Attentive Gated Lexicon Reader with Contrastive Contextual Co-Attention for Sentiment Classification

Yi Tay[†,*] **Luu Anh Tuan**[ψ*], **Siu Cheung Hui**[φ], **Jian Su**[δ]
[†]ytay017@e.ntu.edu.sg
[ψ]at.luu@i2r.a-star.edu.sg
[φ]asschui@ntu.edu.sg
[δ]sujian@i2r.a-star.edu.sg
[†,φ]School of Computer Science and Engineering, Nanyang Technological University
[ψ,δ]A*Star, Institute for Infocomm Research, Singapore

Abstract

This paper proposes a new neural architecture that exploits readily available sentiment lexicon resources. The key idea is that that incorporating a word-level prior can aid in the representation learning process, eventually improving model performance. To this end, our model employs two distinctly unique components, i.e., (1) we introduce a lexicon-driven contextual attention mechanism to imbue lexicon words with long-range contextual information and (2), we introduce a contrastive co-attention mechanism that models contrasting polarities between all positive and negative words in a sentence. Via extensive experiments, we show that our approach outperforms many other neural baselines on sentiment classification tasks on multiple benchmark datasets.

1 Introduction

Across the rich history of sentiment analysis research (Kim and Hovy, 2004; Liu, 2012; Pang et al., 2008), sentiment lexicons have been extensively used as features for sentiment classification tasks. Lexicons, either handcrafted or algorithmically generated, consist of words and their associated polarity score. For instance, lexicons assign a high positive score for the word '*excellent*' but a negative score for the word '*terrible*'. Traditionally, the summation of lexicon scores has been treated as a reasonable heuristic estimate (or feature) that is capable of supporting opinion mining applications. Throughout the years, plenty of lexicon lists have been built for various specific domains or general purposes (Hu and Liu, 2004; Mohammad et al., 2013; Wilson et al., 2005). They are indeed valuable resources that should be exploited.

However, sentiment lexicons are in reality hardly useful without context. After all, the complexity and ambiguity of natural language pose great challenges for the crude bag-of-words generalization of lexicons. Firstly, the concept of semantic compositionality is non-existent in simple lexicon approaches which raises problems when handling flipping negation (*not happy*), content word negation (*ameliorates pain*) or unbounded dependencies (*no body passed the exam*). Secondly, lexicons also do not handle word sense, e.g., not being able to differentiate the meaning of *hot* in the phrases *'a hot, attractive person'* and a *'a scorching hot day'*. Thirdly, simple summation over lexicon scores cannot deal with sentences with double contrasting polarities, e.g., the lexicon polarity score of *'Thanks for making this uncomfortable situation more comfortable'* becomes negative because *uncomfortable* has a higher negative lexicon score over the positive score of the word *comfortable*. Lastly, strongly positive or negative words may occur in neutral context which forces an inclination of predictions towards a non-neutral polarity. As such, the exploitation of readily available lexicon lists is an inherently challenging task.

Deep learning has demonstrated incredibly competitive performance in many NLP tasks (Liu et al., 2015; Bradbury et al., 2016; Tai et al., 2015). With no exception, the task of sentiment analysis is recently also dominated by neural architectures. It has been proven from the fact that the top systems from SemEval Sentiment analysis challenges (e.g., notably 2016 and 2017) have mainly leveraged the effectiveness of deep learning models. The main advantage of deep learning approach is that it is effective in exploring both linguistic and semantic relations between words, thus can overcomes the problems of lexicon-based approach. However, current deep learning approach

[*] Denotes equal contribution.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3443–3453
Brussels, Belgium, October 31 - November 4, 2018.

for sentiment analysis usually faces with the major shortcoming, i.e., being limited by the quantity of high quality labeled data. Manual labeling of data, however, is costly and require domain expert knowledge which is not always available in practice.

Given the pros and cons of previous two previous approaches, we aim to combine the best of both worlds - the traditional sentiment lexicon and modern deep learning architectures. To the best of our knowledge, the only work that combines the two paradigms within *end-to-end* neural networks is the Lexicon RNN model (Teng et al., 2016). In their approach, sentiment lexicons are extracted from the hidden states of a recurrent neural network and passed through a simple feed-forward neural network to produce a new polarity weight. This approach, however, has some limitations which will be illustrated using the following example:

"Thanks for making this horrible situation at work more bearable."

Firstly, the Lexicon RNN does not consider the interactions between positive or negative lexicon words, which makes it susceptible to misleading strong lexicon priors. In the above example, the word '*horrible*' is a strongly negative word in most lexicons. As a result, the Lexicon RNN (and many other lexicon based approaches in general) will assign a negative polarity to the sentence. Clearly, modeling similarity between two contrasting polarity words ('*horrible*' and '*bearable*') can help the model resolve this confusion. Secondly, the RNN encoder in the Lexicon RNN is restricted by the sequential nature of the recurrent model, resulting in a limited global view of the entire sentence. For example, the word pairs ('*horrible*', '*bearable*') and ('*thanks*', '*bearable*') are useful for detecting the polarity of the sentence but do not have any explicit interaction even with a sequential RNN encoder. Moreover, the word pair ('*thanks*', '*bearable*') is very far apart in the above example sentence, making it challenging for RNN encoders to capture interactions between them. Finally, the Lexicon RNN faces difficulty dealing with more than two classes due to its design, i.e., linear combination of two scalar scores. In order to cope with this weakness, the authors define hardcoded *dataset specific* thresholds for 5-way classification. Adapting this to 3-way (positive, negative and neutral) is cumbersome as thresholds have to be found by either maximizing over the development set or defined heuristically.

In this paper, we introduce a new end to end paradigm that integrates lexicon information into neural network for the task of sentiment analysis. More specifically, instead of learning a lexicon-based score, we propose to learn an auxiliary embedding by exploiting lexicon information. The key motivation behind the auxiliary representation is that compositional learning with prior/global knowledge of positive and negative inclined words can lead to improved representations. Next, a gating mechanism controls the additive blend between this lexicon-based representation and a standard attention-based recurrent model. In essence, this supporting network aims to learn a *'lexicon-based'* view of the sentence and can be interpreted as *'learning to compose'* by exploiting lexicon information. Finally, instead of the combination of two scalar values (the base lexicon score and sentence bias score) as in the Lexicon RNN model, we propose to use the k-class softmax function at the final layer. Intuitively, it is a more natural solution for fine-grained sentiment classification over the cumbersome tuning of ad-hoc threshold values. Our contributions can be summarized as follows:

- We propose to learn an auxiliary embedding by exploiting lexicon information rather than learning a lexicon-based score. Its design is a more natural and flexible solution for k-class sentiment classification.
- We propose a *contextual attention* (CA) mechanism that learns to attend to lexicon words based on the context. Unlike Lexicon RNN which extracts the hidden representations from the recurrent model, contextual attention allows a wider, global and more complete view of the context (sentence) by matching against every single word in the sentence. In addition to semantic *compositionality*, our model also benefits from semantic *similarity*.
- We propose to model the interaction between the positive and negative lexicon words inside the neural network. Positive and negative lexicon words are modeled seperately and subsequently compared using *contrasive co-attention* (CC) which learns the relative importance of positive lexicons with respect

to negative lexicons (and vice versa). Modeling such intricacies between positive and negative words allows our model to deal with scenarios such as contrasting polarities, neutrality and also sarcasm. We also discover that our CC mechanism produces a neutralizing effect which negates misleading attention on words with intense polarity even though the context is neutral.

Overall, we propose AGLR (*Attentive Gated Lexicon Reader*), a new attention-based neural architecture that exploits sentiment lexicons for learning to compose an auxiliary sentence embedding. Our model achieves state-of-the-art performance on several benchmark datasets. Finally, our AGLR, a *single neural model*, also achieves competitive performance with respect to top teams in SemEval runs which are mostly comprised of extensively engineered ensembles.

2 Related Work

Sentiment lexicons have a rich traditional in sentiment analysis research and have been exploited in many statistical methods across the years (Hu and Liu, 2004; Kim and Hovy, 2004; Agarwal et al., 2011; Mohammad et al., 2013; Tang et al., 2014b,a; Teng et al., 2016). It is easy to see how sentiment lexicons are able to benefit opinion mining applications. More specifically, sentiment lexicons form an integral role in the winning solutions of SemEval 2013 (Mohammad et al., 2013) and 2014 (Miura et al., 2014). In many of these these approaches, standard machine learning classifiers (such as Support Vector Machines) are trained on discrete features partly derived from resources such as sentiment lexicon.

In recent years, we see a shift of the state-of-the-art from discrete models to neural models (Socher et al., 2013; Kim, 2014; Dong et al., 2014; Tang et al., 2016; Tai et al., 2015; Ren et al., 2016; Zhang et al., 2016; Teng and Zhang, 2016). This ranges from learning sentiment-specific word embeddings (Tang et al., 2014b; Faruqui et al., 2015) to end-to-end neural architectures (Teng et al., 2016; Angelidis and Lapata, 2017). The winning solution of SemEval 2016 (Deriu et al., 2016) utilized ensembles of convolutional neural networks (CNN). Recurrent-based models such as the bidirectional long short-term memory (BiLSTM) (Hochreiter and Schmidhuber, 1997; Graves et al., 2013) are popular and standard strong baselines for many opinion mining tasks including sentiment analysis (Tay et al., 2017) and sarcasm detection (Tay et al., 2018c). These neural models such as the BiLSTM are capable of modeling semantic compositionality and produce a feature vector which can be used for classification.

To integrate the information of lexicon inside Lexicon RNN model, Teng et al. (2016) proposed to use the hidden representations from a BiLSTM to influence the lexicon score, i.e., learning context-sensitive lexicon features. However, our method can be considered as a vastly different paradigm and instead learns a d-dimensional embedding using neural attention (Bahdanau et al., 2014; Luong et al., 2015) instead of a lexicon score. The key idea of neural attention is that it allows neural networks to look (or attend) to certain words in a sequence. This concept has indeed profoundly impacted the fields of NLP, giving rise to many variant architectures including end-to-end memory networks (Sukhbaatar et al., 2015; Li et al., 2017).

Our approach draws inspiration from memory networks and co-attentive models for machine comprehension (Xiong et al., 2016; Seo et al., 2016). In fact, the auxiliary network can be interpreted as a form of *multi-layered attention* which draws connection to vanilla memory networks. Attending over two sequences (or bidirectional attention) are intuitive approaches for NLP tasks such as information retrieval (Tay et al., 2018b) and generic text matching (Tay et al., 2018a). In our work, we adapt this to model the similarities between (1) lexicon-context and (2) contrasting polarities which borrows inspiration from (Riloff et al., 2013). Since our matching problem is derived from the same sequence (identified by a lexicon prior), this work can be interpreted as a form of self-attention (Vaswani et al., 2017) which draws relations to the intra-attentive model for sarcasm detection (Tay et al., 2018c).

3 Attentive Gated Lexicon Reader

In this section, we describe our proposed deep learning model for sentiment classification. The key idea of our model is to generate two representations, i.e., a lexicon-based auxiliary embedding of the sentence and also a generic compositional representation of the sentence. The former is generated via a supporting network that consists of contextual attention and contrastive

co-attention layers. The latter is generated by a vanilla attention-based BiLSTM model. A gating mechanism then combines them for prediction.

3.1 Embedding Layer

Firstly, we extract all lexicon words from the input sequence and then separately[1] denote them as positive or negative words. Overall, our model accepts three sequences as an input. (1) the original sentence, (2) a list of positive lexicon words found in the sentence and (3) a list of negative lexicon words found in the sentence. The three sequences are indexed into a word embedding layer $W \in \mathbb{R}^{|V| \times d}$ which outputs three matrices $S \in \mathbb{R}^{d \times L_s}$ (sentence embeddings), $P \in \mathbb{R}^{d \times G_p}$ (positive lexicon embeddings) and $N \in \mathbb{R}^{d \times G_n}$ (negative lexicon embeddings). d is the dimensionality of the word embeddings and L_s, G_p and G_n are the maximum sequence lengths of sentence, positive lexicon and negative lexicon respectively.

3.2 Learning Sentence Representation

To learn sentence representations of the input sequence, we pass $S = (w_1, w_2 \cdots w_{L_s})$ into a Bidirectional Long Short-Term Memory (LSTM) layer. As such, the output of the BiLSTM is described as follows:

$$h_t = BiLSTM(h_{t-1}, w_t) \tag{1}$$

where h_t is the hidden representation at step t. Given a sequence of inputs $w_1, w_2 \cdots w_L$, the output of the BiLSTM layer is a sequence of hidden states $h_1, h_2 \cdots h_L$. Note that since we use a bidirectional LSTM, then $h_t \in \mathbb{R}^{2r}$ where r is the dimensionality of the BiLSTM layer. In our case r is set to $\frac{d}{2}$ such that the output vector has dimensionality d.

Sentence Attention To learn a final sentence representation of the sentence, we adopt an attention mechanism. The attention mechanism is defined by the following equations:

$$\mathbf{Y} = \tanh(\mathbf{W}_y \, H) \tag{2}$$

$$a_c = softmax(w_y^T \, \mathbf{Y}) \text{ and } s = H \, a_c^T \tag{3}$$

where $s \in \mathbb{R}^d$ is the output sentence representation, $\mathbf{W}_y \in \mathbb{R}^{d \times d}$ and $w_y \in \mathbb{R}^d$ are parameters of the attention layer. Intuitively, the attention layer learns to pay attention to important segments of the sentence, producing a weighted representation of the hidden states of the BiLSTM layer.

3.3 Learning Auxiliary Lexicon Embedding

This layer aims to learn a single d-dimensional lexicon-based representation of the sentence. In order to learn the lexicon embedding, our model adopts a two layer attention mechanism, namely the contextual attention (CA) and contrastive co-attention (CC).

Contextual Attention (CA) We utilize an attention mechanism to learn the relative importance of each lexicon word based on the sentence representation. This layer is applied to and is functionally identical for both P and N. As such, for notational convenience, we use Q to represent either positive (P) or negative (N), and G to represent the maximum number of lexicon words. Let $Q \in \mathbb{R}^{G \times d}$ be a sequence of lexicon words and $H \in \mathbb{R}^{L_s \times d}$ be the intermediate hidden representations obtained from the contextual BiLSTM layer:

$$M = tanh(Q \, \mathbf{U} \, H^T) \tag{4}$$

where $\mathbf{U} \in \mathbb{R}^{d \times d}$ are the parameters of this layer. Next, we apply a column-wise max pooling of M. The key idea is to generate an attention vector:

$$a = sm(\max_{col}(M)) \; ; \; c_i = a_i * q_i \tag{5}$$

where $a \in \mathbb{R}^G$. The softmax function normalizes the values of the vector $\max_{col}(M)$ into a probability distribution. To learn the context-sensitive weight importance of each lexicon word, we then apply the attention vector on Q. $C = \{c_1, c_2 \cdots c_G\}$ is the context-sensitive lexicon representation of Q. Intuitively, the CA mechanism attends to each lexicon word based on its *maximum* influence on each word of the main sentence. There are several advantages to our context attention mechanism. Unlike Lexicon RNN which simply extracts the hidden representation (generated from BiLSTM) of the lexicon word, our approach has a global view of the entire sentence which allows each lexicon word to benefit from wider contextual knowledge as opposed to being limited to the temporal compositionality provided by the BiLSTM layer. Overall, the outputs of this layer are two matrices (positive and negative lexicon embeddings) which are context-sensitive.

[1]For our experiments, we mainly use ST140 lexicon and therefore use $score > 0$ to separate positive and negative words. Notably, about $\approx 85\%$ of all words in the sentence has a lexicon assignment.

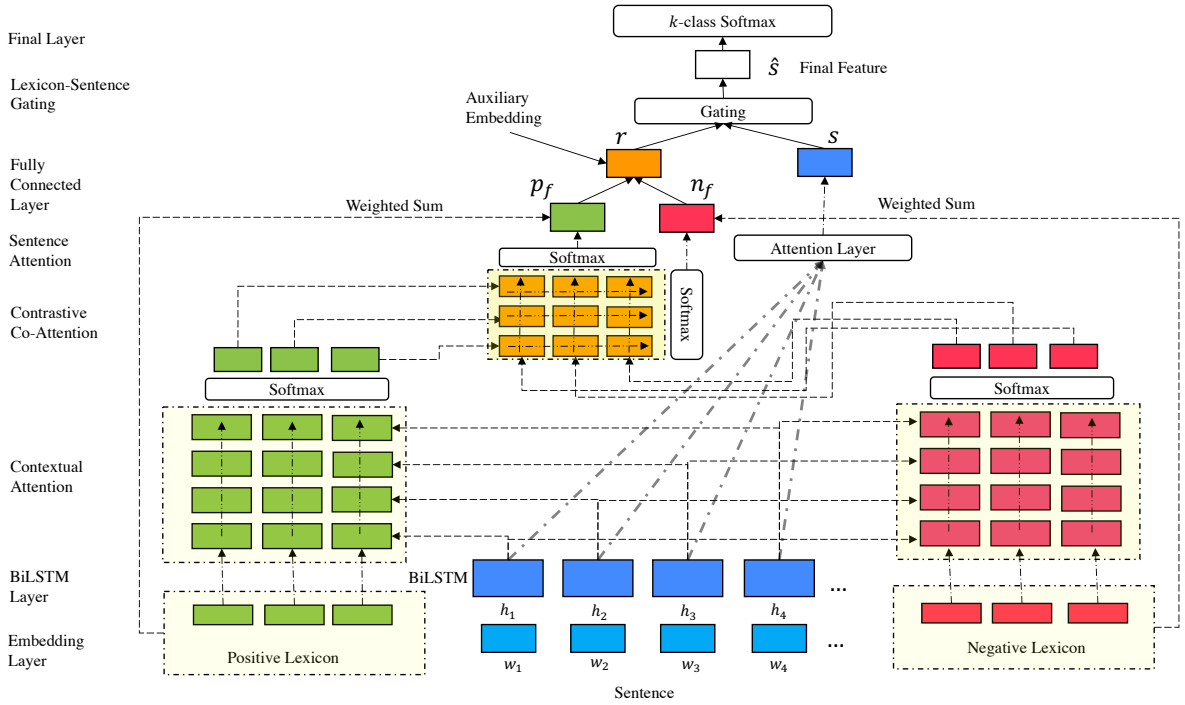


Figure 1: Illustration of our proposed Attentive Gated Lexicon Reader model (*best viewed in color*).

Note that these lexicon embeddings retain their dimensionality passing through this layer.

Contrastive Co-Attention (CC) This layer aims to model the contrast between polarities. Intuitively, this layer helps to model sentences with double or conflicting polarities. It also aims to negate strongly positive or negative words in the case of a neutral context. In order to do so, we employ a contrastive co-attention model that learns to weight the relative importance of each positive lexicon word based on the negative lexicon (and vice versa). We accept the contextualized positive and negative lexicon embeddings from the previous layer as an input. Let $\hat{P} \in \mathbb{R}^{G \times d}$ be the contextualized positive lexicons and $\hat{N} \in \mathbb{R}^{G \times d}$ be the contextualized negative lexicons, our co-attention layer learns a soft attention alignment between positive and negative lexicon embeddings. Similar to our contextual attention layer, we first learn an affinity matrix Z that models the relationship between positive and negative lexicon embeddings:

$$Z = tanh(P \, \mathbf{A} \, N^T) \tag{6}$$

Next, we apply both column-wise and row-wise max-pooling on the affinity matrix Z to obtain two attention vectors. The two attention vectors are then normalized with the softmax function (denoted as sm).

$$a_p = sm(\max_{col}(Z)) \; ; \; a_n = sm(\max_{row}(Z)) \tag{7}$$

a_p is the attention vector for the positive lexicon embeddings and a_n is the attention vector for the negative lexicon embeddings. The final vector representations are therefore:

$$p_f = P \, a_p^\top \; ; \; n_f = N \, a_n^\top \tag{8}$$

where $p_f \in \mathbb{R}^d$ and $n_f \in \mathbb{R}^d$ are the vector representations for positive lexicon and negative lexicon respectively. Note that this layer, unlike the contextual attention layer, is named 'co-attention' because both P and N are both 'attended over' concurrently. It is also good to note that attentions are applied over the original embeddings P, N and not the *contextualized* embeddings $\hat{P}, \hat{N}$.

Fully-Connected Layer Next, we pass the concatenation of p and n through a fully-connected layer to learn the final representation for the auxiliary lexicon embedding, i.e., $r = tanh(\mathbf{W}_h \, ([p; n]) + b_h)$ where $\mathbf{W}_h \in \mathbb{R}^{2d \times d}$ are the parameters of the hidden layer and b_h is the bias value. The output $r \in \mathbb{R}^d$ is the final auxiliary lexicon-based embedding.

Learning Final Representations To combine the lexicon-based representation with the sentence

representation, we adopt a gating mechanism.

$$\hat{s} = \sigma(w_g \odot r) \odot r + (1 - \sigma(w_g \odot r)) \odot s \quad (9)$$

where $w_g \in \mathbb{R}^d$ are the parameters of this layer, σ is the sigmoid function. $\hat{s}$ is the overall final representation.

3.4 Final Layer and Optimization

Finally, we pass $\hat{s}$ the overall final representation into a softmax layer.

$$y = softmax(\mathbf{W}_f \hat{s} + b_f) \quad (10)$$

where $y \in \mathbb{R}^k$, where k is the number of classes (2 for positive and negative and 3 including neutral). $\mathbf{W}_f$ and b_f are standard parameters of a linear regression layer. For optimization, we adopt the standard cross entropy loss function with L2 regularization.

$$L = -\sum_{i=1}^{N} [y_i \log o_i + (1 - y_i) \log(1 - o_i)] + R \quad (11)$$

where o is the output of the softmax layer and $R = \lambda \|\psi\|_2^2$ is the L2 regularization.

4 Empirical Evaluation

This section describes our empirical experiments.

4.1 Evaluation Procedure

In this section, we describe the datasets used, evaluation metric and implementation details.

Datasets We conduct our experiments[2] on subsets of sentiment analysis benchmarks from SemEval 2013 (Nakov et al., 2013), SemEval 2014 (Rosenthal et al., 2014) and SemEval 2016 (Nakov et al., 2016). More specifically, we focus on the sentence level of sentiment analysis and evaluate on the datasets of SemEval 2013 task 2, SemEval 2014 task 9 and SemEval 2016 task 4, which we will name as *SemEval13*, *SemEval14* and *SemEval16* respectively in this section. For fair comparison, we use the same setting of training, development and testing as in SemEval competitions. To further evaluate the performance of methods when data is limited, for *SemEval16*, we experiment on two different training settings. The first, TRAIN, uses only the 2016 training set while the other, TRAIN-ALL, appends the 2013 training set to the 2016 training set, following the official setting of SemEval 2016 while TRAIN explores the setting where training data is limited.

Evaluation Metrics We evaluate on two settings, i.e., 3-way (positive, negative and neutral) and also binary (positive and negative) classification. We report the accuracy and macro-averaged F1 score for all settings.

Compared Baselines In this section, we list the neural baselines we use for comparisons.

- **NBOW-MLP** (Neural Bag-of-Words + Multi-layered Perceptron) is a simple sum of all word embeddings which is connected to a 2-layer MLP of 100 dimensions.
- **CNN** (Convolutional Neural Network) is another popular neural encoder for learning sentence representations. We use a filter size of 3 and 150 filters.
- **BiLSTM** (Bidirectional Long Short-Term Memory) is a standard strong neural baseline for many NLP tasks. The size of the LSTM is set to 150.
- **AT-BiLSTM** (Attention-based BiLSTM) is an extension of the BiLSTM model with neural attention.
- **Lexicon RNN** (Lexicon Recurrent Neural Network) is the model of (Teng et al., 2016). The first neural model that incorporates sentiment lexicon. The size of the BiLSTM in this model is also set to 150.

All models except Lexicon RNN optimize the softmax cross entropy loss. The authors use Lexicon RNN for binary and 5-way classification. In order to adapt Lexicon RNN to 3-way classification (positive, negative, neutral), we adapt the 5-way formulation that minimizes the MSE (mean square error) loss to 3-way. The output is scaled[3] to $s \in [-1, 1]$ where $s > 0.25$ is treated as positive, $s < -0.25$ is treated as negative and everything in between is neutral.

[2] SemEval 2015 was omitted due to space in favor of SemEval 2016 since testing sets are significantly larger in the latter.

[3] We experimented with other thresholds but found 0.25 to work the best.

	SemEval13				*SemEval14*					
	3-way		Binary		3-way		Binary		AVG	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
NBOW-MLP	65.18	60.94	85.44	82.30	65.68	60.35	89.44	81.60	76.44	71.30
CNN	71.41	68.23	85.74	82.60	70.05	66.22	89.86	82.09	79.27	74.79
BiLSTM	72.06	70.00	85.89	82.79	71.62	68.34	90.20	83.09	79.94	76.06
AT-BiLSTM	72.21	69.89	86.13	83.22	71.83	68.01	90.20	83.46	80.09	76.15
Lexicon RNN	69.97	68.69	86.43	83.54	70.75	67.06	**91.13**	**84.60**	79.57	75.97
AGLR	**73.27**	**71.79**	**86.72**	**84.18**	**73.29**	**70.48**	90.37	84.15	**80.91**	**77.65**

Table 1: Experimental results on test datasets *SemEval2013* and *SemEval2014*.

	Sem2016 (TRAIN)				*Sem2016* (TRAIN-ALL)					
	3-way		Binary		3-way		Binary		AVG	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
NBOW-MLP	54.31	52.90	79.69	77.33	61.09	55.91	84.90	82.01	70.00	67.04
CNN	54.67	52.17	79.79	75.28	62.68	57.71	84.10	81.31	70.31	66.62
BiLSTM	55.57	52.33	81.90	77.12	63.26	60.31	85.89	84.14	71.66	68.50
AT-BiLSTM	56.95	54.53	80.09	73.93	64.20	61.64	86.77	83.67	72.00	68.44
Lexicon RNN	51.02	50.45	81.72	79.00	61.41	60.50	86.68	83.82	70.21	68.44
AGLR	**59.01**	**56.67**	**82.79**	**80.10**	**66.62**	**64.36**	**87.16**	**84.98**	**73.90**	**71.53**

Table 2: Experimental results on *Sem2016* with two training settings TRAIN and TRAIN-ALL.

Implementation Details Text are lowercased, tokenized using NLTK's tweet tokenizer and padded to the maximum sequence length of the dataset. For Lexicon RNN and AGLR, we use[4] the ST140 (Sentiment140) lexicon which was created by distant supervision (Mohammad et al., 2013). The maximum numbers of positive lexicons and negative lexicons extracted per sample are tuned amongst $\{5, 8, 10\}$. Models are trained for a maximum of 30 epochs with early stopping if the performance on the development set does not improve after 5 epochs. Results reported are the test scores from the model that performed best on the development set. The batch size is tuned amongst $\{50, 100, 300\}$. L2 Regularization tuned amongst $\{10^{-6}, 10^{-7}, 10^{-8}\}$. Dropout is set to 0.5. We optimized all networks with the RMSprop optimizer and with initial learning rate tuned amongst $\{0.01, 0.005, 0.001\}$. Word embeddings are initialized with Glove 27B (Pennington et al., 2014) ($d = 200$) trained on tweets and are trainable parameters. The size of the BiLSTM is $d = 100$.

4.2 Experimental Results

Table 1 and Table 2 report the results of our experiments. The results on TRAIN-ALL are higher than TRAIN for *SemEval16* in lieu of the larger dataset. Firstly, we observe that our proposed AGLR outperforms all neural baselines on 3-way classification. The overall performance of AGLR achieves state-of-the-art performance. On average, AGLR outperforms Lexicon RNN and AT-BiLSTM by $1\% - 3\%$ in terms of F1 score. We also observe that AGLR always improves AT-BiLSTM which ascertains the effectiveness of learning auxiliary lexicon embeddings. The key idea here is that the auxiliary lexicon embeddings provide a *different view* of the sentence which supports the network in making predictions.

We also observe that Lexicon RNN does not handle 3-way classification well. Even though it has achieved good performance on binary classification, the performance on 3-way classification is lackluster (the performance of AGLR outperforms Lexicon RNN by up to 8% on *SemEval16* TRAIN). This could also be attributed to the MSE based loss function. Conversely, by learning an auxiliary embedding (instead of a scalar score), our model becomes more flexible at the final layer and can be adapted to using a k softmax function. Finally, we observe that BiLSTM and AT-BiLSTM outperform Lexicon RNN on average with Lexicon RNN being slightly better on binary classification.

[4]We also used Bing Liu's opinion lexicon but found it to perform slightly worse.

Comparisons against Top SemEval Systems Table 3 reports the results of our proposed approach against the top team of each SemEval run, i.e., *NRC-Canada* (Mohammad et al., 2013) for 2013 Task 2, *Team-X* (Miura et al., 2014) for 2014 Task 9, *SwissCheese* (Deriu et al., 2016) for 2016 Task 4. We follow the exact training datasets allowed for each SemEval run. Following the competition setting, with the exception of accuracy for SemEval 2016, all metrics reported are the macro averaged F1 score of positive and negative classes.

		Top System	Ours
SemEval13	Tweets	69.02	**70.10**
SemEval14	Tweets	70.96	**71.11**
	Sarcasm	56.50	**58.87**
	LiveJournal	69.44	**72.52**
SemEval16	Tweets	**63.30**	61.90
	Tweets (Acc)	64.60	**66.60**

Table 3: Comparisons against top SemEval systems. Results reported are the F^{PN} metric scores used in the SemEval tasks.

We observe that AGLR achieves competitive performance relative to the top runs in SemEval 2013, 2014 and 2016. It is good to note that SemEval approaches are often heavily engineered containing ensembles and many handcrafted features which include extensive use of sentiment lexicons, POS tags and negation detectors. Recent SemEval runs gravitate towards neural ensembles. For instance, the winning approach for *SwissCheese* (SemEval 2016) uses an ensemble of 6 CNN models along with a meta-classifier (random forest classifier). On the other hand, our proposed model is *a single neural model*. In addition, SwissCheese also uses emoticon-based distant supervision which exploits a huge corpus of sentences (millions) for training. Conversely, our approach only uses the 2013 and 2016 training sets which are significantly smaller. Given these conditions, we find it remarkable that our single model is able to achieve competitive performance relative to the extensively engineered approach of SwissCheese. Moreover, we actually outperform significantly in terms of pure accuracy. AGLR performs competitively on SemEval 2013 and 2014 as well. The good performance on the *sarcasm* dataset could be attributed to our contrastive attention mechanism.

Ablation Study In this section, we study the impacts and contribution of the different components of our model. Specifically, we tested 3 settings. The first, we removed CC only. In this case, positive and negative lexicons are *summed* instead of a *weighted summed* using attention. In the next setting, we removed CA only. Similarly, embeddings are summed instead of attentively summed. Finally, we removed both CA and CC. In this case, all lexicons are considered neural bag-of-words (NBOW) and passed to the MLP layer. Table 4 shows the results of this ablation study on *SemEval16* using the TRAIN-ALL setting.

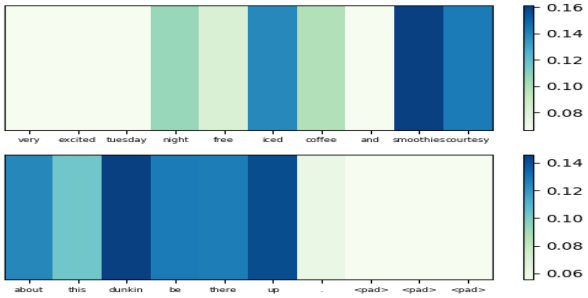
Model	Acc	F1
AT-BiLSTM only	64.20 (-2.42)	61.64 (-2.96)
AGLR (*-CA and -CC*)	62.42 (-4.20)	59.48 (-4.88)
AGLR (*-CA*)	65.81 (-0.81)	60.47 (-3.89)
AGLR (*-CC*)	64.38 (-2.24)	61.26 (-3.10)
AGLR	66.62	64.36

Table 4: Ablation study on *SemEval16* (TRAIN-ALL)

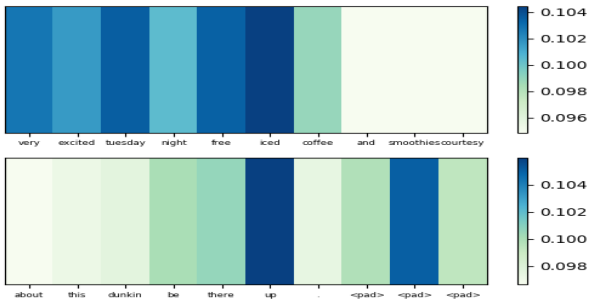
It is clear that both CC and CA are critical to the performance of AGLR. Removing either or both can cause performance to degrade. In particular, we also observe that CA seems to be less important than CC, i.e., performance drops more as compared to removing CA. We also note that removing both and a simple NBOW for lexicons can degrade performance since the base AT-BiLSTM is better than using NBOW lexicons as an auxiliary support. As such, the design of the auxiliary embeddings must be treated with care.

Qualitative Analysis In order to study what are the specific roles of the contextual and contrastive attention mechanism, we inspect the attention maps over the positive and negative lexicons. We use the following example in which the ground truth label is *positive*: *"Very excited about Tuesday night @user free iced coffee and smoothies courtesy of Dunkin Donuts will be set up."*. Figure 2a shows the attention maps for contextual attention. We observe that contextual attention focuses more on the context, i.e., focusing on words such as 'night', 'iced coffee' and 'smoothies'. On the other hand, Figure 2b shows the attention maps after contrastive attention. We observe that contrastive attention learns more polarity specific attentions, i.e., shifting some focus to *'very excited'*. We also observe that the contrastive attention tends to shift its attention weights to less meaningful words for the negative lexicon if the ground truth label is positive (and vice versa). We

believe that this indicates that there is an absence of negative sentiment.



(a) Attention over positive and negative lexicons for contextual attention.



(b) Attention over positive and negative lexicons for contrastive attention.

Figure 2: Visualization of Contextual Attention and Contrastive Co-Attention.

5 Conclusion

We proposed a novel method of incorporating lexicons into neural models for the task of sentiment analysis. More specifically, we learn an auxiliary lexicon embedding using neural attention. Our proposed model AGLR achieves an overall state-of-the-art performance on multiple benchmark datasets outperforming strong neural baselines such as AT-BiLSTM and Lexicon RNN. The performance of AGLR is also competitive relative to top SemEval systems which utilized neural ensembles or very extensive feature engineering.

References

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media*. Association for Computational Linguistics, pages 30–38.

Stefanos Angelidis and Mirella Lapata. 2017. Multiple instance learning networks for fine-grained sentiment analysis. *arXiv preprint arXiv:1711.09645* .

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2016. Quasi-recurrent neural networks. *CoRR* abs/1611.01576.

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision.

Li Dong, Furu Wei, Chuanqi Tan, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, pages 6645–6649.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*. pages 168–177. https://doi.org/10.1145/1014052.1014073.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 1367.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .

Cheng Li, Xiaoxiao Guo, and Qiaozhu Mei. 2017. Deep memory networks for attitude identification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*. pages 671–680.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5(1):1–167.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of*

the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. pages 1433–1443. http://aclweb.org/anthology/D/D15/D15-1168.pdf.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 2014. Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data.

Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242* .

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *SemEval@ NAACL-HLT*. pages 1–18.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. Semeval-2013 task 2: Sentiment analysis in twitter. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. volume 2, pages 312–320.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543.

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation.

Sara Rosenthal, Alan Ritter, Preslav Nakov, and Veselin Stoyanov. 2014. Semeval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pages 73–80. http://www.aclweb.org/anthology/S14-2009.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* .

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. Citeseer.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pages 2440–2448.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective lstms for target-dependent sentiment classification. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. pages 3298–3307.

Duyu Tang, Furu Wei, Bing Qin, Ming Zhou, and Ting Liu. 2014a. Building large-scale twitter-specific sentiment lexicon: A representation learning approach.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014b. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. volume 1, pages 1555–1565.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2017. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. *arXiv preprint arXiv:1712.05403* .

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018a. Hermitian co-attention networks for text matching in asymmetrical domains. In *IJCAI*. pages 4425–4431.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018b. Multi-cast attention networks for retrieval-based question answering and response prediction. *arXiv preprint arXiv:1806.00778* .

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018c. Reasoning with sarcasm by reading in-between. *arXiv preprint arXiv:1805.02856* .

Zhiyang Teng, Duy-Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 1629–1638. http://aclweb.org/anthology/D/D16/D16-1169.pdf.

Zhiyang Teng and Yue Zhang. 2016. Bidirectional tree-structured lstm with head lexicalization. *arXiv preprint arXiv:1611.06788* .

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. pages 5998–6008.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*. pages 347–354. http://aclweb.org/anthology/H/H05/H05-1044.pdf.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *CoRR* abs/1611.01604.

Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis.