer, we get two matrices $h_s \in R^{n\times 2d}$ and $h_t \in R^{m\times 2d}$. Each row in these matrices represent the semantic meaning of one word.
  - $I = h_s \cdot h_t^T \in R^{n\times m}$, I_{ij} is the interaction between $word_i$ in sentence and $word_j$ in aspect.
  - We first row-wise normalize I with softmax operation, then do a column-wise average to get $\beta \in R^m$. β indicates important parts in the aspect term.
  - We apply column-wise normailization on I and get α
  - Final sentence attention is $\gamma = \alpha \cdot \beta^T \in R^n$

Method

- AOA layer: generate an attention vector from aspect hidden states to indicate important words in the sentence.



Method

- Final classification
  - The classification feature is $\mathrm{r} = \mathrm{h}_s^{\mathrm{T}} \cdot \gamma \in R^{2D}$
  - The probability of sentiment label c is:
    - $P(y = c) = \frac{\exp(W \cdot r + b)_c}{\sum_i \exp(W \cdot r + b)_i}$
    - Cross-entropy loss $= -\sum_{s,a} \sum_c I(y = c) log P(y = c) + \lambda ||\theta||^2$
  - Minimize the loss function with regard to all the parameters.

Experiments

- Two domain-specific datasets

Dataset	Positive	Neutral	Negative
Laptop-Train	994	464	870
Laptop-Test	341	169	128
Restaurant-Train	2164	637	807
Restaurant-Test	728	196	196

Table 1: Statistics of the datasets from SemEval 2014 Task 4.

Experiments

- Comparison results

Methods	Restaurant	Laptop
TD-LSTM [24]	0.756	0.681
AT-LSTM [29]	0.762	0.689
ATAE-LSTM [29]	0.772	0.687
IAN [12]	0.786	0.721
AOA-LSTM	**0.812** (0.797$\pm$0.008)	**0.745** (0.726$\pm$0.008)

Table 2. Comparison results. For our method, we run it 10 times and show ”best (mean$\pm$std)”. Performance of these baselines are cited from their original papers.

Experiments

- Case study

Aspect	Sentence	Ans./Pred.
appetizers	the appetizers are ok , but the service is slow .	0/0
service	the appetizers are ok , but the service is slow .	-1/-1
food	great food but the service was dreadful !	+1/+1
service	great food but the service was dreadful !	-1/-1
boot time	boot time is super fast , around anywhere from 35 seconds to 1 minute .	+1/+1

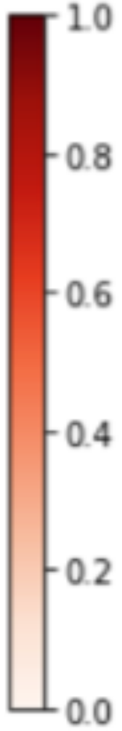


Table 3: Examples of final attention weights for sentences. The color depth denotes the importance degree of the weight in attention vector γ.

- [1] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, pages 151–160.
- [2] Yoshua Bengio, R´ejean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. Journal of machine learning research 3(Feb):1137–1155.
- [3] Sepp Hochreiter and J¨urgen Schmidhuber. 1997.Long short-term memory. Neural computation 9(8):1735–1780.
- [4] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehen-sion. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. pages 593–602.