Recurrent Entity Networks with Delayed Memory Update for Targeted Aspect-based Sentiment Analysis

Fei Liu **Trevor Cohn** **Timothy Baldwin**
School of Computing and Information Systems
The University of Melbourne
Victoria, Australia
fliu3@student.unimelb.edu.au
t.cohn@unimelb.edu.au tb@ldwin.net

Abstract

While neural networks have been shown to achieve impressive results for sentence-level sentiment analysis, targeted aspect-based sentiment analysis (TABSA) — extraction of fine-grained opinion polarity w.r.t. a pre-defined set of aspects — remains a difficult task. Motivated by recent advances in memory-augmented models for machine reading, we propose a novel architecture, utilising external "memory chains" with a delayed memory update mechanism to track entities. On a TABSA task, the proposed model demonstrates substantial improvements over state-of-the-art approaches, including those using external knowledge bases.[1]

1 Introduction

Targeted aspect-based sentiment analysis (TABSA) is the task of identifying fine-grained opinion polarity towards a specific aspect associated with a given target. The task requires classification of opinions on different entities across a range of different attributes, with the expectation that there will be no overt opinion expressed on a given entity for many attributes. This can be seen in Example (1), e.g., where opinions on the aspects SAFETY and PRICE are expressed for entity *LOC1* but not entity *LOC2*:[2]

(1) LOC1 is your best bet for secure although expensive and LOC2 is too far.

Target	Aspect	Sentiment
LOC1	SAFETY	positive
LOC1	PRICE	negative
LOC2	TRANSIT-LOCATION	negative

The earliest work on (T)ABSA relied heavily on feature engineering (Wagner et al., 2014; Kiritchenko et al., 2014), but more recent work based on deep learning has used models such as LSTMs to automatically learn aspect-specific word and sentence representations (Tang et al., 2016a).

Despite these successes, keeping track of multiple entity–aspect pairs remains a difficult task, even for an LSTM. As reported in Saeidi et al. (2016), a target-dependent biLSTM is ineffective, both in terms of aspect detection and sentiment classification, compared to a simple logistic regression model with n-gram features. Intuitively, we would expect that a model which better captures linguistic structure via the original word sequencing should perform better, which provides the motivation for this research.

More recently, successful works in (T)ABSA have explored the idea of leveraging external memory (Tang et al., 2016b; Chen et al., 2017). Their models are largely based on memory networks (Weston et al., 2015), originally developed for reasoning-focused machine reading comprehension tasks. In contrast to memory networks, where each input sentence/word occupies a memory slot and is then accessed via attention independently, recent advances in machine reading suggest that processing inputs sequentially is beneficial to overall performance (Seo et al., 2017; Henaff et al., 2017).

However, successful machine reading models may not be directly applicable to TABSA due to the key difference in the granularity of inputs between the two tasks: on the Children's Book Test corpus (CBT), for example, competitive models take as input a window of text, centred around candidate entities, with crucial information contained within that window (Hill et al., 2015; Henaff et al., 2017). In TABSA, given the fine-grained nature of the task, it is common practice for models to

[1] Code available at https://github.com/liufly/delayed-memory-update-entnet.

[2] Note that in our dataset, all entity mentions have been pre-nomalised to *LOCn*, where n is an index.

Proceedings of NAACL-HLT 2018, pages 278–283
New Orleans, Louisiana, June 1 - 6, 2018. ©2018 Association for Computational Linguistics

operate at the word- rather than chunk/sentence-level. It is not uncommon to see examples like Example (1), where the sentence starts with *LOC1*, but the negative PRICE sentiment towards the entity is not expressed until much later. Moreover, phrases such as *best bet* and *although* play the role of triggers, indicating that succeeding tokens bear aspect/sentiment signal. This key difference necessitates the ability to model the delayed activation of memory updates.

In this work, we propose a novel model architecture for TABSA, augmented with multiple "memory chains", and equipped with a delayed memory update mechanism, to keep track of numerous entities independently. We evaluate the effectiveness of the proposed model over the task of TABSA, and achieve substantial improvements over a number of baselines, including one incorporating external knowledge bases, setting a new state of the art in both sentiment classification and aspect detection.

2 Methodology

Task description. In TABSA, a sentence s typically consists of a sequence of words: $\{w_1, \ldots, w_i, \ldots, w_m\}$ where w_i denotes words interleaved with one or more targets (t), which we assume to be pre-identified as with *LOC1* and *LOC2* in Example (1). Following Saeidi et al. (2016), we frame the task as a 3-class classification problem: given a sentence s, a pre-identified set of target entities T and fixed set of aspects A, predict the sentiment polarity $y \in \{\textit{positive}, \textit{negative}, \textit{none}\}$ over the full set of target–aspect pairs $\{(t, a) : t \in T, a \in A\}$. For example, (*LOC1*,SAFETY) has gold-standard polarity *positive*, while (*LOC1*,TRANSIT-LOCATION) has polarity *none*.

Proposed model. To this end, we design a neural network architecture, capable of tracking and updating the states of entities at the right time with external memory, making it a natural fit for the task. Specifically, our model maintains a number of "memory chains" $\boldsymbol{h}^j$, one for each entity with the key $\boldsymbol{k}^j$ and dynamically updates the states ($\boldsymbol{h}^j$) of them as it progresses through the sentence with the help of the delay recurrence $\boldsymbol{d}^j$, taking previous activations into account. An illustration of our model is provided in Figure 1.

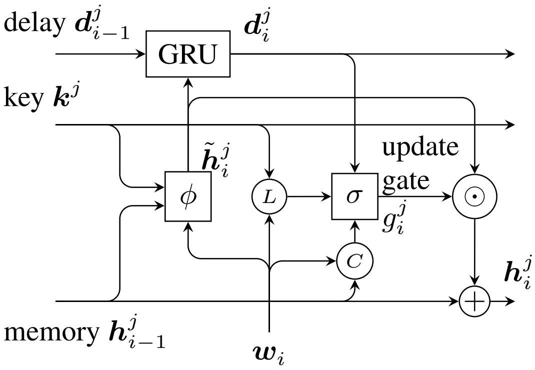

Figure 1: Illustration of our model with a single memory chain at time i. σ, ϕ and GRU represent Equations (2), (3) and (4), while circled nodes L, C, $\odot$ and $+$ depict the location, content terms, Hadamard product, and addition, resp.

Delayed memory update. Update of each memory chain is controlled by a gating mechanism, consisting of three components: the "content" term $\boldsymbol{w}_i \cdot \boldsymbol{h}^j_{i-1}$, the "location" term $\boldsymbol{w}_i \cdot \boldsymbol{k}^j$ and the "delay" term $\boldsymbol{v} \cdot \boldsymbol{d}^j_i$ where $\boldsymbol{d}^j_i$ carries knowledge regarding previous activation of the gate and $\boldsymbol{v}$ is a trainable parameter vector. All three terms may lead to the activation of g^j_i, but differ in how they turn the gate on. While the "location" term causes the gate to open for memory chains whose keys ($\boldsymbol{k}^j$) match the input, the "content" term triggers the activation when the content of the entities ($\boldsymbol{h}^j_{i-1}$) matches the input. The delay term models how and when the gate was turned on in the past with a GRU (Chung et al., 2014) and how past activations should influence the current one.

More formally, with arrows denoting processing direction, the update gate is defined as:

$$\overrightarrow{g}^j_i = \sigma(\boldsymbol{w}_i \cdot \overrightarrow{\boldsymbol{h}}^j_{i-1} + \boldsymbol{w}_i \cdot \boldsymbol{k}^j + \overrightarrow{\boldsymbol{v}} \cdot \overrightarrow{\boldsymbol{d}}^j_i) \quad (2)$$

where $\overrightarrow{g}^j_i$ is the update gate value for the j-th memory at time i,[3] $\boldsymbol{k}^j$ is the embedding for the j-th entity (key), $\overrightarrow{\boldsymbol{h}}^j_{i-1}$ is the hidden memory representation responsible for keeping track of the state of the j-th entity (content), and σ is the sigmoid activation function. The delay recurrence $\overrightarrow{\boldsymbol{d}}^j_i$ is defined as:

$$\overrightarrow{\tilde{\boldsymbol{h}}}^j_i = \phi(\overrightarrow{\boldsymbol{U}}\overrightarrow{\boldsymbol{h}}^j_{i-1} + \overrightarrow{\boldsymbol{V}}\boldsymbol{k}^j + \overrightarrow{\boldsymbol{W}}\boldsymbol{w}_i) \quad (3)$$

$$\overrightarrow{\boldsymbol{d}}^j_i = \overrightarrow{\mathrm{GRU}}(\overrightarrow{\tilde{\boldsymbol{h}}}^j_i, \overrightarrow{\boldsymbol{d}}^j_{i-1}) \quad (4)$$

[3] While $\overrightarrow{g}^j_i$ could instead be a vector for finer-grained control, following Henaff et al. (2017), we use a scalar for simplicity.

where $\overrightarrow{\tilde{\boldsymbol{h}}}_i^j$ is the new candidate memory vector to be incorporated into the existing memory $\overrightarrow{\boldsymbol{h}}_{i-1}^j$ to form the new memory $\overrightarrow{\boldsymbol{h}}_i^j$, ϕ is the parametric ReLU activation function (He et al., 2015), and $\overrightarrow{\boldsymbol{U}}$, $\overrightarrow{\boldsymbol{V}}$ and $\overrightarrow{\boldsymbol{W}}$ are trainable weight matrices.

Once the update gate value has been computed, the j-th memory is then updated according to the intensity of $\overrightarrow{g}_i^j$:

$$\overrightarrow{\mathring{\boldsymbol{h}}}_i^j = \overrightarrow{\boldsymbol{h}}_{i-1}^j + \overrightarrow{g}_i^j \odot \overrightarrow{\tilde{\boldsymbol{h}}}_i^j \tag{5}$$

where $\odot$ is the Hadamard product, and $\overrightarrow{\mathring{\boldsymbol{h}}}_i^j$ is the unnormalised memory representation for the j-th entity.

Essentially, gate $\overrightarrow{g}_i^j$ determines how much the j-th memory should be updated, factoring in three elements: (1) how similar the current input $\boldsymbol{w}_i$ is to the entity being tracked ($\boldsymbol{k}^j$); (2) how related the current input $\boldsymbol{w}_i$ is to the state of the j-th entity ($\overrightarrow{\boldsymbol{h}}_{i-1}^j$); and (3) how past activation should influence the current one. Update of the memory of an entity is only triggered when the gate is activated.

Normalisation. Following the update, the model performs a normalisation step, allowing the memory to forget: $\overrightarrow{\boldsymbol{h}}_i^j = \overrightarrow{\mathring{\boldsymbol{h}}}_i^j / \|\overrightarrow{\mathring{\boldsymbol{h}}}_i^j\|$ where $\|\overrightarrow{\mathring{\boldsymbol{h}}}_i^j\|$ denotes the Euclidean norm of $\overrightarrow{\mathring{\boldsymbol{h}}}_i^j$. As all information stored in $\overrightarrow{\boldsymbol{h}}_i^j$ is constrained to be of unit length, when new information $\overrightarrow{\tilde{\boldsymbol{h}}}_i^j$ is added to the existing memory $\overrightarrow{\boldsymbol{h}}_{i-1}^j$, the cosine distance between the original and updated memory decreases, allowing the model to forget information deemed out-of-date.

Bi-directionality. We apply the above steps both forward and backward over the sentence, enabling the model to capture sentiment terms appearing before and after its associated entity. The memory representation incorporating contexts from both directions is obtained by $\boldsymbol{h}_i^j = \overrightarrow{\boldsymbol{h}}_i^j + \overleftarrow{\boldsymbol{h}}_i^j$, with $\overleftarrow{\boldsymbol{h}}_i^j$ computed analogously to $\overrightarrow{\boldsymbol{h}}_i^j$.

Final classifier. Our model predicts the sentiment polarity $\hat{\boldsymbol{y}}$ to the given target $\boldsymbol{t}$ and aspect $\boldsymbol{a}$ embeddings by incorporating the states of all tracked entities in the form of a weighted sum $\boldsymbol{u}$:

$$p^j = \text{softmax}\left((\boldsymbol{k}^j)^\top \boldsymbol{W}_{att} \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{a} \end{bmatrix} \right) \tag{6}$$

$$\boldsymbol{u} = \sum_j p^j \boldsymbol{h}_m^j \tag{7}$$

where $[\,]$ denotes concatenation, m is sentence length, and $\boldsymbol{W}_{att}$ is a trainable weight matrix. Here, the values of both $\boldsymbol{t}$ and $\boldsymbol{a}$ take the embedding values of their corresponding words (i.e. $\boldsymbol{t}$ and $\boldsymbol{a}$ are drawn from the same embedding matrix as are the input words $\boldsymbol{w}_i$). In the case of multi-word aspect expressions (e.g. TRANSIT-LOCATION), we take the mean of the embeddings of the constituent words. We then transform $\boldsymbol{u}$ to get:

$$\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{R}\phi(\boldsymbol{H}\boldsymbol{u} + \boldsymbol{a})) \tag{8}$$

Training is carried out based on cross entropy loss.

$$\mathcal{L} = \text{CrossEntropy}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \tag{9}$$

Comparision with `EntNet`. While our model is largely inspired by Recurrent Entity Networks (`EntNets`: Henaff et al. (2017)), it differs in three main respects. First, we explicitly model the delay of activation of the update gates g^j with the GRU in Equations (2) and (4) as opposed to making $\boldsymbol{h}_i^j$ implicitly assume the same responsibility in `EntNets`. Admittedly, for `EntNets` on `bAbI` and `CBT`, given the coarse-grained nature and the difference in the granularity of inputs (sentences vs. words), the demand for modelling delayed memory update is less obvious. With this delayed gate activation mechanism, we essentially decouple the duty of capturing transitions of activations between steps from the task of entity state tracking. That is, $\boldsymbol{h}_t^j$ is now dedicated to keeping track of the state of the j-th entity only and released from the burden of monitoring the activation of the update gate. Second, tailoring to the task of TABSA, we incorporate not only the target $\boldsymbol{t}$ but also the aspect $\boldsymbol{a}$ when trying to determine the attention in the softmax function. Third, the proposed model is bi-directional.

3 Experiments

3.1 Experimental Setup

Dataset. To test the effectiveness of our model, we use `Sentihood`, a dataset constructed by Saeidi et al. (2016) for the purpose of detecting aspects and identifying sentiments for each target–aspect pair, consisting of $5,215$ sentences, $3,862$ of which contain a single target, and the remainder multiple targets. Each sentence is annotated with a list of tuples $\{(t, a, y)\}$ with each identifying the sentiment polarity y towards a specific aspect a of

Model	Aspect			Sentiment	
	Acc.	F_1	AUC	Acc.	AUC
LR (Saeidi et al., 2016)	—	39.3	92.4	87.5	90.5
LSTM-Final (Saeidi et al., 2016)	—	68.9	89.8	82.0	85.4
LSTM-Loc (Saeidi et al., 2016)	—	69.3	89.7	81.9	83.9
LSTM+TA+SA (Ma et al., 2018)	66.4	76.7	—	86.8	—
SenticLSTM (Ma et al., 2018)	67.4	78.2	—	89.3	—
EntNet†	66.3	69.8	89.5	87.6	89.7
Our model†	**73.5**	**78.5**	**94.4**	**91.0**	**94.8**

Table 1: Performance on Sentihood. We take the results reported in Saeidi et al. (2016) and Ma et al. (2018), resp; **Bold** = best performance; "—" = not reported; † = average performance over 5 runs.

a given target t in s. Ultimately, given a sentence s, we are interested in both detecting the mention of an aspect a for target t (a label other than *none*), and also identifying the specific sentiment y w.r.t. the target–aspect pair. A detailed description of the task is presented in Section 2.

Model configuration. We initialise our model with GloVe (300-D, trained on 42B tokens, 1.9M vocab, not updated during training: Pennington et al. (2014)) [4] and pre-process the corpus with tokenisation using NLTK (Bird et al., 2009) and case folding. Training is carried out over 800 epochs with the FTRL optimiser (McMahan et al., 2013) and a batch size of 128 and learning rate of 0.05. We use the following hyper-parameters for weight matrices in both directions: $\boldsymbol{R} \in \mathbb{R}^{300\times3}$, $\boldsymbol{H}$, $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$ are all matrices of size $\mathbb{R}^{300\times300}$, $\boldsymbol{v} \in \mathbb{R}^{300}$, and hidden size of the GRU in Equation (4) is 300. Dropout is applied to the output of ϕ in the final classifier (Equation (8)) with a rate of 0.2. Moreover, we employ the technique introduced by Gal and Ghahramani (2016) where the same dropout mask is applied to the input $\boldsymbol{w}_i$ at every step with a rate of 0.2. Lastly, to curb overfitting, we regularise the last layer (Equation (8)) with an L_2 penalty on its weights: $\lambda\|\boldsymbol{R}\|$ where $\lambda = 0.001$.

We empirically set the number of memory chains to 6, with the keys of two of them set to the same embeddings as the target words *LOC1* and *LOC2*, resp., and the other 4 chains with free key embeddings which are updated during training, and therefore free to capture any entities.[5]

Consistent with Saeidi et al. (2016), we tackle the data unbalanced problem (*none* $\gg$ *positive* + *negative*) by sampling the same number of training instances within a batch randomly from each class.

Evaluation. We benchmark against baseline systems presented in the works of Saeidi et al. (2016) and Ma et al. (2018): (1) LR: a logistic regression classifier with n-gram and POS tag features; (2) LSTM-Final: a biLSTM taking the final states as representations; (3) LSTM-Loc: a biLSTM taking the states at the location where target t is mentioned as representations; (4) LSTM+TA+SA: a biLSTM equipped with complex target and sentence-level attention mechanisms; (5) SenticLSTM: an improved version of (4) incorporating the SenticNet external knowledge base (Cambria et al., 2016). We additionally implement a bi-directional EntNet with the same hyper-parameter settings and GloVe embeddings as our model (Henaff et al., 2017).

In terms of evaluation, we adopt the standard 70/10/20 train/validation/test split, and report the test performance corresponding to the model with the best validation score. Following Saeidi et al. (2016), we consider the top 4 aspects only (GENERAL, PRICE, TRANSIT-LOCATION, and SAFETY) and employ the following evaluation metrics: macro-average F_1 and AUC for aspect detection ignoring the *none* class, and accuracy and macro-average AUC for sentiment classification. Following Ma et al. (2018), we also report strict accuracy for aspect detection, as the fraction of sentences where all aspects are detected correctly.

[4] http://nlp.stanford.edu/data/glove.42B.300d.zip

[5] In line with the findings of Henaff et al. (2017) that tying key vectors damages model performance, we observed similar performance deterioration when using tied keys only. While we also experimented with various configurations (all tied vs. all free), this hybrid setup results in the best performance on the validation set.

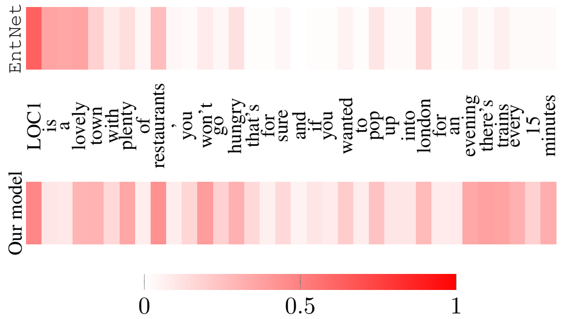

Figure 2: Example of the gate value g_t averaged across memory chains, forward and backward, in EntNet vs. our model.

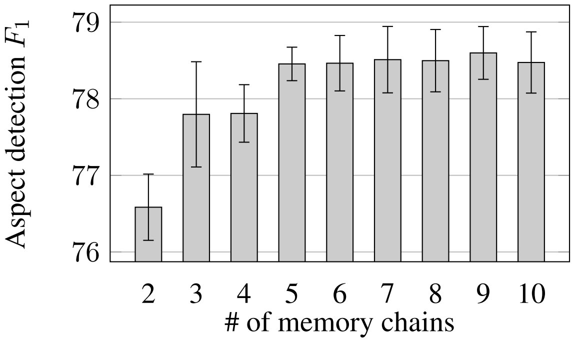

Figure 3: Sensitivity study of model performance to # of memory chains n. Note that we report average performance over 5 runs with standard deviation.

3.2 Results

The experimental results are presented in Table 1.

State-of-the-art results. Our model achieves state-of-the-art results for both aspect detection and sentiment classification. It is impressive that the proposed model, equipped only with domain-independent general-purpose GloVe embeddings, outperforms SenticLSTM, an approach heavily reliant on external knowledge bases and domain-specific embeddings.

EntNet vs. our model. We see consistent performance gains for our model in both aspect detection and sentiment classification, compared to EntNet, esp. for aspect detection, underlining the benefit of delayed update gate activation.

3.3 Discussion

To better understand what the model has learned, we visualise the average gate value g_t in Figure 2, where colour intensity indicates how much memory is updated. Observe that, while updated less by the mention of *LOC1*, our model carries out memory updates upon seeing *lovely town* and *plenty of restaurants*, key phrases associated with aspects such as GENERAL and DINNING. Perhaps even more importantly, despite the distance between *LOC1* and the final portion of the sentence, our model recognises the relevance to TRANSIT-LOCATION and keeps the update gates open to track this particular aspect, as opposed to EntNet where the last phase is overlooked. The ultimate prediction for the TRANSIT-LOCATION aspect of *LOC1* is correct with our model (*positive*), but not detected by EntNet (*none*), resulting in a false negative. More interestingly, with EntNet, once distant from a target, it can be frequently observed that the activation rate of g_t tends to drop, a tendency not so apparent with our model.

In Figure 3, we further study the sensitivity of model performance to the number of memory chains n (2 of which are constrained to track *LOC1* and *LOC2*, the rest are unconstrained chains). Observe that, when $n < 5$, the model suffers from insufficient capacity (not enough memory chains) to capture the various aspects required by the task, with aspect detection F_1 remaining below 78. In particular, when $n = 2$ (no unconstrained chains), model performance drops substantially to a F_1 of 76.6 ± 0.4. Once $n \geq 5$, aspect detection F_1 increases to around 78, and is quite stable even with as many as $n = 10$ chains.

4 Conclusion

In this paper, we have proposed a model which is capable of dynamically tracking entities with a delayed memory update mechanism, and demonstrated the effectiveness of the method over the task of targeted aspect-based sentiment analysis.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback, and gratefully acknowledge the support of Australian Government Research Training Program Scholarship and National Computational Infrastructure (NCI Australia). This work was also supported in part by the Australian Research Council.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Bjoern Schuller. 2016. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*. Osaka, Japan, pages 2666–2677.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Copenhagen, Denmark, pages 452–461.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop*. Montréal, Canada.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS 2016)*. Barcelona, Spain, pages 1027–1035.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV 2015)*. Washington, DC, USA, pages 1026–1034.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking the world state with recurrent entity networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*. Toulon, France.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. In *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*. San Juan, Puerto Rico.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland, pages 437–442.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *Proceedings of the 32rd AAAI Conference on Artificial Intelligence (AAAI 2018)*. New Orleans, USA.

H. Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. 2013. Ad click prediction: A view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2013)*. Chicago, USA, pages 1222–1230.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*. Doha, Qatar, pages 1532–1543.

Marzieh Saeidi, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel. 2016. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*. Osaka, Japan, pages 1546–1556.

Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Query-reduction networks for question answering. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*. Toulon, France.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective lstms for target-dependent sentiment classification. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*. Osaka, Japan, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*. Austin, USA, pages 214–224.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. DCU: Aspect-based polarity classification for SemEval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland, pages 223–229.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*. San Diego, USA.