Attention and Lexicon Regularized LSTM for Aspect-based Sentiment Analysis

Lingxian Bao
Universitat Pompeu Fabra
lingxian.bao@upf.edu

Patrik Lambert
Universitat Pompeu Fabra
patrik.lambert@upf.edu

Toni Badia
Universitat Pompeu Fabra
toni.badia@upf.edu

Abstract

Attention based deep learning systems have been demonstrated to be the state of the art approach for aspect-level sentiment analysis, however, end-to-end deep neural networks lack flexibility as one can not easily adjust the network to fix an obvious problem, especially when more training data is not available: e.g. when it always predicts *positive* when seeing the word *disappointed*. Meanwhile, it is less stressed that attention mechanism is likely to "over-focus" on particular parts of a sentence, while ignoring positions which provide key information for judging the polarity. In this paper, we describe a simple yet effective approach to leverage lexicon information so that the model becomes more flexible and robust. We also explore the effect of regularizing attention vectors to allow the network to have a broader "focus" on different parts of the sentence. The experimental results demonstrate the effectiveness of our approach.

1 Introduction

Sentiment analysis (also called opinion mining) has been one of the most active fields in NLP due to its important value to business and society. It is the field of study that tries to extract opinion (*positive, neutral, negative*) expressed in natural languages. Most sentiment analysis works have been carried out at document level (Pang et al., 2002; Turney, 2002) and sentence level (Wilson et al., 2004), but as opinion expressed by words is highly context dependent, one word may express opposite sentiment under different circumstances. Thus aspect-level sentiment analysis (ABSA) was proposed to address this problem. It finds the polarity of an opinion associated with a certain aspect, such as *food, ambiance, service,* or *price* in a restaurant domain.

Although deep neural networks yield significant improvement across a variety of tasks compared to previous state of the art methods, end-to-end deep learning systems lack flexibility as one cannot easily adjust the network to fix an obvious problem: e.g. when the network always predicts *positive* when seeing the word *disappointed*, or when the network is not able to recognize the word *dungeon* as an indication of *negative* polarity. It could be even trickier in a low-resource scenario where more labeled training data is simply not available. Moreover, it is less stressed that attention mechanism is likely to over-fit and force the network to "focus" too much on a particular part of a sentence, while in some cases ignoring positions which provide key information for judging the polarity. In recent studies, both Niculae and Blondel (2017) and Zhang et al. (2019) proposed approaches to make the attention vector more sparse, however, it would only encourage the over-fitting effect in such scenario.

In this paper, we describe a simple yet effective approach to merge lexicon information with an attention LSTM model for ABSA in order to leverage both the power of deep neural networks and existing linguistic resources, so that the framework becomes more flexible and robust without requiring additional labeled data. We also explore the effect of regularizing attention vectors by introducing an attention regularizer to allow the network to have a broader "focus" on different parts of the sentence.

2 Related works

ABSA is a fine-grained task which requires the model to be able to produce accurate prediction given different aspects. As it is common that one sentence may contain opposite polarities associated to different aspects at the same time, attention-based LSTM (Wang et al., 2016) was first proposed to allow the network to be able to as-

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 253–259
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

sign higher weights to more relevant words given different aspects. Following this idea, a number of researches have been carried out to keep improving the attention network for ABSA (Ma et al., 2017; Tay et al., 2017; Cheng et al., 2017; He et al., 2018; Zhu and Qian, 2018).

On the other hand, a lot of works have been done focusing on leveraging existing linguistic resources such as sentiment to enhance the performance; however, most works are performed at document and sentence level. For instance, at document level, Teng et al. (2016) proposed a weighted-sum model which consists of representing the final prediction as a weighted sum of the network prediction and the polarities provided by the lexicon. Zou et al. (2018) described a framework to assign higher weights to opinion words found in the lexicon by transforming lexicon polarity to sentiment degree.

At sentence level, Shin et al. (2017) used two convolutional neural networks to separately process sentence and lexicon inputs. Lei et al. (2018) described a multi-head attention network where the attention weights are jointly learned with lexicon inputs. Wu et al. (2018) proposed a new labeling strategy which breaks a sentence into clauses by punctuation to produce more lower-level examples, inputs are then processed at different levels with linguistic information such as lexicon and POS, and finally merged back to perform sentence level prediction. Meanwhile, some other similar works that incorporate linguistic resources for sentiment analysis have been carried out (Rouvier and Favre, 2016; Qian et al., 2017).

Regarding the attention regularization, instead of using *softmax* and *sparesmax*, Niculae and Blondel (2017) proposed *fusemax* as a regularized attention framework to learn the attention weights; Zhang et al. (2019) introduced L_{max} and $Entropy$ as regularization terms to be jointly optimized with the loss. However, both approaches share the same idea of shaping the attention weights to be sharper and more sparse so that the advantage of the attention mechanism is maximized.

In our work, different from the previously mentioned approaches, we incorporate polarities obtained from lexicons directly into the attention-based LSTM network to perform aspect-level sentiment analysis, so that the model improves in terms of robustness without requiring extra training examples. Additionally, we find that the attention vector is likely to over-fit which forces the network to "focus" on particular words while ignoring positions that provide key information for judging the polarity; and that by adding lexical features, it is possible to reduce this effect. Following this idea, we also experimented reducing the over-fitting effect by introducing an attention regularizer. Unlike previously mentioned ideas, we want the attention weights to be less sparse. Details of our approach are in following sections.

3 Methodology

3.1 Baseline AT-LSTM

In our experiments, we replicate AT-LSTM proposed by Wang et al. (2016) as our baseline system. Comparing with a traditional LSTM network (Hochreiter and Schmidhuber, 1997), AT-LSTM is able to learn the attention vector and at the same time to take into account the aspect embeddings. Thus the network is able to assign higher weights to more relevant parts of a given sentence with respect to a specific aspect.

Formally, given a sentence S, let $\{w_1, w_2, ..., w_N\}$ be the word vectors of each word where N is the length of the sentence; $v_a \in R^{d_a}$ represents the aspect embeddings where d_a is its dimension; let $H \in R^{d \times N}$ be a matrix of the hidden states $\{h_1, h_2, ..., h_N \in R^d\}$ produced by LSTM where d is the number of neurons of the LSTM cell. Thus the attention vector α is computed as follows:

$$M = tanh(\begin{bmatrix} W_h H \\ W_v v_a \otimes e_N \end{bmatrix})$$

$$\alpha = softmax(w^T M)$$

$$r = H\alpha^T$$

where, $M \in R^{(d+d_a)\times N}, \alpha \in R^N, r \in R^d, W_h \in R^{d\times d}, W_v \in R^{d_a \times d_a}, w \in R^{d+d_a}$. α is a vector consisting of attention weights and r is a weighted representation of the input sentence with respect to the input aspect. $v_a \otimes e_N = [v_a, v_a, ..., v_a]$, that is, the operator repeatedly concatenates v_a for N times. Then, the final representation is obtained and fed to the output layer as below:

$$h^* = tanh(W_p r + W_x h_N)$$

$$\hat{y} = softmax(W_s h^* + b_s)$$

where, $h^* \in R^d$, W_p and W_x are projection parameters to be learned during training; W_s and b_s

are weights and biases in the output layer. The prediction $\hat{y}$ is then plugged into the cross-entropy loss function for training, and L_2 regularization is applied.

$$loss = -\sum_{i} y_i log(\hat{y}_i) + \lambda \|\Theta\|_2^2 \qquad (1)$$

where i is the number of classes (three way classification in our experiments); λ is the hyperparameter for L_2 regularization; Θ is the regularized parameter set in the network.

3.2 ATLX

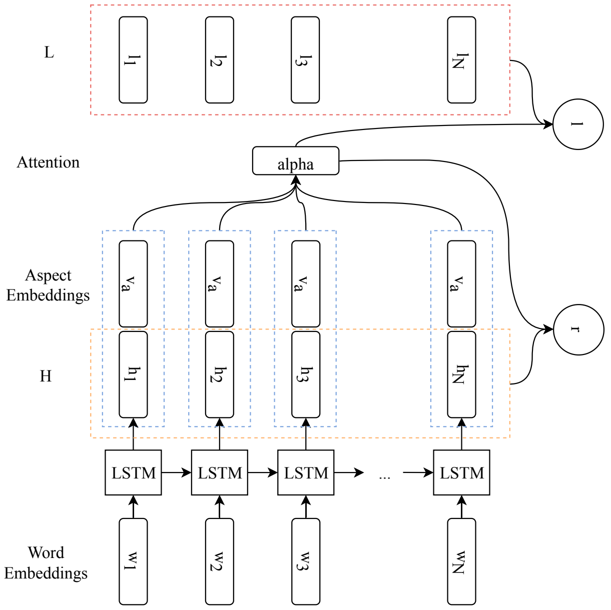

Figure 1: ATLX model diagram

3.2.1 Lexicon Build

Similar to Shin et al. (2017), but in a different way, we build our lexicon by merging 4 existing lexicons to one: MPQA, Opinion Lexicon, Opener and Vader. SentiWordNet was in the initial design but was removed from the experiments as unnecessary noise was introduced, e.g. *highly* is annotated as *negative*. For categorical labels such as *negative, weakneg, neutral, both, positive*, we convert them to values in $\{-1.0, -0.5, 0.0, 0.0, 1.0\}$ respectively. Regarding lexicons with real value annotations, for each lexicon, we adopt the annotated value standardized by the maximum polarity in that lexicon. Finally, the union U of all lexicons is taken where each word $w_l \in U$ has an associated vector $v_l \in R^n$ that represents the polarity given by each lexicon. n here is the number of lexicons; average values across all available lexicons are taken for missing values. e.g. the lexical feature for word *adorable* is represented by $[1.0, 1.0, 1.0, 0.55]$, which are taken from MPQA(1.0), Opener(1.0), Opinion Lexicon(1.0) and Vader(0.55) respectively. For words outside U, a zero vector of dimension n is supplied.

3.2.2 Lexicon Integration

To merge the lexical features obtained from U into the baseline, we first perform a linear transformation to the lexical features in order to preserve the original sentiment distribution and have compatible dimensions for further computations. Later, the attention vector learned as in the baseline is applied to the transformed lexical features. In the end, all information is added together to perform the final prediction.

Formally, let $V_l \in R^{n \times N}$ be the lexical matrix for the sentence, V_l then is transformed linearly:

$$L = W_l \cdot V_l$$

where $L \in R^{d \times N}, W_l \in R^{d \times n}$. Later, the attention vector learned from the concatenation of H and $v_a \otimes e_N$ is applied to L:

$$l = L \cdot \alpha^T$$

where $l \in R^d, \alpha \in R^N$. Finally h^* is updated and passed to output layer for prediction:

$$h^* = tanh(W_p r + W_x h_N + W_l l)$$

where $W_l \in R^{d \times d}$ is a projection parameter as W_p and W_x. The model architecture is shown in Figure 1.

3.3 Attention Regularization

As observed in both Figure 2 and Figure 3, the attention weights in ATLX seem less sparse across the sentence, while the ones in the baseline are focusing only on the final part of the sentence. It is reasonable to think that the attention vector might be over-fitting in some cases, causing the network to ignore other relevant positions, since the attention vector is learned purely on training examples. Thus we propose a simple attention regularizer to further validate our hypothesis, which consists of adding into the loss function a parameterized standard deviation or negative entropy term for the attention weights. The idea is to avoid the attention vector to have heavy weights in few positions, instead, it is preferred to have higher weights for more positions. Formally, the attention regularized loss is computed as:

$$loss = -\sum_i y_i log(\hat{y}_i) + \lambda\|\Theta\|_2^2 + \epsilon \cdot R(\alpha) \quad (2)$$

compared to equation (1), a second regularization term is added, where ϵ is the hyper-parameter for the attention regularizer; R stands for the regularization term defined in (3) or (4); and α is the distribution of attention weights. Note that during implementation, the attention weights for batch padding positions are excluded from α.

We experiment two different regularizers, one uses standard deviation of α defined in equation (3); the other one uses the negative entropy of α defined in equation (4).

$$R(\alpha) = \sigma(\alpha) \quad (3)$$

$$R(\alpha) = -[-\sum_i^N \alpha_i \cdot log(\alpha_i)] \quad (4)$$

4 Experiments

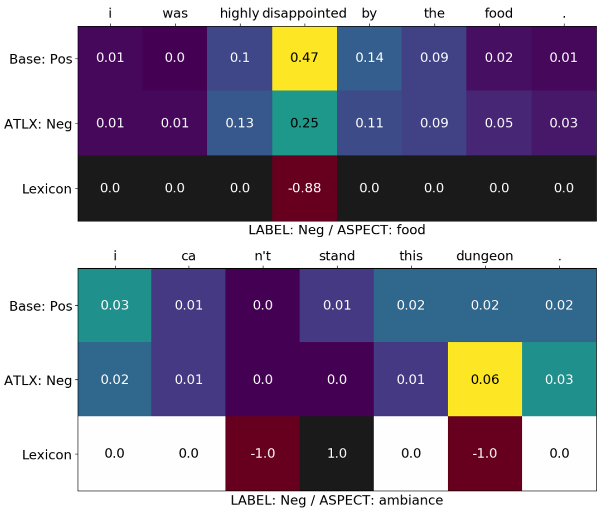

Figure 2: Comparison of attention weights between baseline and ATLX; The rows annotated as "Lexicon" indicates the average polarity per word given by U.

4.1 Dataset

Same as Wang et al. (2016), we experiment on SemEval 2014 Task 4, restaurant domain dataset. The data consists of reviews of restaurants with aspects: {*food, price, service, ambience, miscellaneous*} and associated polarities: {*positive, neutral, negative*}. The objective is to predict the polarity given a sentence and an aspect. There are 3,518 training examples and 973 test examples in the corpus. To initialize word vectors with pretrained word embeddings, the 300 dimensional Glove vectors trained on 840b tokens are used, as described in the original paper.

	Pos	Neu	Neg	In Corpus
MPQA	2298	440	4148	908
OL	2004	3	4780	732
Opener	2298	440	4147	908
Vader	3333	0	4170	656
Merged U	5129	404	7764	1234

Table 1: Lexicon statistics of positive, neutral, negative words and number of words covered in corpus.

4.2 Lexicons

As shown in Table 1, we merge four existing and online available lexicons into one. The merged lexicon U as described in section 3.2.1 is used for our experiments. After the union, the following postprocess is carried out: $\{bar, try, too\}$ are removed from U since they are unreasonably annotated as negative by MPQA and Opener; $\{n't, not\}$ are added to U with -1 polarity for negation.

4.3 Evaluation

Cross validation is applied to measure the performance of each model. In all experiments, the training set is randomly shuffled and split into 6 folds with a fixed random seed. According to the code released by Wang et al. (2016), a development set containing 528 examples is used, which is roughly $\frac{1}{6}$ of the training corpus. In order to remain faithful to the original implementation, we thus evaluate our model with a cross validation of 6 folds.

As shown in Table 2, compared to the baseline system, ATLX is not only able to improve in terms of accuracy, but also the variance of the performance across different sets gets significantly reduced. On the other hand, by adding attention regularization to the baseline system without introducing lexical features, both the standard deviation regularizer (base$^{\text{std}}$) and the negative entropy regularizer (base$^{\text{ent}}$) are able to contribute positively; where base$^{\text{ent}}$ yields largest improvement. By combining attention regularization and lexical features together, although the model is able to further improve, the difference is too small to draw strong conclusion.

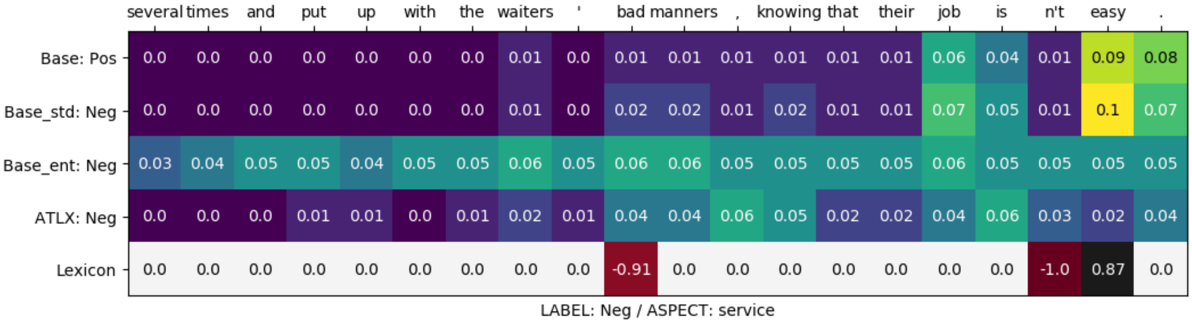

Figure 3: Comparison of attention weights between baseline, base$^{\text{std}}$, base$^{\text{ent}}$ and ATLX.

	CV	σ^{CV}	TEST	σ^{TEST}
base	75.27	1.420	81.48	1.157
base$^{\text{std}}$	74.67	1.688	81.57	0.915
base$^{\text{ent}}$	**75.93**	1.467	82.24	0.863
ATLX	75.64	1.275	82.62	**0.498**
ATLX$^{\text{std}}$	75.64	1.275	82.68	0.559
ATLX$^{\text{ent}}$	75.53	**1.265**	**82.86**	1.115
ATLX*	74.99	1.638	82.03	1.409
base$^{\text{LX}}$	71.98	1.588	79.24	2.322

Table 2: Mean accuracy and standard deviation of cross validation results on 6 folds of development sets and one test set. Note that in our replicated baseline system, test accuracy ranges from 80.06 to 83.45; 83.1 was reported in the original paper.

5 Discussion

5.1 ATLX

As described in previously, the overall performance of the baseline gets enhanced by leveraging lexical features independent from the training data, which makes the model more robust and flexible. The example in Figure 2, although the baseline is able to pay relatively high attention to the word *disappointed* and *dungeon*, it is not able to recognize these words as clear indicators of *negative* polarity; while ATLX is able to correctly predict *positive* for both examples. On the other hand, it is worth mentioning that the computation of the attention vector α does not take lexical features V_l into account. Although it is natural to think that adding V_l as input for computing α would be a good option, the results of ATLX* in Table 2 suggest otherwise.

In order to understand where does the improvement of ATLX come from, lexical features or the way we introduce lexical features to the system? We conduct a support experiment to verify its impact (base$^{\text{LX}}$), which consists of naively concatenating input word vector with its associated lexical vector and feed the extended embedding to the baseline. As demonstrated in Table 2, by comparing baseline with base$^{\text{LX}}$, we see that the simple merge of lexical features with the network without carefully designed mechanism, the model is not able to leverage new information; and in contrast, the overall performance gets decreased.

5.2 Attention Regularization

As shown in Figure 3, when comparing ATLX with the baseline, we find that although the lexicon only provides non-neutral polarity information for three words, the attention weights of ATLX are less sparse and less spread out than in the baseline. Also, this effect is general as the standard deviation of the attention weights distribution for the test set in ATLX (0.0219) are significantly lower than in the baseline (0.0354).

Thus it makes us think that the attention weights might be over-fitting in some cases as it is purely learned on training examples. This could cause that by giving too much weight to particular words in a sentence, the network ignores other positions which could provide key information for classifying the polarity. For instance, the example in Figure 3 shows that the baseline which predicts *positive* is "focusing" on the final part of the sentence, mostly the word *easy*; while ignoring the *bad manners* coming before, which is key for judging the polarity of the sentence given the aspect *service*. In contrast, the same baseline model trained with attention regularized by standard deviation is able to correctly predict *negative* just by "focusing" a little bit more on the *"bad manners"* part.

However, the hard regularization by standard deviation might not be ideal as the optimal minimum value of the regularizer will imply that all words in the sentence have homogeneous weight,

Parameter name	Value
ϵ base$^{\text{std}}$	1e-3
ϵ base$^{\text{ent}}$	0.5
ϵ ATLX$^{\text{std}}$	1e-4
ϵ ATLX$^{\text{ent}}$	0.006

Table 3: Attention regularization parameter settings

which is the opposite of what the attention mechanism is able to gain.

Regarding the negative entropy regularizer, taking into account that the attention weights are output of $softmax$ which is normalized to sum up to 1, although the minimum value of this term would also imply homogeneous weight of $\frac{1}{N}$, it is interesting to see that with almost evenly distributed α, the model remains sensitive to few positions with relatively higher weights; e.g. in Figure 3, the same sentence with entropy regularization demonstrates that although most positions are closely weighted, the model is still able to differentiate key positions even with a weight difference of 0.01 and predict correctly.

6 Parameter Settings

In our experiments, apart from newly introduced parameter ϵ for attention regularization, we follow Wang et al. (2016) and their released code.

More specifically, we set batch size as 25; aspect embedding dimension d_a equals to 300, same as Glove vector dimension; number of LSTM cell d as 300; number of LSTM layers as 1; dropout with 0.5 keep probability is applied to h^*; AdaGrad optimizer is used with initial accumulate value equals to 1e-10; learning rate is set to 0.01; L2 regularization parameter λ is set to 0.001; network parameters are initialized from a random uniform distribution with min and max values as -0.01 and 0.01; all network parameters except word embeddings are included in the L2 regularizer. The hyperparmerter ϵ for attention regularization is shown in Table 3.

7 Conclusion and Future Works

In this paper, we describe our approach of directly leveraging numerical polarity features provided by existing lexicon resources in an aspect-based sentiment analysis environment with an attention LSTM neural network. Meanwhile, we stress that the attention mechanism may over-fit on particular positions, blinding the model from other relevant positions. We also explore two regularizers to reduce this overfitting effect. The experimental results demonstrate the effectiveness of our approach.

For future works, since the lexical features can be leveraged directly by the network to boost performance, a fine-grained lexicon which is domain and aspect specific in principle could further improve similar models. On the other hand, although the negative entropy regularizer is able to reduce the overfitting effect, a better attention framework could be researched, so that the attention distribution would be sharp and spare but at the same time, being able to "focus" on more positions.

References

Jiajun Cheng, Hui Wang, Shenglin Zhao, Xin Zhang, Irwin King, and Jiani Zhang. 2017. Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*, pages 97–106.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1121–1131.

Sepp; Hochreiter and J?urgen Schmidhuber. 1997. Long Short Term Memory. *Neural Computation*, 9(8):1735–1780.

Zeyang Lei, Yujiu Yang, and Min Yang. 2018. Sentiment Lexicon Enhanced Attention-Based LSTM for Sentiment Classification. *AAAI-2018-short paper*, pages 8105–8106.

Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *IJCAI International Joint Conference on Artificial Intelligence*, pages 4068–4074.

Vlad Niculae and Mathieu Blondel. 2017. A Regularized Framework for Sparse and Structured Neural Attention. *NIPS*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. *Empirical Methods in Natural Language Processing (EMNLP)*, 10(July):79–86.

Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. 2017. Linguistically Regularized LSTM for Sentiment Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1679–1689.

Mickael Rouvier and Benoit Favre. 2016. SENSEI-LIF at SemEval-2016 Task 4 : Polarity embedding fusion for robust sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 207–213.

Bonggun Shin, Timothy Lee, and Jinho D Choi. 2017. Lexicon Integrated CNN Models with Attention for Sentiment Analysis. *ACL*, pages 149–158.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. Dyadic Memory Networks for Aspect-based Sentiment Analysis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*, pages 107–116.

Zhiyang Teng, Duy-Tin Vo, and Yue Zhang. 2016. Context-Sensitive Lexicon Features for Neural Sentiment Analysis. *EMNLP*, pages 1629–1638.

Peter D Turney. 2002. Thumbs up or thumbs down? Semantic Orientation applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, (July):417–424.

Yequan Wang, Minlie Huang, Li Zhao, and Xiaoyan Zhu. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 606–615.

Theresa Wilson, Theresa Wilson, Janyce Wiebe, Janyce Wiebe, Rebecca Hwa, and Rebecca Hwa. 2004. Just how mad are you? Finding strong and weak opinion clauses. *Proceedings of the National Conference on Artificial Intelligence*, pages 761–769.

Ou Wu, Tao Yang, Mengyang Li, and Ming Li. 2018. $$-hot Lexicon Embedding-based Two-level LSTM for Sentiment Analysis.

Jiajun Zhang, Yang Zhao, Haoran Li, and Chengqing Zong. 2019. Attention with sparsity regularization for neural machine translation and summarization. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 27(3):507–518.

Peisong Zhu and Tieyun Qian. 2018. Enhanced Aspect Level Sentiment Classification with Auxiliary Memory. In *COLING*, pages 1077–1087.

Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2018. A Lexicon-Based Supervised Attention Model for Neural Sentiment Analysis. In *COLING*, pages 868–877.

A Supplemental Material

A.1 Resource Details

Lexical resources: MPQA[1], Opinion Lexicon[2], Opener[3], and Vader[4]. Glove vectors[5]. Code[6] released by Wang et al. (2016). Experiments described in this paper are implemented with TensorFlow[7].

[1] http://mpqa.cs.pitt.edu/#subj_lexicon

[2] https://www.cs.uic.edu/l̃iub/FBS/sentiment-analysis.html#lexicon

[3] https://github.com/opener-project/VU-sentiment-lexicon/tree/master/VUSentimentLexicon/EN-lexicon

[4] https://github.com/cjhutto/vaderSentiment

[5] https://nlp.stanford.edu/projects/glove/

[6] https://www.wangyequan.com/publications/

[7] https://www.tensorflow.org/