Double Embeddings and CNN-based Sequence Labeling for Aspect Extraction

Hu Xu[1], **Bing Liu**[1], **Lei Shu**[1] and **Philip S. Yu**[1,2]
[1]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[2]Institute for Data Science, Tsinghua University, Beijing, China
{hxu48, liub, lshu3, psyu}@uic.edu

Abstract

One key task of fine-grained sentiment analysis of product reviews is to extract product aspects or features that users have expressed opinions on. This paper focuses on supervised aspect extraction using deep learning. Unlike other highly sophisticated supervised deep learning models, this paper proposes a novel and yet simple CNN model [1] employing two types of pre-trained embeddings for aspect extraction: general-purpose embeddings and domain-specific embeddings. Without using any additional supervision, this model achieves surprisingly good results, outperforming state-of-the-art sophisticated existing methods. To our knowledge, this paper is the first to report such double embeddings based CNN model for aspect extraction and achieve very good results.

1 Introduction

Aspect extraction is an important task in sentiment analysis (Hu and Liu, 2004) and has many applications (Liu, 2012). It aims to extract opinion targets (or aspects) from opinion text. In product reviews, aspects are product attributes or features. For example, from "*Its speed is incredible*" in a laptop review, it aims to extract "speed".

Aspect extraction has been performed using supervised (Jakob and Gurevych, 2010; Chernyshevich, 2014; Shu et al., 2017) and unsupervised approaches (Hu and Liu, 2004; Zhuang et al., 2006; Mei et al., 2007; Qiu et al., 2011; Yin et al., 2016; He et al., 2017). Recently, supervised deep learning models achieved state-of-the-art performances (Li and Lam, 2017). Many of these models use handcrafted features, lexicons, and complicated neural network architectures (Poria et al., 2016; Wang et al., 2016, 2017; Li and Lam, 2017). Although these approaches can achieve better performances than their prior works, there are two other considerations that are also important. (1) Automated feature (representation) learning is always preferred. How to achieve competitive performances without manually crafting features is an important question. (2) According to Occam's razor principle (Blumer et al., 1987), a simple model is always preferred over a complex model. This is especially important when the model is deployed in a real-life application (e.g., chatbot), where a complex model will slow down the speed of inference. Thus, to achieve competitive performance whereas keeping the model as simple as possible is important. This paper proposes such a model.

To address the first consideration, we propose a double embeddings mechanism that is shown crucial for aspect extraction. The embedding layer is the very first layer, where all the information about each word is encoded. The quality of the embeddings determines how easily later layers (e.g., LSTM, CNN or attention) can decode useful information. Existing deep learning models for aspect extraction use either a pre-trained general-purpose embedding, e.g., GloVe (Pennington et al., 2014), or a general review embedding (Poria et al., 2016). However, aspect extraction is a complex task that also requires fine-grained domain embeddings. For example, in the previous example, detecting "speed" may require embeddings of both "Its" and "speed". However, the criteria for good embeddings for "Its" and "speed" can be totally different. "Its" is a general word and the general embedding (trained from a large corpus) is likely to have a better representation for "Its". But, "speed" has a very fine-grained meaning (e.g., how many instructions per second) in the *laptop* domain,

[1]The code of this paper can be found at https://www.cs.uic.edu/~hxu/.

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 592–598
Melbourne, Australia, July 15 - 20, 2018.

whereas "speed" in general embeddings or general review embeddings may mean how many miles per second. So using in-domain embeddings is important even when the in-domain embedding corpus is not large. Thus, we leverage both general embeddings and domain embeddings and let the rest of the network to decide which embeddings have more useful information.

To address the second consideration, we use a pure Convolutional Neural Network (CNN) (LeCun et al., 1995) model for sequence labeling. Although most existing models use LSTM (Hochreiter and Schmidhuber, 1997) as the core building block to model sequences (Liu et al., 2015; Li and Lam, 2017), we noticed that CNN is also successful in many NLP tasks (Kim, 2014; Zhang et al., 2015; Gehring et al., 2017). One major drawback of LSTM is that LSTM cells are sequentially dependent. The forward pass and backpropagation must serially go through the whole sequence, which slows down the training/testing process [2]. One challenge of applying CNN on sequence labeling is that convolution and max-pooling operations are usually used for summarizing sequential inputs and the outputs are not well-aligned with the inputs. We discuss the solutions in Section 3.

We call the proposed model Dual Embeddings CNN (DE-CNN). To the best of our knowledge, this is the first paper that reports a double embedding mechanism and a pure CNN-based sequence labeling model for aspect extraction.

2 Related Work

Sentiment analysis has been studied at document, sentence and aspect levels (Liu, 2012; Pang and Lee, 2008; Cambria and Hussain, 2012). This work focuses on the aspect level (Hu and Liu, 2004). Aspect extraction is one of its key tasks, and has been performed using both unsupervised and supervised approaches. The unsupervised approach includes methods such as frequent pattern mining (Hu and Liu, 2004; Popescu and Etzioni, 2005), syntactic rules-based extraction (Zhuang et al., 2006; Wang and Wang, 2008; Qiu et al., 2011), topic modeling (Mei et al., 2007; Titov and McDonald, 2008; Lin and He, 2009; Moghaddam and Ester, 2011), word alignment (Liu et al., 2013) and label propagation (Zhou et al., 2013; Shu et al., 2016).

Traditionally, the supervised approach (Jakob and Gurevych, 2010; Mitchell et al., 2013; Shu et al., 2017) uses Conditional Random Fields (CRF) (Lafferty et al., 2001). Recently, deep neural networks are applied to learn better features for supervised aspect extraction, e.g., using LSTM (Williams and Zipser, 1989; Hochreiter and Schmidhuber, 1997; Liu et al., 2015) and attention mechanism (Wang et al., 2017; He et al., 2017) together with manual features (Poria et al., 2016; Wang et al., 2016). Further, (Wang et al., 2016, 2017; Li and Lam, 2017) also proposed aspect and opinion terms co-extraction via a deep network. They took advantage of the gold-standard opinion terms or sentiment lexicon for aspect extraction. The proposed approach is close to (Liu et al., 2015), where only the annotated data for aspect extraction is used. However, we will show that our approach is more effective even compared with baselines using additional supervisions and/or resources.

The proposed embedding mechanism is related to cross domain embeddings (Bollegala et al., 2015, 2017) and domain-specific embeddings (Xu et al., 2018a,b). However, we require the domain of the domain embeddings must exactly match the domain of the aspect extraction task. CNN (LeCun et al., 1995; Kim, 2014) is recently adopted for named entity recognition (Strubell et al., 2017). CNN classifiers are also used in sentiment analysis (Poria et al., 2016; Chen et al., 2017). We adopt CNN for sequence labeling for aspect extraction because CNN is simple and parallelized.

3 Model

The proposed model is depicted in Figure 1. It has 2 embedding layers, 4 CNN layers, a fully-connected layer shared across all positions of words, and a softmax layer over the labeling space $\mathcal{Y} = \{B, I, O\}$ for each position of inputs. Note that an aspect can be a phrase and B, I indicate the beginning word and non-beginning word of an aspect phrase and O indicates non-aspect words.

Assume the input is a sequence of word indexes $\mathbf{x} = (x_1, \ldots, x_n)$. This sequence gets its two corresponding continuous representations $\mathbf{x}^g$ and $\mathbf{x}^d$ via two separate embedding layers (or embedding matrices) W^g and W^d. The first embedding matrix W^g represents general embeddings pre-

[2] We notice that a GPU with more cores has no training time gain on a low-dimensional LSTM because extra cores are idle and waiting for the other cores to sequentially compute cells.

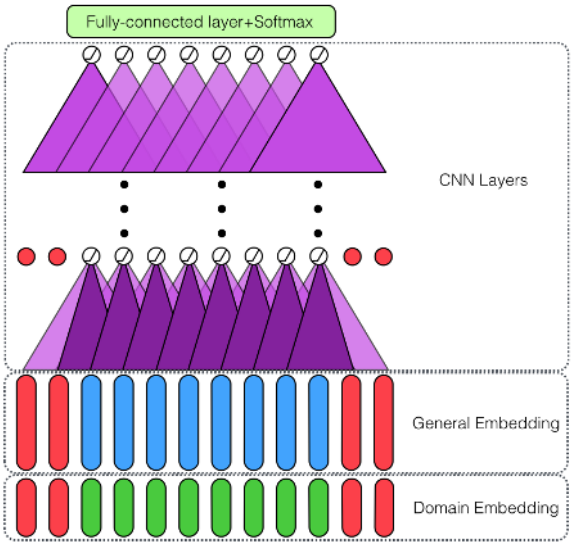

Figure 1: Overview of DE-CNN: red vectors are zero vectors; purple triangles are CNN filters.

trained from a very large general-purpose corpus (usually hundreds of billions of tokens). The second embedding matrix W^d represents domain embeddings pre-trained from a small in-domain corpus, where the scope of the domain is exactly the domain that the training/testing data belongs to. As a counter-example, if the training/testing data is in the *laptop* domain, then embeddings from the *electronics* domain are considered to be out-of-domain embeddings (e.g., the word "adapter" may represent different types of adapters in *electronics* rather than exactly a *laptop* adapter). That is, only laptop reviews are considered to be in-domain.

We do not allow these two embedding layers trainable because small training examples may lead to many unseen words in test data. If embeddings are tunable, the features for seen words' embeddings will be adjusted (e.g., forgetting useless features and infusing new features that are related to the labels of the training examples). And the CNN filters will adjust to the new features accordingly. But the embeddings of unseen words from test data still have the old features that may be mistakenly extracted by CNN.

Then we concatenate two embeddings $\mathbf{x}^{(1)} = \mathbf{x}^g \oplus \mathbf{x}^d$ and feed the result into a stack of 4 CNN layers. A CNN layer has many 1D-convolution filters and each (the r-th) filter has a fixed kernel size $k = 2c+1$ and performs the following convolution operation and ReLU activation:

$$x_{i,r}^{(l+1)} = \max\left(0, (\sum_{j=-c}^{c} w_{j,r}^{(l)} x_{i+j}^{(l)}) + b_r^{(l)}\right), \quad (1)$$

where l indicates the l-th CNN layer. We apply each filter to all positions $i = 1 : n$. So each filter computes the representation for the i-th word along with $2c$ nearby words in its context. Note that we force the kernel size k to be an odd number and set the stride step to be 1 and further pad the left c and right c positions with all zeros. In this way, the output of each layer is well-aligned with the original input $\mathbf{x}$ for sequence labeling purposes. For the first ($l = 1$) CNN layer, we employ two different filter sizes. For the rest 3 CNN ($l \in \{2, 3, 4\}$) layers, we only use one filter size. We will discuss the details of the hyper-parameters in the experiment section. Finally, we apply a fully-connected layer with weights shared across all positions and a softmax layer to compute label distribution for each word. The output size of the fully-connected layer is $|\mathcal{Y}| = 3$. We apply dropout after the embedding layer and each ReLU activation. Note that we do not apply any max-pooling layer after convolution layers because a sequence labeling model needs good representations for every position and max-pooling operation mixes the representations of different positions, which is undesirable (we show a max-pooling baseline in the next section).

Description	Training #S./#A.	Testing #S./#A.
SemEval-14 Laptop	3045/2358	800/654
SemEval-16 Restaurant	2000/1743	676/622

Table 1: Dataset description with the number of sentences(#S.) and number of aspect terms(#A.)

4 Experiments

4.1 Datasets

Following the experiments of a recent aspect extraction paper (Li and Lam, 2017), we conduct experiments on two benchmark datasets from SemEval challenges (Pontiki et al., 2014, 2016) as shown in Table 4.1. The first dataset is from the *laptop* domain on subtask 1 of SemEval-2014 Task 4. The second dataset is from the *restaurant* domain on subtask 1 (slot 2) of SemEval-2016 Task 5. These two datasets consist of review sentences with aspect terms labeled as spans of characters.

We use NLTK[3] to tokenize each sentence into a sequence of words.

For the general-purpose embeddings, we use the glove.840B.300d embeddings (Pennington et al., 2014), which are pre-trained from a corpus of 840 billion tokens that cover almost all web pages. These embeddings have 300 dimensions. For domain-specific embeddings, we collect a laptop review corpus and a restaurant review corpus and use fastText (Bojanowski et al., 2016) to train domain embeddings. The laptop review corpus contains all laptop reviews from the Amazon Review Dataset (He and McAuley, 2016). The restaurant review corpus is from the Yelp Review Dataset Challenge [4]. We only use reviews from restaurant categories that the second dataset is selected from [5]. We set the embedding dimensions to 100 and the number of iterations to 30 (for a small embedding corpus, embeddings tend to be under-fitted), and keep the rest hyper-parameters as the defaults in fastText. We further use fastText to compose out-of-vocabulary word embeddings via subword N-gram embeddings.

4.2 Baseline Methods

We perform a comparison of DE-CNN with three groups of baselines using the standard evaluation of the datasets[6] [7]. The results of the first two groups are copied from (Li and Lam, 2017). The first group uses single-task approaches.

CRF is conditional random fields with basic features[8] and GloVe word embedding(Pennington et al., 2014).

IHS_RD (Chernyshevich, 2014) and **NLANGP** (Toh and Su, 2016) are best systems in the original challenges (Pontiki et al., 2014, 2016).

WDEmb (Yin et al., 2016) enhanced CRF with word embeddings, linear context embeddings and dependency path embeddings as input.

LSTM (Liu et al., 2015; Li and Lam, 2017) is a vanilla BiLSTM.

BiLSTM-CNN-CRF (Reimers and Gurevych, 2017) is the state-of-the-art from the Named Entity Recogntion (NER) community. We use this baseline[9] to demonstrate that a NER model may need further adaptation for aspect extraction.

The second group uses multi-task learning and also take advantage of gold-standard opinion terms/sentiment lexicon.

RNCRF (Wang et al., 2016) is a joint model with a dependency tree based recursive neural network and CRF for aspect and opinion terms co-extraction. Besides opinion annotations, it also uses handcrafted features.

CMLA (Wang et al., 2017) is a multi-layer coupled-attention network that also performs aspect and opinion terms co-extraction. It uses gold-standard opinion labels in the training data.

MIN (Li and Lam, 2017) is a multi-task learning framework that has (1) two LSTMs for jointly extraction of aspects and opinions, and (2) a third LSTM for discriminating sentimental and non-sentimental sentences. A sentiment lexicon and high precision dependency rules are employed to find opinion terms.

The third group is the variations of DE-CNN.

GloVe-CNN only uses glove.840B.300d to show that domain embeddings are important.

Domain-CNN does not use the general embeddings to show that domain embeddings alone are not good enough as the domain corpus is limited for training good general words embeddings.

MaxPool-DE-CNN adds max-pooling in the last CNN layer. We use this baseline to show that the max-pooling operation used in the traditional CNN architecture is harmful to sequence labeling.

DE-OOD-CNN replaces the domain embeddings with out-of-domain embeddings to show that a large out-of-domain corpus is not a good replacement for a small in-domain corpus for domain embeddings. We use all *electronics* reviews as the out-of-domain corpus for the *laptop* and all the Yelp reviews for *restaurant*.

DE-Google-CNN replaces the glove embeddings with GoogleNews embeddings[10], which are pre-trained from a smaller corpus (100 billion tokens). We use this baseline to demonstrate that general embeddings that are pre-trained from a larger corpus performs better.

DE-CNN-CRF replaces the softmax activation with a CRF layer[11]. We use this baseline to

[3] http://www.nltk.org/

[4] https://www.yelp.com/dataset/challenge

[5] http://www.cs.cmu.edu/~mehrbod/RR/Cuisines.wht

[6] http://alt.qcri.org/semeval2014/task4

[7] http://alt.qcri.org/semeval2016/task5

[8] http://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html

[9] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

[10] https://code.google.com/archive/p/word2vec/

[11] https://github.com/allenai/allennlp

Model	Laptop	Restaurant
CRF	74.01	69.56
IHS_RD	74.55	-
NLANGP	-	72.34
WDEmb	75.16	-
LSTM	75.25	71.26
BiLSTM-CNN-CRF	77.8	72.5
RNCRF	78.42	-
CMLA	77.80	-
MIN	77.58	73.44
GloVe-CNN	77.67	72.08
Domain-CNN	78.12	71.75
MaxPool-DE-CNN	77.45	71.12
DE-LSTM	78.73	72.94
DE-OOD-CNN	80.21	74.2
DE-Google-CNN	78.8	72.1
DE-CNN-CRF	80.8	74.1
DE-CNN	**81.59***	**74.37***

Table 2: Comparison results in F_1 score: numbers in the third group are averaged scores of 5 runs as in (Li and Lam, 2017). * indicates the result is statistical significant at the level of 0.05.

demonstrate that CRF may not further improve the challenging performance of aspect extraction.

4.3 Hyper-parameters

We hold out 150 training examples as validation data to decide the hyper-parameters. The first CNN layer has 128 filters with kernel sizes $k = 3$ (where $c = 1$ is the number of words on the left (or right) context) and 128 filters with kernel sizes $k = 5$ ($c = 2$). The rest 3 CNN layers have 256 filters with kernel sizes $k = 5$ ($c = 2$) per layer. The dropout rate is 0.55 and the learning rate of Adam optimizer (Kingma and Ba, 2014) is 0.0001 because CNN training tends to be unstable.

4.4 Results and Analysis

Table 4.3 shows that DE-CNN performs the best. The double embedding mechanism improves the performance and in-domain embeddings are important. We can see that using general embeddings (GloVe-CNN) or domain embeddings (Domain-CNN) alone gives inferior performance. We further notice that the performance on *Laptops* and *Restaurant* domains are quite different. *Laptops* has many domain-specific aspects, such as "adapter". So the domain embeddings for *Laptops* are better than the general embeddings. The *Restaurant* domain has many very general aspects like "staff", "service" that do not deviate much from their general meanings. So general embeddings are not bad. Max pooling is a bad operation as indicated by MaxPool-DE-CNN since the max pooling operation loses word positions. DE-OOD-CNN's performance is poor, indicating that making the training corpus of domain embeddings to be exactly in-domain is important. DE-Google-CNN uses a much smaller training corpus for general embeddings, leading to poorer performance than that of DE-CNN. Surprisingly, we notice that the CRF layer (DE-CNN-CRF) does not help. In fact, the CRF layer can improve 1-2% when the laptop's performance is about 75%. But it doesn't contribute much when laptop's performance is above 80%. CRF is good at modeling label dependences (e.g., label I must be after B), but many aspects are just single words and the major types of errors (mentioned later) do not fall in what CRF can solve. Note that we did not tune the hyperparameters of DE-CNN-CRF for practical purpose because training the CRF layer is extremely slow.

One important baseline is BiLSTM-CNN-CRF, which is markedly worse than our method. We believe the reason is that this baseline leverages dependency-based embeddings(Levy and Goldberg, 2014), which could be very important for NER. NER models may require further adaptations (e.g., domain embeddings) for opinion texts.

DE-CNN has two major types of errors. One type comes from inconsistent labeling (e.g., for the restaurant data, the same aspect is sometimes labeled and sometimes not). Another major type of errors comes from unseen aspects in test data that require the semantics of the conjunction word "and" to extract. For example, if A is an aspect and when "A and B" appears, B should also be extracted but not. We leave this to future work.

5 Conclusion

We propose a CNN-based aspect extraction model with a double embeddings mechanism without extra supervision. Experimental results demonstrated that the proposed method outperforms state-of-the-art methods with a large margin.

6 Acknowledgments

This work was supported in part by NSF through grants IIS-1526499, IIS-1763325, and IIS1407927, CNS-1626432, NSFC 61672313, and a gift from Huawei Technologies.

References

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. 1987. Occam's razor. *Information processing letters*, 24(6):377–380.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. 2017. Think globally, embed locally—locally linear meta-embedding of words. *arXiv preprint arXiv:1709.06671*.

Danushka Bollegala, Takanori Maehara, and Ken-ichi Kawarabayashi. 2015. Unsupervised cross-domain word representation learning. *arXiv preprint arXiv:1505.07184*.

Erik Cambria and Amir Hussain. 2012. *Sentic Computing Techniques, Tools, and Applications 2nd Edition*.

Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. 2017. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230.

Maryna Chernyshevich. 2014. Ihs r&d belarus: Cross-domain extraction of product features using crf. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 309–313.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 388–397.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee.

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 *KDD '04*, pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *EMNLP '10*, pages 1035–1045.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01*, pages 282–289.

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892.

Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *CIKM '09*, pages 375–384.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Kang Liu, Liheng Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *IJCAI '13*, pages 2134–2140.

Pengfei Liu, Shafiq Joty, and Helen 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*, pages 1433–1443.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *WWW '07*, pages 171–180.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *ACL '13*, pages 1643–1654.

Samaneh Moghaddam and Martin Ester. 2011. ILDA: interdependent lda model for learning latent aspects and their ratings from online product reviews. In *SIGIR '11*, pages 665–674.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2:1–135.

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)*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT-EMNLP '05*, pages 339–346.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark.

Lei Shu, Bing Liu, Hu Xu, and Annice Kim. 2016. Lifelong-rl: Lifelong relaxation labeling for separating entities and aspects in opinion targets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Lei Shu, Hu Xu, and Bing Liu. 2017. Lifelong learning crf for supervised aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 148–154.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680.

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. In *ACL '08: HLT*, pages 308–316.

Zhiqiang Toh and Jian Su. 2016. Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 282–288.

Bo Wang and Houfeng Wang. 2008. Bootstrapping both product features and opinion words from chinese customer reviews with cross-inducing. In *IJCNLP '08*, pages 289–295.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv preprint arXiv:1603.06679*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018a. Lifelong domain word embedding via meta-learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press.

Hu Xu, Sihong Xie, Lei Shu, and Philip S. Yu. 2018b. Dual attention network for product compatibility and function satisfiability analysis. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. *arXiv preprint arXiv:1605.07843*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2013. Collective opinion target extraction in Chinese microblogs. In *EMNLP '13*, pages 1840–1850.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *CIKM '06*, pages 43–50.