Target Extraction via Feature-Enriched Neural Networks Model

Dehong Ma, Sujian Li, and Houfeng Wang(✉)

MOE Key Lab of Computational Linguistics, Peking University,
Beijing 100871, China
{madehong,lisujian,wanghf}@pku.edu.cn

Abstract. Target extraction is an important task in target-based sentiment analysis, which aims at identifying the boundary of target in given text. Previous works mainly utilize conditional random field (CRF) with a lot of handcraft features to recognize the target. However, it is hard to manually extract effective features to boost the performance of CRF-based methods. In this paper, we employ gated recurrent units (GRU) with label inference, to find valid label path for word sequence. At the same time, we find that character-level features play important roles in target extraction, and represent each word by concatenating word embedding and character-level representations which are learned via character-level GRU. Further, we capture boundary features of each word from its context words by convolution neural networks to assist the identification of the target boundary, since the boundary of a target is highly related to its context words. Experiments on two datasets show that our model outperforms CRF-based approaches and demonstrate the effectiveness of features learned from character-level and context words.

1 Introduction

Target extraction is a fundamental work in the task of target-based sentiment analysis, which tries to find all targets (e.g. entity, product...) in open corpus like tweets, product comments, etc. For example, in the sentence *"I vote to send Dwyane Wade to the NBA All-Star Game."*, the destination of target extracting is to identify all targets: *person: Dwyane Wade* and *organization: NBA*. The popular approach is regarding target extraction task as a sequence labeling problem. The goal of sequence labeling is to assign a label for each element in the sequence, and we can use Hidden Markov Model (HMM), Max Entropy and Conditional Random Field (CRF) to tackle sequence labeling task. Generally in target extraction task, the label set is composed of the three symbols {*B, I, O*}, which stand for the target beginning, the target inside and non-target respectively. In the above example, the labels of the words *Dwyane Wade* and *NBA* are "*B-PERSON, I-PERSON*" and "*B-ORGANIZATION*" respectively while all the other words are labeled *O*.

Although CRF-based approaches [19] could achieve good results on target extraction, they suffer from automatically extracting effective features for boosting system performance. Recently, neural network methods have exhibited their

© Springer Nature Switzerland AG 2018
M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 353–364, 2018.
https://doi.org/10.1007/978-3-319-99495-6_30

ability of feature extraction. [26] study the effect of word embeddings and automatic feature combinations on the task by extending a CRF baseline using neural networks. [23] use recursive neural networks (RNN) to extract features, feed features to CRF and get good performance on target extraction. However, they just use neural networks as a feature extractor and do not make full use of neural networks' ability on sequence labeling. In this paper, we prefer to explore the potentials of neural networks in sequence labeling for the task of target extraction. To make use of neural networks, we take gated recurrent unit (GRU) [3] networks rather than CRF as decoder because GRU is good at modeling long distance dependency which is good for sequence labeling. As we know, there are dependencies between target labels. For example, label I will never follow label O in a sequence of labels. To avoid these illegal transitions between labels, we adopt a transition matrix [5] which measures the probability of jumping from label i to label j to ensure valid paths of labels and discourage all other paths.

In target extraction, we find that character-level features are a key factor for deciding the labels of a sequence. In the example above, the initial characters of the targets are uppercase. In addition, many words have different variants, but with a similar meaning. In such cases, characters can be used to strengthen the word representation. Further, out-of-vocabulary words are hard to be tackled because they have the same representation without distinction. But character-level representation of word could address this problem in some degree. Thus, to incorporate character-level features into our model, we use character-level GRU on word character sequence to obtain character-level representation for each word.

Although GRU can learn long distance dependency of words, the context of a word also plays an important role in target extraction. In the example above, 2-word contexts of *Dwyane* are *'to send'* and *'Wade to'*. From the left context and right context, the label of *Dwyane* tends to be *B*. From the context of *Wade*, we can infer that the label of *Wade* should be *I*. To learn context features of word, we employ convolution neural networks (CNN) on the word context.

We evaluate our model on two open-domain datasets [19], and experimental results show that our method achieves the state-of-the-art performance and validate the effectiveness of character-level features and context features.

2 Model

In this section, we first display the details of our model. After that, we introduce the details of our model. The overall architecture of our model is shown in Fig. 1.

Our model consists of three parts which are character-level layer, word-level layer and label inference layer. The character-level layer mainly learns the character-level representation for each word to assist word level layer and address the OOV words problem via stack bidirectional gated recurrent unit networks (SBi-GRU). The word-level layer has two parts which the first part also utilize SBi-GRU to learn the long distance dependencies between words and the second part learn the local feature for each word in its context. The last part finds a valid path via modeling labels dependencies with transition score matrix.

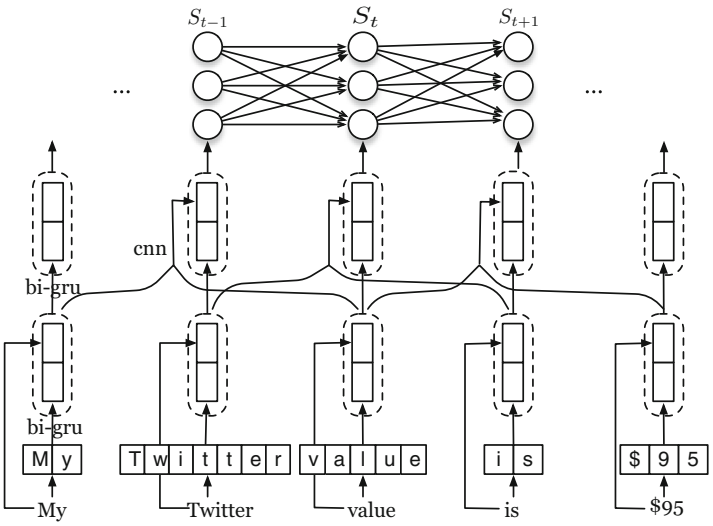

Fig. 1. The overall architecture of our model.

2.1 Embeddings

The first step of neural networks is to transform words and characters into distributed representation, which is also called embeddings [1,16–18].

In our model, word embeddings and characters embedding are used. Formally, we have a word dictionary W of size $|W|$ and a character dictionary C of size $|C|$. W and C are extracted from training data. Word and character will be transformed into corresponding real-value vector if they are in the dictionary. Otherwise, they will be assigned a unique real-value vector. We suppose that a sentence consists of n words and each word i is composed of m characters. The word embeddings of all words are $[W_1, W_2, ..., W_n]$ and the character embeddings of word i is $[C_i^1, C_i^2, ..., C_i^m]$, where $W_i \in R^{|W|*d_w}$ and $C_i^j \in R^{|C|*d_c}$. d_w and d_c are word embedding size and character embedding size. Word embeddings and character embeddings can be learned during training process or pre-trained from large corpus by language model.

2.2 Character-Level Layer

Character-level features are important to target extracting and have positive impact on out-of-vocabulary word problem. To incorporate character-level features, we adopt stack bidirectional gated recurrent units networks (SBi-GRU). The Gated Recurrent Unit (GRU) networks is an extension of recurrent neural networks (RNN) because RNN suffers from the gradient vanishing and

exploration when processing long sequence, and GRU is defined by:

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1}) \tag{1}$$
$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1}) \tag{2}$$
$$\hat{h_t} = tanh(W_h \cdot x_t + U_h \cdot (r_t \odot h_t)) \tag{3}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h_t} \tag{4}$$

where x_t is the input at time step t, h_{t-1} is the hidden state at time step $t-1$, z_t and r_t are update gate and reset gate respectively, σ is sigmoid function, W and U are weight metrics, and $\odot$ denotes element-wise multiplication. For simplification, we use $h_t = GRU(x_t, h_{t-1})$ to denotes the definition of GRU.

For sequence modeling, it is useful to consider both forward and backward information at the same time. Bidirectional GRU (Bi-GRU) is good at learning both direction information, and forward and backward information can be computed by:

$$h_t^f = GRU(x_t, h_{t-1}^f); \tag{5}$$
$$h_t^b = GRU(x_t, h_{t-1}^b). \tag{6}$$

The bidirectional hidden state at time t is the concatenation of forward and backward hidden state, which is defined as:

$$\overline{h}_t = h_t^f \oplus h_t^b. \tag{7}$$

where $\oplus$ is a operation that concatenates two tensors alone the last dimension. For simplification, we use $\overline{h}_t = BiGRU(x_t, \overline{h}_{t-1})$ to stand for bidirectional GRU.

Stack bidirectional GRU (SBi-GRU) can learn high level abstract features for sequence. Therefore, we utilize SBi-GRU to learn effective features for each word. Assuming a stack bidirectional GRU has N layer using the same layer function, and the hidden states $\overline{h}^{(n)}$ are iteratively computed from $n = 1$ to $n = N$ by:

$$\overline{h}_t^{(n)} = BiGRU(\overline{h}_t^{(n-1)}, \overline{h}_{t-1}^{(n)}). \tag{8}$$

where $\overline{h}_t^{(x)}$ is the hidden state of layer x at time t, and $\overline{h}^{(0)}$ is the input sequences.

With the character sequence $[C_i^1, C_i^2, ..., C_i^m]$ of word i as inputs, we can obtain the stack bidirectional hidden states $[\overline{h}_1^{(N)}, \overline{h}_2^{(N)}, ..., \overline{h}_m^{(N)}]$ as the hidden states of character sequence. We can get the final character-level features c_i^r for word i by applying max-pooling or mean-pooling operation on its hidden states.

2.3 Word-Level Layer

In word-level layer, there two parts which serve to learn representation for each word. The first part is a deep architecture consisting of a SBi-GRU networks which are able to build up progressively higher level representations of

sequence data. The input sequences are the concatenation of word embeddings $[W_1, W_2, ..., W_n]$ and character-level word representations $[c_1^r, c_2^r, ..., c_n^r]$. The hidden state of word i from SBi-GRU is $\overline{h}_t^{(n)}$ according to Eq. 8.

In target extracting, it is beneficial to capture local features around word. We employ the convolution neural networks (CNN) to learn local features on context window $[W_{t-\lfloor \frac{k}{2} \rfloor}, ..., W_t, ..., W_{t+\lfloor \frac{k}{2} \rfloor}]$ for word t, and k is the context window size, which is defined as:

$$c_i = f(W_f \cdot X + b_f). \tag{9}$$

where f is non-linear function, $W_f \in R^{uv}$ is filter used to produce new features, X is the concatenation of context window of word t, and b_f is bias.

The $filter$ will produce a feature map $c = [c_1, c_2, ..., c_{k-u+1}]$. We then apply the max-pooling operation on c and get the local features $\hat{c} = max(c)$. The concatenation of the outputs of stack bidirectional GRU and convolution neural networks is word t's features $f(t) = \overline{h}_t^{(n)} \oplus \hat{c}$. The word features are fed into linear transformation layer with activation:

$$f_t = \sigma(W_l \cdot f(t) + b_l). \tag{10}$$

where σ is activation function, W_l and b_l is weight matrix and bias respectively. After that, we use non-linear layer to project f_t into the label space by:

$$y_t = \sigma(W_p \cdot f_t + b_p). \tag{11}$$

where W_p and b_p are weight matrix and bias respectively, and σ is non-linear function.

2.4 Label Inference Layer

In sequence labeling task, there are dependencies between labels, but word level loss discards this kind of label dependency information. To model the dependencies between labels of sequence, [5] introduce a transition score $A_{i,j}$ to measure the probability jumping from label i to label j in a successive words. For an input sequence $x = [x_i, x_2, ..., x_n]$ and its label sequence $y = [y_1, y_2, ..., y_n]$, a sentence level score is the sum of transition score and labeling probability, which is computed by:

$$S(x, y, \theta) = \sum_{t=1}^{n} (A_{i-1,i} + y_t^i). \tag{12}$$

where y_t^i is the score of i_{th} label at t_{th} word, and $\theta = [W_z, W_r, W_h, U_z, U_r, U_h, W_f, b_f, W_l, b_l, W_p, b_p, W_s, b_s]$ are the parameters of our model. We normalize this score over all paths Y via softmax by:

$$p(y|x) = \frac{exp(S(x, y, \theta))}{\sum_{\hat{y} \in Y} exp(S(x, \hat{y}, \theta))}. \tag{13}$$

The word embeddings, character-embeddings and θ will be optimized during training processing, and the training loss is computed by:

$$\text{loss}(x, y) = -\log(p(y|x)). \tag{14}$$

From the formulation above, it is evident that we encourage our model to produce a valid sequence of output labels. While decoding, we predict the output sequence that obtains the maximum score given by:

$$y^* = \arg\max_{\hat{y} \in Y}(S(x, \hat{y}, \theta)). \tag{15}$$

$y*$ can be found by dynamic algorithm like viterbi.

3 Experiments

3.1 Datasets and Evaluate Metric

To verify the effectiveness of our model, we conduct experiments on the data of Mitchell [19][1] which is composed of English and Spanish tweets annotated with entity and sentiment, and we report ten-fold cross-validation results used in [19] and [26].

In order to evaluate the performance of target extracting, we adopt f measure which is defined as:

$$F = 2 * P * R/(P + R);$$

$$P = T/N; R = T/G.$$

where t is the number of correctly predicted targets, n and g are the number of predicted targets and ground truth targets separately.

3.2 Hyperparamters Setting

In our experiments, the character embeddings of two datasets are initialized by Xavier [6]. The word embeddings of English Dataset and Spanish Dataset are from [20][2] and [4][3] respectively. All unknown characters and words, weight matrices and biases are initialized by Xavier. The hidden state size of character-level and word-level stack bidirectional GRU are set to 300 and 600 respectively, and the layer number of character-level and word-level stack bidirectional GRU are all set to 2. We use Adam [11] to optimize all parameters of our model. We also use dropout on word embeddings and character embeddings to avoid overfitting, and the dropout rate is set to 0.5.

[1] http://www.m-mitchell.com/code/index.html.
[2] https://nlp.stanford.edu/projects/glove/.
[3] https://spinningbytes.com/resources/embeddings/.

3.3 Model Comparison

To evaluate the effectiveness of our model comprehensively, we list some baselines for comparison. All baselines are introduced as follows.

- **Discrete** is a CRF-based approach, which incorporates many handcraft features including surface features, linguistic features, cluster features and sentiment features [26].
- **Neural** is an extension of *Discrete* with two changes. The discrete features in *Discrete* are replaced by continuous word embeddings, and a hidden neural layer is added between the inputs and outputs [26].
- **Integrated** makes a combination of discrete models and neural models by integrating both types of inputs into a same CRF framework for the reason that both features can complements each other. [26].
- **GRU** is a baseline completely based on neural networks without any handcraft features and CRF components. *GRU* utilizes gated recurrent units networks to model the long distance dependency between words. *GRU* then uses transition matrix to measure the dependencies between labels and obtain the predicted label for each word.
- **Bi-GRU** extends the *GRU* model by model the input sequence with bidirectional gated recurrent networks for both forward information and backward information which play key role in target extracting, and the other components are the same as *GRU*.

Table 1 shows the performances of our model and other baselines. All baselines can be split into two parts. Baselines in first part are based on CRF and the second part baselines take GRU incorporating label inference as decoder.

From the results of CRF-based approaches, we can see that the performance of Neural model is worst in Spanish dataset and Discrete model obtains worst results in English dataset. Neural model outperforms Discrete model about 4.83% on English dataset, while Discrete model improves the performance of Spanish dataset about 1.73% compared with Neural model. This verifies that both discrete features and word embeddings are useful for target extracting.

Table 1. Performances of baselines and our model.

Model/Dataset	English			Spanish		
	P	R	F	P	R	F
Discrete	0.5937	0.3483	0.4384	0.7077	0.4775	0.5700
Neural	0.5364	0.4487	0.4867	0.6559	0.4782	0.5527
Integrated	0.6069	0.5163	0.5567	0.7023	0.6200	0.6576
GRU	0.5649	0.3849	0.4569	0.6157	0.5045	0.5532
Bi-GRU	0.5780	0.4078	0.4772	0.6281	0.5381	0.5794
Our model	0.6245	**0.5185**	0.5658	0.6917	**0.6325**	0.6605
Ensemble	**0.6451**	0.5089	**0.5687**	**0.7201**	0.6189	**0.6654**

The Integrated approach achieves the highest result among CRF-based methods on two datasets. The great improvements of the F-measure demonstrate that it is useful to integrate both discrete and neural features into a framework because both kinds of features can complement each other.

From the performances of approaches in second part, we can observe that GRU performances better than the worst system on two datasets without any handcraft features, which demonstrates that recurrent neural networks with label inference is an alternative approach for target extracting. Bi-GRU outperforms GRU about 2.03% and 2.62% on English and Spanish datasets respectively. Compared with GRU, Bi-GRU incorporates both forward information and backward information, we can infer that bidirectional information plays great important roles in modeling the boundary features in sequence labeling. However, the performances of Bi-GRU are much worse than the Integrated model. This phenomenon implies that only bidirectional information is not enough for target extracting and more useful features should be added to Bi-GRU.

Finally, we can see that our model achieves the state-of-the-art on two datasets, which demonstrates the effectiveness of our model. This validates that character-level features and context features have great influence on target extracting. We also use an ensemble of 6 our models and improve the performance about 0.29% and 0.49% on two datasets respectively. We can also observe that ensemble model can greatly improve the precision but has negative effect on the recall compared with single model.

3.4 Model Variants

In this subsection, we design a series of variants to validate the effectiveness of our model. The first variant is *SBi-GRU* which contains a SBi-GRU and does not contain character-level stack bidirectional GRU and context CNN. The second variant is *SBi-GRU-Context* which incorporates context information of each word by CNN, and the last variant *SBi-GRU+Character* integrates character-level features into SBi-GRU via applying stack bidirectional GRU to character sequence. Table 2 shows the results of our model and its variants.

From the Table 2, we can see that SBi-GRU+Context improves the performances about 0.91% and 0.11% on two datasets compared with SBi-GRU model. This verifies that the context information of each word promotes the performance

Table 2. Performance of the variants of our model.

Model/Dataset	English			Spanish		
	P	R	F	P	R	F
SBi-GRU	0.5728	0.4152	0.4806	0.6458	0.5328	0.5837
SBi-GRU+Context	0.5682	0.4294	0.4897	0.6393	0.5391	0.5848
SBi-GRU+Character	0.5843	**0.5313**	0.5561	0.6773	0.6324	0.6538
Our model	**0.6245**	0.5185	**0.5658**	**0.6917**	**0.6325**	**0.6605**

of target extracting indeed because the boundary of target is highly related to surrounding words. We can also see that context information have positive effect on improving recall and is little harmful to the precision. But higher recall means that system can cover more existing target, good system generally has higher F-measure and similar precision and recall.

Compared with SBi-GRU and SBi-GRU+Context, SBi-GRU+Character improves both precision and recall and outperforms SBi-GRU about 7.55% and 7.01%, which demonstrates that character-level features are very important to target extracting because character-level features include morphological characteristics and grammatical features. Further, character-level features can address OOV words problems in some degree.

Our model integrates character-level features and context information into SBi-GRU and achieve the best performances. From the results, we can see that the improvements from SBi-GRU to our model are greater the accumulation of the improvements from SBi-GRU to SBi-GRU+Context and from SBi-GRU to SBi-GRU+Character ($8.52\% > 0.91\% + 7.55\%$ and $7.68\% > 0.11\% + 7.01\%$), and we can infer that the character-level features and context information can complement each other without negative effects in target extracting.

In a word, context information and character-level features play an important role in target extracting, and we can integrate them into SBi-GRU for better performances.

3.5 Error Cases

In this subsection, we will show some error cases in English dataset predicted by our model to show the shortages of our model. Figure 2 shows the error cases.

(1) sergio aguero greets man city fans at the etihad stadium
(2) sergio aguero greets man city fans at the etihad stadium
(3) check out my personal newspaper on the tweeted times
(4) check out my personal newspaper on the tweeted times

Fig. 2. Error cases. Red, yellow, blue and green denote begin word of Person, inside word of Person, begin word of Organization, and inside word of Organization respectively. (1) and (3) are the correct labeling sentence. (2) and (4) are labeled by our model. (Color figure online)

There are main four kinds of errors caused by our model. The first error case is **sergio aguero** which should be a person name but is recognized as organization by our model. From the first error case, we can see that our model has ability to correctly recognizes the boundary of target but does not match the true target type. The reason may be that context information is not enough and we need take the long distance information into account.

The second error case caused by our model is **man city** which should be a organization target while is ignored by our model. In fact, *man* is the abbreviation of *manchester*, and *man* is not very common to be a target word. However, *city* is often regarded as inside word of target. This error case implies that our model is not good at finding new target via obvious clues.

The third error case is **etihad stadium** which should not be a target and is labeled as an organization by our model. *etihad stadium* does not appear in train data and is an existing place, but it should not be regarded as a target. Although our model does not correctly identify this non-target, it shows our model have potential in find new words.

The final error case is **the tweeted times**, our model only recognizes *tweeted*, *times* and misses *the*. This target contains a very common used function word *the* which almost does not be regarded as a part of target. Our model may learn the information about *the* and gives the wrong label. To avoid this kind of error, we need to let our model to associate with the word collection and word context.

From four error cases above, we can observe that our model can achieve comparable results but still has some problems illustrated above. The main reason may be that our model is lack of modeling higher level features like long distance features, word collection, etc., which are key factors for target extracting.

4 Related Work

Target extracting is a fundamental task for target-based sentiment analysis [7,8]. Early works often used unsupervised approaches which rely on predefined rules and handcraft features. For example, [21] introduced an unsupervised information extraction system which mines reviews in order to build a model of important product features, their evaluation by reviewers and their relative quality across products. [14] proposed a unsupervised approach which consists of two forms of recommendations based on semantic similarity and aspect associations respectively for aspect extraction. [25] developed a model to extract aspect term via unsupervised learning of distributed representations of words and dependency paths.

As supervised learning methods, [24] solved the target extraction by introducing a phrase dependency parsing which segments an inputs sentence into phrases and links segments with directed arcs. [15] developed a centering theory to extracting explicit and implicit opinion target from new comments. [22] proposed a double propagation method which propagates information between opinion words and targets to extract target and opinion word. [13] developed a partially-supervised word alignment model to mine opinion relation and used graph-based algorithm to estimate the confidence of candidate.

Recently, Target extracting is often regarded as sequence labeling task. The sequence labeling approaches mainly focus on Hidden Markov Model (HMM) and CRF-based framework. For example, [10] proposed a lexicalized HMM-based framework to extract specific product related entities expressing reviewers' opinion. [12] proposed a CRF-based framework which employs features to

extract opinion and object features for review sentence. [9] modeled opinion target extraction as information extraction task and address it by conditional random fields. [19] extracted name entities and their sentiment jointly by CRF.

With the development of neural networks (NN) approaches, NN methods also achieve good performances on target extracting. [26] developed a model which integrates discrete features and word embeddings into CRF to jointly extract target and their opinions. [23] built a joint model which integrates recursive neural networks and conditional random fields into a unified framework to explicit aspect and opinion term extraction. Although above NN approaches achieve good performance, they do not take the advantage of recurrent neural networks on sequence labeling [2]. Therefore, we use stack bidirectional GRU with label inference which integrates character-level features and context features to tackle target extracting, and experimental results on two open-domain datasets validate the effectiveness of our model.

5 Conclusion

In this paper, we propose to use stack bidirectional GRU (SBi-GRU) with label inference to address target extracting for target-based sentiment analysis. The first step of our model is to use SBi-GRU to model each word's character-level features which have great influence on target extracting. Then, SBi-GRU is also used to learn long distance feature for each word on the concatenation of character-level features and word embeddings, and convolution neural networks are adopted to capture local features around word. Finally, local features with outputs from sentence-level SBi-GRU are used to infer the target. Experiments on two datasets show the effectiveness of our model and verify the effectiveness of character-level and local features. Error cases imply the shortages of our model, in future work, we will explore how to learn global features for extracting target.

Acknowledgments. We would like to thank the anonymous reviewers for their insightful suggestions. Our work is supported by National Natural Science Foundation of China (No. 61370117). The corresponding author of this paper is Houfeng Wang.

References

1. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. JMLR **3**, 1137–1155 (2003)
2. Chen, X., Qiu, X., Zhu, C., Liu, P., Huang, X.: Long short-term memory neural networks for Chinese word segmentation. In: EMNLP, pp. 1197–1206 (2015)
3. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
4. Cieliebak, M., Deriu, J., Egger, D., Uzdilli, F.: A twitter corpus and benchmark resources for German sentiment analysis. In: SocialNLP, p. 45 (2017)
5. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. JMLR **12**, 2493–2537 (2011)

6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS, pp. 249–256 (2010)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: SIGKDD, pp. 168–177. ACM (2004)
8. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: AAAI, vol. 4, pp. 755–760 (2004)
9. Jakob, N., Gurevych, I.: Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In: EMNLP, pp. 1035–1045 (2010)
10. Jin, W., Ho, H.H., Srihari, R.K.: A novel lexicalized hmm-based learning framework for web opinion mining. In: ICML, pp. 465–472 (2009)
11. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Li, F., et al.: Structure-aware review mining and summarization. In: ACL, pp. 653–661 (2010)
13. Liu, K., Xu, H.L., Liu, Y., Zhao, J.: Opinion target extraction using partially-supervised word alignment model. In: IJCAI, vol. 13, pp. 2134–2140 (2013)
14. Liu, Q., Liu, B., Zhang, Y., Kim, D.S., Gao, Z.: Improving opinion aspect extraction using semantic similarity and aspect associations. In: AAAI, pp. 2986–2992 (2016)
15. Ma, T., Wan, X.: Opinion target extraction in Chinese news comments. In: Coling, pp. 782–790 (2010)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
17. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech, vol. 2, p. 3 (2010)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119. Curran Associates, Inc. (2013)
19. Mitchell, M., Aguilar, J., Wilson, T., Van Durme, B.: Open domain targeted sentiment. In: ENMLP, pp. 1643–1654 (2013)
20. Pennington, J., Socher, R., Manning, C.: GloVe: global vectors for word representation. In: EMNLP, pp. 1532–1543 (2014)
21. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Kao, A., Poteet, S.R. (eds.) Natural Language Processing and Text Mining, pp. 9–28. Springer, London (2007). https://doi.org/10.1007/978-1-84628-754-1_2
22. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)
23. Wang, W., Pan, S.J., Dahlmeier, D., Xiao, X.: Recursive neural conditional random fields for aspect-based sentiment analysis. arXiv preprint arXiv:1603.06679 (2016)
24. Wu, Y., Zhang, Q., Huang, X., Wu, L.: Phrase dependency parsing for opinion mining. In: EMNLP, pp. 1533–1541 (2009)
25. Yin, Y., Wei, F., Dong, L., Xu, K., Zhang, M., Zhou, M.: Unsupervised word and dependency path embeddings for aspect term extraction. arXiv preprint arXiv:1605.07843 (2016)
26. Zhang, M., Zhang, Y., Vo, D.T.: Neural networks for open domain targeted sentiment. In: EMNLP, pp. 612–621 (2015)