Improving Aspect Identification with Reviews Segmentation

Tianhao Ning[1,2], Zhen Wu[1,2], Xin-Yu Dai[1,2(✉)], Jiajun Huang[1,2], Shujian Huang[1,2], and Jiajun Chen[1,2]

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
{ningth,wuz,huangjj}@nlp.nju.edu.cn, {daixinyu,yincy,huangsj,chenjj}@nju.edu.cn

[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China

Abstract. Aspect identification, a key sub-task in Aspect-Based Sentiment Analysis (ABSA), aims to identify aspect categories from online user reviews. Inspired by the observation that different segments of a review usually express different aspect categories, we propose a reviews-segmentation-based method to improve aspect identification. Specifically, we divide a review into several segments according to the sentence structure, and then automatically transfer aspect labels from the original review to its derived segments. Trained with the new constructed segment-level dataset, a classifier can achieve better performance for aspect identification. Another contribution of this paper is extracting alignment features, which can be leveraged to further improve aspect identification. The experimental results show the effectiveness of our proposed method.

Keywords: Aspect identification · Reviews segmentation
Alignment features

1 Introduction

Sentiment analysis and opinion mining have drawn increasing attention in recent years because of the rapid growth of user-generated reviews on the Internet. For a product, users usually evaluate it from multiple aspects in a review. For example, a review "*Get this computer for portability and fast processing!!!*" of laptop domain contains two aspects, namely *portability* and *cpu operation performance*. So instead of classifying the overall sentiment of a review into binary polarity (positive or negative), a finer-grained task, known as Aspect-Based Sentiment Analysis (ABSA) [16], is proposed to discover more detailed entities, attributes, and emotions of users towards various aspects from reviews. In ABSA, a key sub-task is to identify aspect categories from reviews before sentimental polarity can be predicted towards each aspect.

Supported by the NSFC (No. 61472183, 61672277, 61772261).

M. Zhang et al. (Eds.): NLPCC 2018, LNAI 11108, pp. 416–428, 2018.
https://doi.org/10.1007/978-3-319-99495-6_35

For a specific domain of product or service, the set of aspect categories is usually predefined as E#A, where E is an entity and A is an attribute of E [10] such as *LAPTOP#PRICE*. Users usually express opinions toward multiple aspect categories in a review. Thus aspect identification can be formulated as a multi-label classification problem. Some previous works focus on designing the classification models and feature representations [17–19] and obtain some competitive results.

Different from previous works, we observe that different segments of a review usually express different aspect categories. For example, as shown in Table 1, the review "*Fantastic for the price, but the keys were not illuminated.*" can be divided into two segments, namely "*Fantastic for the price.*" and "*But the keys were not illuminated.*". These two segments are mutually independent, the former segment expresses aspect category *LAPTOP#PRICE* and the latter describes aspect category *KEYBOARD#DESIGN_FEATURES*. The example shows that each segment and its aspect categories have finer-grained mapping relation than the whole review and overall aspect categories have. Therefore, we claim that classification performance can be improved if we obtain the finer-grained mapping dataset, because we do not need to consider the interference from other segments when dealing with current segment.

To address the issues, we propose a reviews-segmentation-based method to divide a review into multiple segments, and then transfer aspect labels from the original review to corresponding segments automatically. These two steps will help us construct a review-segment-level labeled dataset with finer-grained mapping relation. After reviews segmentation and labels transferring, like solving other classification problems, we train a classifier on the constructed dataset for predicting aspect categories of new reviews. In this paper, we use Long Short-Term Memory (LSTM) [5] as classifier.

In addition, we also observe that in reviews some words have strong indication for aspects. For example, in the review of Table 1, the word "*price*" expresses the aspect category *LAPTOP#PRICE*, the words "*keys*" and "*illuminated*" indicate the aspect category *KEYBOARD#DESIGN_FEATURES*. However, due to the sparseness of the training data, it is hard to learn some sparse words like "*illuminated*" as important features to identify some aspect categories. Therefore, we introduce alignment algorithm in machine translation to extract alignment

Table 1. Examples of aspect categories identification.

Review: Fantastic for the price, but the keys were not illuminated **Aspects:** LAPTOP#PRICE, KEYBOARD#DESIGN_FEATURES
Segment_1: Fantastic for the price **Aspects:** LAPTOP#PRICE **Segment_2:** But the keys were not illuminated **Aspects:** KEYBOARD#DESIGN_FEATURES

features between words and aspect categories, which are used for further improving the aspect identification performance.

The main contributions of our work can be summarized as follows:

1. We improve the performance of aspect identification with reviews segmentation. Especially, we propose an effective method to divide a review into multiple segments and transfer the aspect labels from reviews to the corresponding segments.
2. We introduce the alignment algorithm in machine translation to extract the alignment features to further improve aspect identification.

2 Related Work

The ABSA task was added to the SemEval challenges since 2014 [11]. The subtask aspect identification of ABSA predefines aspect categories for a specific domain, so it can be regarded as a multi-label classification problem. Some early works employ traditional features and classification algorithms for aspect identification. [6] follows the one-vs-all strategy and build a binary Support Vector Machine (SVM) [2] classifier with ngrams and lexicon features for each aspect category. However, if a token implying an aspect, e.g., "expensive", is not taken as a feature, the SVM classifier cannot correctly identify its corresponding category. Therefore, [21] enhances the results from the SVM classifier by using implicit aspect indicators [4]. In addition, Maximum Entropy is also adopted for aspect identification with bag-of-words-like features (e.g. words, lemmas) [14].

Recently, neural network based models are explored to solve this problem. [17] extracts lexicon, syntax and word cluster as features, and trains a binary single layer feedforward network for each aspect category. [18] enhances the system of [17] by adding neural network features learned from a Deep Convolutional Neural Network system [15]. Different from previous works, [13] does not use traditional hand-crafted features, and directly train a convolutional neural network to output probability distributions over all aspect categories.

However, the above works all pay attention to designing hand-crafted features and classification models, but ignore the phenomenon that different segments of a review usually expressed different aspect categories, which motivates our work.

3 Method

3.1 Overview of Our Method

From our observation mentioned in Sect. 1, we have strong motivation to train a segment-level classifier to capture finer-grained mapping relation. To achieve the goal, we propose an effective method to build the corresponding segment-level dataset from an original review-level dataset.

Firstly, we use reviews segmentation method according to punctuations or dependency parsing tree to divide a long review into multiple segments. As Fig. 1

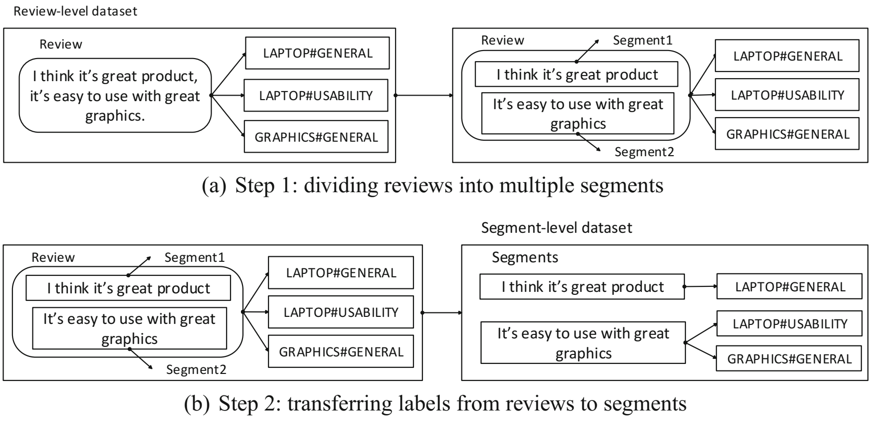


Fig. 1. Reviews segmentation and labels transferring.

shows, in the step 1 the review "*I think it's great product, it's easy to use with great graphics.*" will be divided into two segments, namely "*I think it's great product.*" and "*It's easy to use with great graphics.*".

Secondly, we train an LSTM classifier on the original review dataset and design some conservative rules (refer to Algorithm 1) to transfer the aspect labels from reviews to the corresponding segments. In above example, the label *LAPTOP#GENERAL* will be transferred to the segment "*I think it's great product.*". The labels *LAPTOP#USABILITY* and *GRAPHICS#GENERAL* will be transferred to the other segment "*It's easy to use with great graphics.*". After labels transferring, we will have a segment-level dataset.

Finally, a new LSTM classifier will be trained on the constructed segment-level dataset for predicting aspect labels of new reviews.

3.2 Reviews Segmentation

Reviews Segmentation with Punctuations. In linguistics, a clause is the smallest grammatical unit that can express a complete proposition. Sometimes some sentences themselves are clauses. The simplest reviews segmentation approach is to divide a review into several clauses according to punctuations. However, this approach does not work when there is no punctuation in the sentence. For instance, there is no punctuation in the review "*It's more expensive but well worth it in the long run*", whereas it has two clauses "*It's more expensive*" and "*well worth it in the long run*". These two clauses express different aspects. The similar sentences are quite common in real reviews.

Reviews Segmentation with Dependency Parsing Tree. The above example shows that we cannot divide a review into multiple segments when there is no punctuation in the sentence. Therefore, we need to consider more structural information. Here we present two typical cases in which there are multiple aspects. In the review "*I like the food but the waiter was rude.*", the first clause "*I like the food*" describes the aspect category *FOOD#GENERAL*, and the second one "*the waiter was rude*" expresses the aspect category *SERVICE#QUALITY*. In this example, the two aspects are expressed in two independent clauses. For another review "*Get this computer for portability and fast processing!*", the word "*portability*" indicates the aspect category *LAPTOP#PORTABILITY*, and the word "*fast processing*" expresses the aspect category *CPU#OPERATION_PERFORMANCE*. Obviously, the two aspects are dispersed in syntactic coordinate structures. To divide this review, the common component "*Get this computer for*" needs to be replicated for each clause.

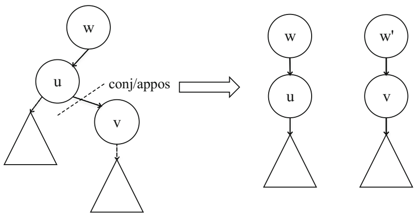


Fig. 2. Sentence segmentation with dependency parsing tree.

Fortunately, dependency parsing can address the above issues. Figure 2 shows how a sentence is divided into multiple relatively complete clauses with dependency parsing tree. We break dependency relations denoted by $\langle u, v\rangle$, whose dependency type is *conj* (conjunt) or *appos* (appositional modifier). More specifically, we denote all ancestors of u and their other descendants as w, except the subtree rooted by u. Let w' be the clone of w, then we append v and its descendants to w'. One exception is those sentences with compound predicates, like "*subject verb1 object1 and verb2 object2*". In this case, *verb1* is u and *verb2* is v, but *subject* is not an ancestor of *verb1* and needs to be appended to *verb2*. In our work, this sentence is divided into "*subject verb1 object1*" and "*subject verb2 object2*".

3.3 Labels Transferring from Reviews to Segments

After reviews segmentation, a complex review is divided into multiple segments. The next step is to transfer the aspect labels from original review to the corresponding segments. One way for labels transferring is to train a classifier (LSTM) on the original review-level dataset, and then predict the aspect labels of each segment. However, with this method, the quality of the constructed segment-level dataset is not guaranteed and heavily depends on the trained classifier. Therefore, based on the classifier trained on the review-level dataset, we add

Algorithm 1. Labels Transferring Algorithm

Require: A review r, its label bag b, and trained classifier f on review-level labeled dataset
Ensure: A set s of segments, and corresponding label bags B'

$s = \mathit{divided}(r)$, dividing r into segments according to the method in Section 3.2
if $|s| = 1$ **then**
    set $B' := \{b\}$
else
    set $B' := [\emptyset] * |s|$ {Initialize the label bags B' whose size is $|s|$.}
    for l in b **do**
        set $flag := False$ {The $flag$ means whether the label l is transferred to segments.}
        for $i = 1$ to $|s|$ **do**
            if $f(s_i)_l >= f(r)_l$ **then**
                add l to B'_i {Transfer label l to i-th segment.}
                set $flag := True$
            end if
        end for
        if $flag == False$ **then**
            if r not in s **then**
                add r to s {If label l is not transferred to segments, make sure that review r is in s.}
                $B'_{|s|+1} := [\emptyset]$ {$|s| + 1$ is the index of r in s}
            end if
            add l to $B'_{|s|+1}$ {Return the label l to the review r.}
        end if
    end for
    return s, B'
end if

some constraints (refer to Algorithm 1) for labels transferring to improve the quality of segment-level dataset.

Algorithm 1 demonstrates the pseudocode of Labels Transferring algorithm. Firstly, we train an LSTM classifier f on review-level labeled dataset. Let $f(x)_y$ be the predicted probability of aspect label y towards the input x. Then, to ensure the reliability of labels transferring, we set some constraints during the transferring. Specifically, for each aspect label l of a review r and each segment s_i divided from r, l would be transferred from r to s_i on condition that $f(s_i)_l >= f(r)_l$. This condition means that the segment s_i has stronger indication for aspect label l compared with the whole review r. If none of the segments satisfies the condition, we will return the label l to the review r and add the original review r and the label l to the new dataset. We use a classifier and fallback strategy to ensure that the labels are transferred to review segments as accurately as possible.

After labels transferring, we have a segment-level labeled dataset, with which we can train a more powerful classifier for aspect identification.

3.4 Alignment Feature

In reviews, some words or phrases have strong indication for expressed aspect categories. For example, for the review "*Fantastic for the price, but the keys were not illuminated.*" in Fig. 3, the word "*illuminated*" is quite significant for the identification of aspect category *KEYBOARD#DESIGN_FEATURES*. However, it is hard to learn the word "*illuminated*" as an effective feature for a classifier due to the sparseness of training data. Therefore, we employ alignment algorithm to extract alignment features between the words in reviews and expressed aspects to improve aspect identification.

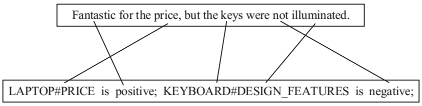


Fig. 3. An alignment example between a review and corresponding parallel data.

Firstly, we build the parallel data for extracting alignment features. The construction process is illustrated with an example. For the review in Fig. 3, we can obtain its paired labels (LAPTOP#PRICE, positive) and (KEYBOARD#DESIGN_FEATURES, negative) (The polarity of aspect is provided by original datasets). Then we rewrite the paired labels as "*LAPTOP#PRICE is positive; KEYBOARD#DESIGN_FEATURES is negative;*". In fact, the original review and the rewritten text are parallel and express the same meaning towards expressed aspects. With the parallel training data, we remove those stop-words and punctuations, and use Giza++ [9] to train IBM model 4 [3] to obtain bidirectional alignment probabilities, which contains the probability from words to aspects and the probability from aspects to words. We add these probabilities as alignment features to improve the performance of aspect identification.

3.5 Sequence Encoder and Aspect Identification

In this work, we adopt Long Short-Term Memory (LSTM) to encode reviews or review segments because of its excellent performance on sequence modeling. For a review or review segment consisting of n words $\{w_1, w_2, \cdots, w_n\}$, each word w_i is mapped to its embedding $\mathbf{w}_i \in \mathbb{R}^d$. LSTM network receives $[\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]$ and generates hidden states $[\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n]$. Then we concatenate the last hidden state $\mathbf{h}_n$ and alignment features in Sect. 3.4 as the final representation $\mathbf{r}$ for aspect identification. We use a linear layer to project representation $\mathbf{r}$ into the target space of C aspect categories. Since aspect identification is a multi-label classification problem, we add no-linear function sigmoid rather than softmax before calculating cross entropy loss:

$$p = \sigma\left(\mathbf{W}_r \mathbf{r} + \mathbf{b}_r\right), \tag{1}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ are weight matrix and bias vector respectively, every dimension p_a of p is in $[0, 1]$ and corresponds to the predicted probability of aspect category a. We set a threshold θ and make a prediction that a sample has a aspect label a when p_a exceeds the threshold θ. The loss function for optimization when training is defined as:

$$L = -\frac{1}{M}\sum_{m=1}^{M}\sum_{a=1}^{C}(p_a^g(d_m)\cdot\log(p_a(d_m)) + (1 - p_a^g(d_m))\cdot\log(1 - p_a(d_m))), \quad (2)$$

where p_a^g is the gold probability of aspect label a with ground truth being 1 and others being 0, M denotes the number of training data, d_m represents the m-th sample of training data. Finally, for a review we merge all the predicted results of its segments as the aspects of the whole review.

4 Experiments

4.1 Setup

- **Dataset:** We evaluate the effectiveness of the proposed method on SemEval-2015 task-12 from two domains (laptop and restaurant)[1]. Statistics of the original datasets are shown in Table 2. In these two datasets, all aspect categories in testing set exist in training set.
- **Preprocessing:** We use NLTK [1] to tokenize reviews and keep a vocabulary of 1500 most frequent words excluding stop-words. We use the dependency parser in Stanford CoreNLP [7] for reviews segmentation. Word embeddings are pretrained with skip-gram model [8] on the Yelp Phoenix Academic Dataset, which includes eighteen million user reviews[2] in restaurant domain.
- **Hyper-parameters selection:** We set word vector size to be 300. The dimensions of hidden states in LSTM are set to be 256. We train all models with AdaDelta [20]. The predicting thresholds θ are obtained via grid search in [0.1, 0.3] with increments of 0.01.
- **Metrics:** We use the precision and recall to compute F1-score as evaluation metrics of the performance of aspect identification.

Table 2. Statistics of the original datasets.

Dataset	Laptop-Train	Laptop-Test	Restaurant-Train	Restaurant-Test
Reviews	1739	761	1315	685
Aspects	81	58	12	12

[1] http://alt.qcri.org/semeval2015/task12/.
[2] https://www.yelp.com/dataset/challenge/.

4.2 Validating the Performance of Labels Transferring

It is obvious that the quality of new constructed datasets has a significant effect on the performance of aspect identification. To evaluate the performance of labels transferring, we randomly select 500 samples from segment-level dataset in restaurant domain, and invite three experience-rich annotators to manually annotate the aspect categories of every segment. The results are as the Table 3 shows.

Table 3. Performance of labels transferring from reviews to segments in restaurant domain.

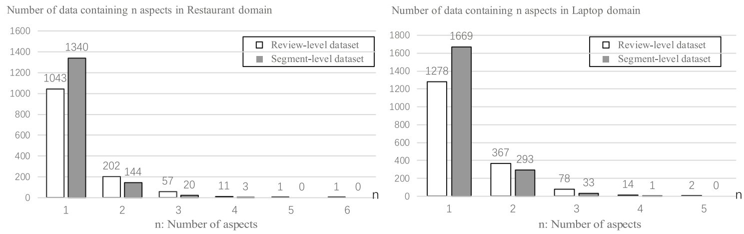
	Precision	Recall	F1-score
Dependency tree	0.9268	0.9172	0.9220
Punctuation	0.9451	0.8498	0.8949

We validate the performance of labels transferring based on two different reviews segmentation methods, namely punctuations and dependency tree. From Table 3, we can observe that even the simple review segmentation method based on punctuations achieves around 90% F1-score, which proves most of aspect labels are correctly transferred to corresponding segments. In addition, we obtain better results of labels transferring with dependency parsing. The precision, recall and F1-score are all above 91%. The results is reasonable because we consider more structural information with dependency parsing.

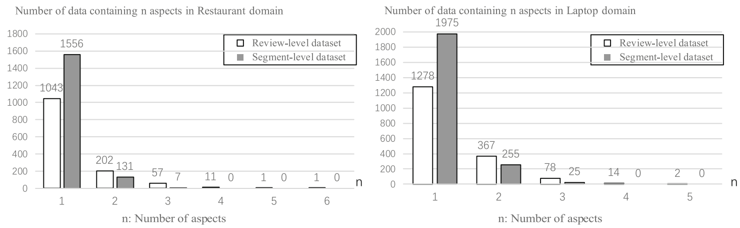
4.3 The Statistics of Reviews Segmentation

In our observation, different segments of a review usually express different aspect categories, which means we can obtain finer-grained mapping datasets after reviews segmentation and labels transferring. In order to validate the rationality of review segmentation, we count the number of data containing n aspects before and after reviews segmentation respectively. As Fig. 4(a) shows, after reviews segmentation and labels transferring, on both two domains, we have more samples with only one aspect, and fewer samples with two or more aspects. More expected results can be achieved after reviews segmentation with dependency parsing, as shown in Fig. 4(b). Compared to segmentation with punctuations, segmentation with dependency parsing achieves more segments with only one aspect.

Overall, Table 3 and Fig. 4 validate our observation that finer-grained correspondence between aspects and review segments exists. Therefore, we can build high-quality segment-level datasets with less interference from other segments after reviews segmentation and labels transferring.



(a) Reviews segmentation with punctuations.



(b) Reviews segmentation with dependency parsing.

Fig. 4. Number of data containing n aspects on two review datasets of SemEval-2015 before and after reviews segmentation respectively.

4.4 Baselines

We compare our method with several baseline methods for aspect identification:

- **TJUdeM:** TJUdeM [21] combines a SVM classifier with implicit aspect indicators. The SVM classifier uses words as features to determine the aspect categories. Additionally, they identify the implicit aspect indicators manually by setting a set of indicators for several aspects.
- **Sentiue:** Sentiue [14] uses a separate Maximum Entropy classifier with bag-of-words-like features (e.g. words, lemmas) for each entity and each attribute. Subsequently, heuristics are applied to the output of the classifiers to determine which aspect categories will be assigned to each sentence.
- **NLANGP:** NLANGP [17] is the winning system of SemEval-2015 task-12 and achieved the best performance in two domains. They train a sigmoid feedforward network as a classifier respectively for each aspect category. They use features containing bag-of-word, n-grams, parsing, and word embeddings learnt from Amazon and Yelp data [12].
- **LSTM:** We train an LSTM classifier on original datasets as one of our baselines.

4.5 Effectiveness of Reviews Segmentation

We use RS as the abbreviation of reviews segmentation. The experimental results are shown in Table 4. We can observe that in restaurant domain LSTM without any feature engineering outperforms the traditional classification model TJUdeM, Sentinue and the winning model NLANGP of SemEval-2015 Task 12.

Table 4. Effectiveness of reviews segmentation for aspect identification on two datasets. On the basis of LSTM, RS_Punc represents reviews segmentation with punctuations, and RS_Tree denotes reviews segmentation with dependency parsing. "**" means that LSTM+RS_Punc and LSTM+RS_Tree are significantly better than LSTM with 99% t-test.

Models	Restaurant			Laptop		
	Precision	Recall	F1-score	Precision	Recall	F1-score
TJUdeM	0.4782	0.5806	0.5245	0.4489	0.4821	0.4649
Sentiue	0.6330	0.4720	0.5410	0.5770	0.4410	0.5000
NLANGP	0.6386	0.6155	0.6268	0.6425	0.4209	0.5086
LSTM	0.6919	0.5809	0.6315	0.6009	0.4241	0.4972
LSTM+RS_Punc	0.6895	0.6219	0.6540**	0.6135	0.4420	0.5138**
LSTM+RS_Tree	0.6895	0.6361	**0.6617****	0.5899	0.4673	**0.5215****

In addition, compared to the baseline methods, LSTM+RS_Punc and LSTM+RS_Tree achieve significant improvements on the two datasets. Especially, LSTM+RS_Tree improves the performance over the LSTM by 3.02% on Restaurant dataset and 2.43% on Laptop dataset in F1-score. The results show that reviews segmentation is effective for aspect identification.

Compared with LSTM+RS_Punc, the model LSTM+RS_Tree achieves better performance on two datasets. The comparison shows that reviews segmentation with dependency parsing is more reasonable because more structural information is considered.

4.6 Effectiveness of Alignment Features

To validate the effectiveness of alignment features, we also report the results of our model incorporating alignment features into representation of review segments. As shown in Table 5, the LSTM+RS_Punc+align and LSTM+RS_Tree+align are our models containing alignment features. When we add alignment features to representation, more promising results are achieved in both two domains. The results show that alignment features can strengthen the connection between some key words and aspect categories. For example, in the review segment "*excellent speed for processing data.*", the alignment probability from "*speed*" to *LAPTOP#PERFORMANCE* is 0.9533. With the probability, it is quite possible that the review segment is assigned with the aspect category *LAPTOP#PERFORMANCE*, which can help improve classifier's performance.

Table 5. Effectiveness of alignment features for aspect identification on two datasets. The "+align" represents the model using alignment features. "*" means that LSTM+RS_Punc+align and LSTM+RS_Tree+align are significantly better than other no alignment features methods with 95% t-test.

Models	Restaurant			Laptop		
	Precision	Recall	F1-score	Precision	Recall	F1-score
LSTM+RS_Punc	0.6895	0.6219	0.6540	0.6135	0.4420	0.5138
LSTM+RS_Punc+align	0.7076	0.6245	**0.6635***	0.6667	0.4346	**0.5262***
LSTM+RS_Tree	0.6895	0.6361	0.6617	0.5899	0.4673	0.5215
LSTM+RS_Tree+align	0.7074	0.6335	**0.6685***	0.6376	0.4620	**0.5358***

5 Conclusion

In this work, we propose a reviews-segmentation-based method to improve aspect identification. Specifically, we firstly divide a review into multiple segments, then propose an algorithm to transfer the aspects from the original reviews to the corresponding segments. With the segment-level dataset, we can train a more powerful classifier for aspect identification. For better identification, we also introduce the alignment algorithm in machine translation to extract alignment probabilities. With our proposed method and novel alignment features, promising results are achieved on two benchmark datasets.

References

1. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, pp. 69–72. Association for Computational Linguistics (2006)
2. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory, pp. 144–152. ACM (1992)
3. Brown, P.F., Pietra, V.J.D., Pietra, S.A.D., Mercer, R.L.: The mathematics of statistical machine translation: parameter estimation. Comput. Linguist. **19**(2), 263–311 (1993)
4. Cruz, I., Gelbukh, A.F., Sidorov, G.: Implicit aspect indicator extraction for aspect based opinion mining. Int. J. Comput. Linguist. Appl. **5**(2), 135–152 (2014)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
6. Kiritchenko, S., Zhu, X., Cherry, C., Mohammad, S.: NRC-Canada-2014: detecting aspects and sentiment in customer reviews. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 437–442 (2014)
7. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: ACL (System Demonstrations), pp. 55–60 (2014)
8. Mikolov, T., Yih, W.t., Zweig, G.: Linguistic regularities in continuous space word representations. In: HLT-NAACL, pp. 746–751 (2013)

9. Och, F.J., Ney, H.: Giza++: training of statistical translation models (2000)
10. Pontiki, M., Galanis, D., Papageorgiou, H., Manandhar, S., Androutsopoulos, I.: SemEval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 486–495. Association for Computational Linguistics, Denver (2015)
11. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: SemEval-2014 task 4: aspect based sentiment analysis. In: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pp. 27–35. Association for Computational Linguistics (2014)
12. Qiu, G., Liu, B., Bu, J., Chen, C.: Opinion word expansion and target extraction through double propagation. Comput. Linguist. **37**(1), 9–27 (2011)
13. Ruder, S., Ghaffari, P., Breslin, J.G.: INSIGHT-1 at SemEval-2016 task 5: deep learning for multilingual aspect-based sentiment analysis. arXiv preprint arXiv:1609.02748 (2016)
14. Saias, J.: Sentiue: target and aspect based sentiment analysis in SemEval-2015 task 12. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 767–771. Association for Computational Linguistics, Denver, June 2015
15. Severyn, A., Moschitti, A.: UNITN: training deep convolutional neural network for twitter sentiment classification. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 464–469 (2015)
16. Thet, T.T., Na, J.C., Khoo, C.S.: Aspect-based sentiment analysis of movie reviews on discussion boards. J. Inf. Sci. **36**(6), 823–848 (2010)
17. Toh, Z., Su, J.: NLANGP: supervised machine learning system for aspect category classification and opinion target extraction. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 496–501 (2015)
18. Toh, Z., Su, J.: NLANGP at SemEval-2016 task 5: improving aspect based sentiment analysis using neural network features. In: Proceedings of SemEval, pp. 282–288 (2016)
19. Xenos, D., Theodorakakos, P., Pavlopoulos, J., Malakasiotis, P., Androutsopoulos, I.: AUEB-ABSA at SemEval-2016 task 5: ensembles of classifiers and embeddings for aspect based sentiment analysis. In: International Workshop on Semantic Evaluation, pp. 312–317 (2016)
20. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
21. Zhang, Z., Nie, J.Y., Wang, H.: TJUdeM: a combination classifier for aspect category detection and sentiment polarity classification. In: SemEval-2015, p. 772 (2015)