Joint Opinion Relation Detection Using One-Class Deep Neural Network

Liheng Xu, Kang Liu and Jun Zhao
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{lhxu, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Detecting opinion relation is a crucial step for fine-gained opinion summarization. A valid opinion relation has three requirements: a correct opinion word, a correct opinion target and the linking relation between them. Previous works prone to only verifying two of these requirements for opinion extraction, while leave the other requirement unverified. This could inevitably introduce noise terms. To tackle this problem, this paper proposes a joint approach, where all three requirements are simultaneously verified by a deep neural network in a classification scenario. Some seeds are provided as positive labeled data for the classifier. However, negative labeled data are hard to acquire for this task. We consequently introduce one-class classification problem and develop a One-Class Deep Neural Network. Experimental results show that the proposed joint approach significantly outperforms state-of-the-art weakly supervised methods.

1 Introduction

Opinion summarization aims to extract and summarize customers' opinions from reviews on products or services (Hu and Liu, 2004; Cardie et al., 2004). With the rapid expansion of e-commerce, the number of online reviews is growing at a high speed, which makes it impractical for customers to read throughout large amounts of reviews to choose better products. Therefore, it is imperative to automatically generate opinion summarization to help customers make more informed purchase decisions, where detecting opinion relation is a crucial step for opinion summarization.

Before going further, we first introduce some notions. An *opinion relation*, is a triple $o = (s, t, r)$, where three factors are involved: s is an *opinion word* which refers to those words indicating sentiment polarities; t is an *opinion target*, which can be any entity or aspect of an entity about which an opinion has been expressed; r refers to the linking relation between s and t. As in Example 1, s={*clear*}, t={*sceen*}, and there is a linking relation between the two words because *clear* is used to modify *screen*.

Example 1. *This mp3 has a clear screen.*

For a valid opinion relation, there are three requirements corresponding to the three factors: (i) the opinion word indicates sentiment polarity; (ii) the opinion target is related to current domain; (iii) the opinion word modifies the opinion target. Previous weakly supervised methods often expand a seed set and identify opinion relation either by co-occurrence statistics (Hu and Liu, 2004; Hai et al., 2012) or syntactic dependencies (Popescu and Etzioni, 2005; Qiu et al., 2009) following the assumption below.

Assumption 1. Terms that are likely to have linking relation with the seed terms are believed to be opinion words or opinion targets.

For example, if one has an opinion word seed *clear* (which satisfies requirement i), and one finds that it modifies the word *screen* in Example 1 (which satisfies requirement iii). Then one infers that *screen* is an opinion target according to Assumption 1 (whether *screen* is correct is not checked). However, in Example 2(a), we can see that *good* is an opinion word and it modifies *thing*, but *thing* is not related to

mp3 domain. If one follows Assumption 1, *thing* will be mistaken as an opinion target. Similarly, in Example 2(b), if one uses *mp3* to extract *another* as an opinion word, he may get an objective word.

Example 2. (a) *This mp3 has many good things.*   (b) *Just another mp3 I bought.*

The reason for the errors above is that Assumption 1 only verifies two requirements for an opinion relation. Unfortunately, this issue occurs frequently in online reviews. As a result, previous methods often suffer from these noise terms. To produce more precise opinion summary, it is argued that we shall follow a more restricted assumption as follows.

Assumption 2. The three requirements: the opinion word, the opinion target and the linking relation between them, shall be all verified during opinion relation detection.

To make accordance with Assumption 2, this paper proposes a novel joint opinion relation detection method, where opinion words, opinion targets and linking relations are simultaneously considered in a classification scenario. Following previous works, we provide a small set of seeds (i.e. opinion words or targets) for supervision, which are regarded as positive labeled examples for classification. However, negative labeled examples (i.e. noise terms) are hard to acquire, because we do not know which term is not an opinion word or target. This leads to One-Class Classification (OCC) problem (Moya et al., 1993). The key to OCC is semantic similarity measuring between terms, and Deep Neural Network (DNN) with word embeddings is a powerful tool for handling this problem. We consequently integrate DNN into a OCC classifier and develop a *One-Class Deep Neural Network* (OCDNN). Concretely, opinion words/targets/relations are first represented by embedding vectors and then *jointly* classified. Experimental results show that the proposed joint method which follows Assumption 2 significantly outperforms state-of-the-art weakly supervised methods which are based on Assumption 1.

2   Related Work

In opinion relation detection task, previous works often used co-occurrence statistics or syntax information to identify opinion relations. For co-occurrence statistical methods, Hu and Liu (2004) proposed a pioneer research for opinion summarization based on association rules. Popescu and Etzioni (2005) defined some syntactic patterns and used Pointwise Mutual Information (PMI) to extract product features. Hai et al. (2012) proposed an opinion feature mining method which employed Likelihood Ratio Tests (LRT) (Dunning, 1993) as the co-occurrence statistical measure. For syntax-based approaches, Riloff and Wiebe (2003) performed syntactic pattern learning while extracting subjective expressions. Zhuang et al. (2006) used various syntactic templates from an annotated movie corpus and applied them to supervised movie feature extraction. Kobayashi et al. (2007) identified opinion relations by searching for useful syntactic contextual clues. Qiu et al. (2009) proposed a bootstrapping framework called *Double Propagation* which introduced eight heuristic syntactic rules to detect opinion relations.

However, none of the above methods could verify opinion words/targets/relations simultaneously during opinion relation detection. To perform joint extraction, various models had been proposed, most of which employed classification or sequence labeling models, such as HMM (Jin and Ho, 2009), SVM (Wu et al., 2009) and CRFs (Breck et al., 2007; Jakob and Gurevych, 2010; Li et al., 2010). Besides, optimal models such as Integer Linear Programming (ILP) were also employed to perform joint inference for opinion extraction (Choi et al., 2006; Yang and Cardie, 2013).

Joint methods had been shown to achieve better performance than pipeline approaches. Nevertheless, most existing joint models rely on full supervision, which have the difficulty of obtaining annotated training data in practical applications. Also, supervised models that are trained on one domain often fail to give satisfactory results when shifted to another domain. Our method does not require annotated data.

3   The Proposed Method

To detect opinion relations, previous methods often leverage some seed terms, such as opinion word seeds (Hu and Liu, 2004; Baccianella et al., 2010) and opinion target seeds (Jijkoun et al., 2010; Hai et al., 2012). These seeds can be used as positive labeled examples to train a classifier. However, it is hard to get negative labeled examples for this task. Because opinion words or targets are often domain

dependent and words that do not bear any sentiment polarity in one domain may be used to express opinion in another domain. It is also very hard to specify in what case there is no linking relation between two words.

To deal with this problem, we employ one-class classification, and develop a One-Class Deep Neural Network (OCDNN) for opinion relation detection. The architecture of OCDNN is shown in Figure 1, which consists of two levels. The lower level learns feature representations unsupervisedly for opinion words/targets/relations, where the left component uses word embedding learning to represent opinion words/targets, and the right component maps linking relations to embedding vectors by a recursive autoencoder. Then the upper level uses the learnt features to perform one-class classification.
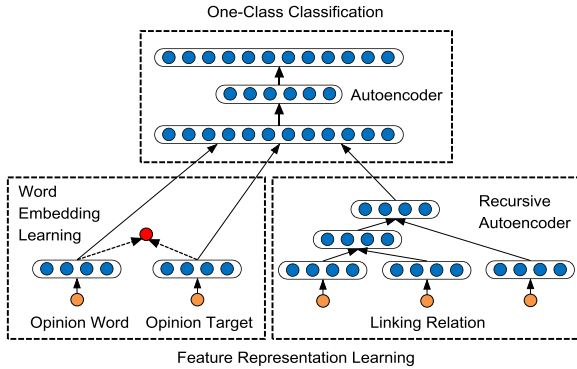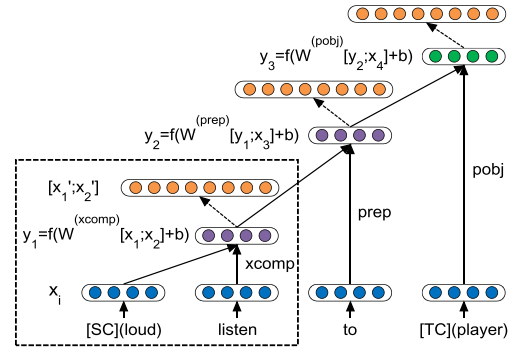
Figure 1: The architecture of OCDNN.

Figure 2: An example of recursive autoencoder.

3.1 Opinion Seed Generation

To obtain training data for OCDNN, we shall first get some seed terms as follows.

Opinion Word Seeds. We manually pick 186 domain independent opinion words from SentiWordNet (Baccianella et al., 2010) as the opinion word seed set SS.

Opinion Target Seeds. Likelihood Ratio Tests (LRT) (Dunning, 1993) used in (Hai et al., 2012) is employed to generate opinion target seeds. LRT aims to measure how greatly two terms T_i and T_j are associated with each other by sentence-level corpus statistics which is defined as follows,

$$LRT = 2[\log L(p_1, k_1, n_1) + \log L(p_2, k_2, n_2) - \log L(p, k_1, n_1) - \log L(p, k_2, n_2)] \qquad (1)$$

where $k_1 = tf(T_i, T_j), k_2 = tf(T_i, \bar{T}_j), k_3 = tf(\bar{T}_i, T_j), k_4 = tf(\bar{T}_i, \bar{T}_j), tf(\cdot)$ denotes term frequency; $L(p, k, n) = p^k(1 - p)^{n-k}$, $n_1 = k_1 + k_3$, $n_2 = k_2 + k_4$, $p_1 = k_1/n_1$, $p_2 = k_2/n_2$ and $p = (k_1 + k_2)/(n_1 + n_2)$. We measure LRT between a domain name (e.g. *mp3*, *hotel*, etc.) and all opinion target candidates. Then N terms with highest LRT scores are added into the opinion target seed set TS.

Linking Relation Seeds. Linking relation can be naturally captured by syntactic dependency, because it directly models the modification relation between opinion word and opinion target. We employ an automatic syntactic opinion pattern learning method called *Sentiment Graph Walking* (Xu et al., 2013) and get 12 opinion patterns with highest confidence as the linking relation seed set RS.

After seed generation, every opinion relation $s_o = (s_s, s_t, s_r)$ in review corpus that satisfies $s_s \in SS$, $s_t \in TS$ and $s_r \in RS$ is taken as a positive labeled training instance.

3.2 Opinion Relation Candidate Generation

The opinion term candidate set is denoted by $C = \{SC, TC\}$, where SC/TC represents opinion word/target candidate. Following previous works (Hu and Liu, 2004; Popescu and Etzioni, 2005; Qiu et al., 2009), we take adjectives or verbs as opinion word candidates, and take nouns or noun phrases as opinion target candidates. A statistic-based method in Zhu et al. (2009) is used to detect noun phrases.

An opinion relation candidate is denoted by $c_o = (c_s, c_t, c_r)$, where $c_s \in SC$, $c_t \in TC$, and c_r is a potential linking relation. To get c_r, we first get dependency tree of a sentence using Stanford Parser (de

Marneffe et al., 2006). Then, the shortest dependency path between a c_s and a c_t is taken as a c_r. To avoid introducing too many noise candidates, we constrain that there are at most four terms in a c_r.

3.3 Word Representation by Word Embedding Learning

Word embedding, a.k.a *word representation*, is a mathematical object associated with each word, which is often used in a vector form, where each dimension's value corresponds to a feature and might even have a semantic or grammatical interpretation (Turian et al., 2010). By word embedding learning, words are embedded into a hyperspace, where two words that are more semantically similar to each other are located closer. This characteristic is precisely what we want, because the key to one-class classification is semantic similarity measuring (illustrated in Section 3.5).

For word representation, we use a matrix $LT \in \mathbb{R}^{n \times |V_w|}$, where *i*-th column represents the embedding vector for term t_i, n is the size of embedding vector and V_w is the vocabulary of LT. Therefore, we can denote t_i by a binary vector $b_i \in \mathbb{R}^{|V_w|}$ and get its embedding vector by $x_i = LTb_i$. The training criterion for word embeddings is,

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{c \in C} \sum_{v \in V_w} max\{0, 1 - s_\theta(c) + s_\theta(v)\} \tag{2}$$

where θ is the parameters of neural network used for training. See Collobert et al. (2011) for the detailed implementation.

3.4 Linking Relation Representation by Using Recursive Autoencoder

The goal of this section is to represent the linking relation between an opinion word and an opinion target by a *n*-element vector as we do during word representation. Specifically, we combine embedding vectors of words in a linking relation by a recursive autoencoder (Socher et al., 2011) according to syntactic dependency structure. In this way, linking relations are no longer limited to the initial seeds during classification, because linking relations that are similar to the seed relations will have similar vector representations.

Figure 2 shows a linking relation representation process by an example: *too loud to listen to the player*. First, we get its dependency path between the opinion word c_s:*loud* and the opinion target c_t:*player*. Then c_s and c_t are replaced by wildcards *[SC]* and *[TC]* because they are not concerned in the linking relation. The dash line box in Figure 2 shows a standard autoencoder, which is a three-layer neural network, where the number of nodes in input layer is equal to that of output layer. It takes two *n*-element vectors as input and compresses semantics of the two vectors into one *n*-element vector in hidden layer by,

$$y = f(W^{(dep)}[x_1; x_2] + b), \ W^{(dep)} = \frac{1}{2}[I_1; I_2; I_b] + \epsilon \tag{3}$$

where $[x_1; x_2]$ is the concatenation of the two input vectors and f is the sigmoid function; $W^{(dep)}$ is a parameter matrix that is chosen according to the dependency relation between x_1 and x_2 (In the case of y_1, $W^{(dep)} = W^{(xcomp)}$), which is initialized by I_i, where I_i is a $n \times n$ unit matrix, I_b is a *n*-element null vector, and ϵ is sampled from a uniform distribution $U[-0.001, 0.001]$ (Socher et al., 2013). Then $W^{(dep)}$ are updated during training. The training criterion of autoencoder is to minimize Euclidean distance between the original input and its output,

$$E_{rae} = ||[x_1; x_2] - [x_1'; x_2']||^2 \tag{4}$$

where $[x_1'; x_2'] = W^{(out)}y$ and $W^{(out)}$ is initialized by $W^{(dep)T}$.

We always start the combination process from *[SC]* and it is repeated along the dependency path. For example, the result vector y_1 of the first combination is used as the input vector when computing y_2. Finally, the linking relation is represented by a *n*-element vector (the green vector in Figure 2).

3.5 One-Class Classification for Opinion Relation Detection

We represent an opinion relation candidate $c_o = (c_s, c_t, c_r)$ by a vector $v_o = [v_s; v_t; v_r]$, which is a concatenation of the opinion word embedding v_s, the opinion target embedding v_t and the linking relation embedding v_r. Then v_o is feed to the upper level autoencoder in Figure 1.

To perform one-class classification, the number of nodes in the hidden layer of the upper level autoencoder is constrained to be smaller than that of the input layer. By using such a "bottleneck" network structure, characteristics of the input are first compressed into the hidden layer and then reconstructed by the output layer (Japkowicz et al., 1995). Concretely, characteristics of positive labeled opinion relations are first compressed into the hidden layer, and then the autoencoder should be able to adequately reconstruct positive instances in the output layer, but should fail to reconstruct negative instances which present different characteristics from positive instances. Therefore, the detection of opinion relation is equivalent to assessing how well a candidate is reconstructed by the autoencoder. As the input vector v_o consists of representations for opinion words/targets/relations, characteristics of the three factors are *jointly* compressed by one hidden layer. Either false opinion word/target/relation will lead to failure of reconstruction. Consequently, our approach follows Assumption 2.

For opinion relation detection, candidates with reconstruction error scores that are smaller than a threshold ϑ are classified as positive. Determining the exact value of ϑ is very difficult. Inspired by other one-class approaches (Liu et al., 2002; Manevitz and Yousef, 2007), we introduce some negative opinion terms to help to estimate ϑ.[1] Although negative instances are hard to acquire, Xu et al. (2013) show that a set of general nouns (such as *thing*, *one*, etc., we denote them by GN) seldom appear to be opinion targets. One the other hand, we create a 50-opinion-word validation set SV from SentiWordNet.

To estimate ϑ, we first introduce a *positive proportion* (pp) score,

$$pp(t) = tf^+(t)/tf(t), \ t \in PE, \ PE = \{c_o | E_r(c_o) < \vartheta\} \tag{5}$$

where PE denotes the opinion relations that are classified as positive, $E_r(\cdot)$ is the reconstruction error of OCDNN and $tf^+(\cdot)$ is the frequency of term in PE. Then an error function E_ϑ is minimized, which balances between the proportion of non-target terms (GN) in PE (which shall be as small as possible) and the proportion of opinion words in validation set (SV) in PE (which shall be as large as possible).

$$E_\vartheta = \sum_{t \in GN \cap PE} [pp(t) - 0]^2 + \sum_{s \in SV \cap PE} [pp(s) - 1]^2 \tag{6}$$

3.6 Opinion Target Expansion

We apply bootstrapping to iteratively expand opinion target seeds. It is because the vocabulary of seed set is limited, which cannot fully represent the distribution of opinion targets. So we expand opinion target seeds in a self-training manner to alleviate this issue. After training OCDNN, all opinion relation candidates are classified, and opinion targets are ranked in descent order by,

$$s(t) = \log tf(t) \times pp(t). \tag{7}$$

Then, top M candidates are added into the target seed set TS for the next training iteration.

4 Experiments

4.1 Datasets and Evaluation Metrics

Datasets. Three real world datasets are selected for evaluation. The first one is called *Customer Review Dataset (CRD)* [2] which contains reviews on five products (denoted by D1 to D5). The second is a benchmark dataset (Wang et al., 2011) on *MP3* and *Hotel*[3]. The last one is crawled from www.amazon.com, which involves *Mattress* and *Phone*. Two annotating criteria are applied.

---

[1]This is not in contradiction with OCC problem, because these negative examples are NOT used during training.

[2]http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

[3]http://timan.cs.uiuc.edu/downloads.html

Annotation 1 is used to evaluate opinion words/targets extraction. Firstly, 10,000 sentences are randomly selected from reviews and all possible terms are extracted along with their contexts. Then, annotators are required to judge whether each term is an opinion word or an opinion target.

Annotation 2 is used to evaluate intra-sentence opinion relation detection. Annotators are required to carefully read through each sentence and find out every opinion relation, which consists of an opinion word, an opinion target, as well as the linking relation between them. The annotation is very labor-intensive, so only 5,000 sentences are annotated for *MP3* and *Hotel*.

Two annotators were required to annotate following the criteria above. When conflicts happened, a third annotator would make the final judgment. Note that Annotation 1 and Annotation 2 were annotated by two different groups. Detailed information of the annotated datasets are shown in Table 1. Furthermore, the kappa values between Annotation 1 and Annotation 2 are 0.88 for opinion words and 0.84 for opinion targets, showing highly substantial agreement.

Domain	#OW	#OT	Kappa_OW	Kappa_OT
Hotel	434	1,015	0.72	0.67
MP3	559	1,158	0.69	0.65
Mattress	366	523	0.67	0.62
Phone	391	862	0.68	0.64

(a) Annotation 1

Domain	#LR	#OW	#OT	Kappa_LR
Hotel	2,196	317	735	0.62
MP3	2,328	342	791	0.61

(b) Annotation 2

Table 1: The detailed information of Annotations. OW/OT/LR stands for opinion words/opinion targets/linking relations. The Kappa-values are calculated by using *exact* matching metric for Annotation 1 and *overlap* matching metric for Annotation 2.

Evaluation Metrics. We perform evaluation in terms of Precision(P), Recall(R) and F-measure(F) according to *exact* and *overlap* matching metrics (Wiebe et al., 2005). The *exact* metric is used to evaluate opinion word/target extraction, which requires exact string match. And the *overlap* metric is used to evaluate opinion relation detection, where an extracted opinion relation is regarded as correct when both the opinion word and the opinion target in it overlap with the gold standard.[4]

Evaluation Settings. Four state-of-the-art weakly supervised approaches are selected as competitors. Two are co-occurrence statistical methods and two are syntax-based methods, all of which follow Assumption 1.

AdjRule extracts opinion words/targets by using adjacency rules (Hu and Liu, 2004).

LRTBOOT is a bootstrapping algorithm which employs *Likelihood Ratio Tests* (Dunning, 1993) as the co-occurrence statistical measure (Hai et al., 2012).

DP denotes the *Double Propagation* algorithm (Qiu et al., 2009).

DP-HITS is an enhanced version of *DP* proposed by Zhang et al. (2010), which ranks terms by

$$s(t) = \log tf(t) \times importance(t) \tag{8}$$

where $importance(t)$ is estimated by the HITS algorithm (Kleinberg, 1999).

OCDNN is the proposed method. The target seed size $N = 40$, the opinion targets expanded in each iteration $M = 20$, and the max bootstrapping iteration number is $X = 10$. The representation learning in lower level of OCDNN is trained on the whole corpus, while the test data are the same for all settings. All results of OCDNN are taken by average performance over five runs with randomized parameters.

4.2 OCDNN vs. the State-of-the-art

We compare *OCDNN* with state-of-the-art methods for opinion words/targets extraction. In *OCDNN*, Eq. 7 is used to rank opinion words/targets. The results on *CRD* and the four domains are shown in Table 2 and Table 3. *DP-HITS* does not extract opinion words so their results for opinion words are not taken into account.

---

[4] Determining the exact boundaries of opinion terms is hard even for human (Wiebe et al., 2005), so we use this relaxation.

Opinion Targets																
Method	D1			D2			D3			D4			D5			Avg.
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	F
AdjRule	0.75	0.82	0.78	0.71	0.79	0.75	0.72	0.76	0.74	0.69	0.82	0.75	0.74	0.80	0.77	0.76
DP	**0.87**	0.81	**0.84**	**0.90**	0.81	**0.85**	**0.90**	0.86	**0.88**	**0.81**	0.84	**0.82**	**0.92**	0.86	**0.89**	**0.86**
DP-HITS	0.83	0.84	0.83	0.86	0.85	**0.85**	0.86	0.88	0.87	0.80	0.85	**0.82**	0.86	0.86	0.86	0.85
LRTBOOT	0.77	**0.87**	0.82	0.74	**0.90**	0.81	0.79	**0.89**	0.84	0.72	**0.88**	0.79	0.74	**0.88**	0.80	0.81
OCDNN	0.83	0.82	0.82	0.86	0.85	**0.85**	0.86	0.87	0.86	0.78	0.84	0.81	0.89	0.85	0.87	0.84
Opinion Words																
AdjRule	0.57	0.75	0.65	0.51	0.76	0.61	0.57	0.73	0.64	0.54	0.62	0.58	0.62	0.67	0.64	0.62
DP	**0.64**	0.73	0.68	0.57	0.79	0.66	0.65	0.70	0.67	0.61	0.65	0.63	**0.70**	0.68	**0.69**	0.67
LRTBOOT	0.60	**0.79**	0.68	0.52	**0.82**	0.64	0.60	**0.76**	0.67	0.56	**0.70**	0.62	0.66	**0.71**	0.68	0.66
OCDNN	**0.64**	0.77	**0.70**	**0.63**	0.79	**0.70**	**0.66**	0.73	**0.69**	**0.68**	**0.70**	**0.69**	**0.70**	0.69	**0.69**	**0.70**

Table 2: Results of opinion terms extraction on *Customer Review Dataset*.

Opinion Targets													
Method	MP3			Hotel			Mattress			Phone			Avg.
	P	R	F	P	R	F	P	R	F	P	R	F	F
AdjRule	0.53	0.55	0.54	0.55	0.57	0.56	0.50	0.60	0.55	0.52	0.51	0.51	0.54
DP	0.66	0.57	0.61	0.66	0.60	0.63	0.55	0.60	0.57	0.60	0.53	0.56	0.59
DP-HITS	0.65	0.62	0.63	0.64	0.66	0.65	0.55	0.67	0.60	0.62	0.64	0.63	0.63
LRTBOOT	0.60	**0.77**	0.67	0.59	**0.78**	0.67	0.55	**0.78**	0.65	0.57	**0.76**	0.65	0.66
OCDNN	**0.70**	0.68	**0.69**	**0.71**	0.70	**0.70**	**0.63**	0.69	**0.66**	**0.69**	0.68	**0.68**	**0.68**
Opinion Words													
AdjRule	0.48	0.65	0.55	0.51	0.68	0.58	0.51	0.68	0.58	0.48	0.61	0.54	0.56
DP	0.58	0.62	0.60	0.60	0.66	0.63	0.54	0.68	0.60	0.55	0.59	0.57	0.60
LRTBOOT	0.52	**0.69**	0.59	0.54	**0.74**	0.62	0.51	**0.73**	0.60	0.50	**0.68**	0.58	0.60
OCDNN	**0.68**	0.65	**0.66**	**0.70**	0.68	**0.69**	**0.59**	0.70	**0.64**	**0.63**	0.59	**0.61**	**0.65**

Table 3: Results of opinion terms extraction on the four domains.

From Table 2, we can see that our method outperforms co-occurrence-based methods *AdjRule* and *LRTBOOT*, but achieves comparable or a little worse results than syntax-based methods *DP* and *DP-HITS*. This is because *CRD* is quite small, which only contains several hundred sentences for each product review set. In this case, methods based on careful-designed syntax rules have superiority over those based on statistics (Liu et al., 2013). For results on larger datasets shown in Table 3, our method outperforms all of the competitors. Comparing *OCDNN* with *DP-HITS*, the two approaches use similar term ranking metrics (Eq. 7 and Eq. 8), but *OCDNN* significantly outperforms *DP-HITS*. Therefore, the *positive proportion* score estimated by *OCDNN* is more effective than the *importance* score in *DP-HITS*. Comparing *OCDNN* with *LRTBOOT*, we find that *LRTBOOT* achieves better recall but lower precision. This is because *LRTBOOT* follows Assumption 1 during bootstrapping, which suffers a lot from error propagation, while our joint classification approach effectively alleviates this issue. We will discuss the impact of error propagation in detail later.

4.3 Assumption 1 vs. Assumption 2

This section evaluates intra-sentence opinion relation detection, which is more useful for practical applications. It also reflects the impacts of Assumption 1 and Assumption 2. The results are shown in Table 4 and Table 5, where *OCDNN* significantly outperforms all competitors. The average improvement of F-measure over the best competitor is 6% on *CRD* and 9% on *Hotel* and *MP3*.

As Assumption 1 only verifies two of the requirements in an opinion relation, it would inevitably introduce noise terms during extraction. For syntax-based method *DP*, it extracts many false opinion relations such as *good thing* and *nice one* (where *thing* and *one* are false opinion targets) or objective expressions like *another mp3* and *every mp3* (which contain false opinion words *another* and *every*). For co-occurrence statistical methods *AdjRule* and *LRTBOOT*, it is very hard to deal with ambiguous linking relations. For example, in phrase *this mp3 is very good except the size*, co-occurrence statistical methods could hardly tell which opinion target does *good* modify (*mp3* or *size*). Our method follows Assumption

Method	D1			D2			D3			D4			D5			Avg.
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	F
AdjRule	0.51	0.66	0.58	0.53	0.63	0.58	0.50	0.61	0.55	0.48	0.60	0.53	0.50	0.61	0.55	0.56
DP	0.66	0.63	0.64	0.68	0.60	0.64	0.69	0.62	0.65	0.66	0.57	0.61	0.67	0.60	0.63	0.64
LRTBOOT	0.53	**0.70**	0.60	0.57	**0.72**	0.64	0.55	**0.69**	0.61	0.52	**0.70**	0.60	0.55	**0.68**	0.61	0.61
OCDNN	**0.76**	0.66	**0.71**	**0.74**	0.67	**0.70**	**0.77**	0.67	**0.72**	**0.70**	0.65	**0.67**	**0.77**	0.66	**0.71**	**0.70**

Table 4: Results of opinion relation detection on *Customer Review Dataset*.

Method	MP3			Hotel			Avg.
	P	R	F	P	R	F	F
AdjRule	0.49	0.55	0.52	0.45	0.53	0.49	0.50
DP	0.63	0.51	0.56	0.59	0.50	0.54	0.55
LRTBOOT	0.54	**0.63**	0.58	0.50	**0.60**	0.55	0.56
OCDNN	**0.73**	0.60	**0.66**	**0.70**	0.59	**0.64**	**0.65**

Table 5: Results of opinion relation detection on the two domains.

2, which verifies all three requirements for opinion word/target/relation in an opinion relation, so the above errors are greatly reduced. Therefore, Assumption 2 is more reasonable than Assumption 1.

4.4 The Effect of Joint Classification

We evaluate the three bootstrapping methods (*DP*, *LRTBOOT* and *OCDNN*) for opinion target expansion. The precision of each iteration is shown in Figure 3. We can see that *DP* and *LRTBOOT* gradually suffer from error propagation and the precision drops quickly along with the number of iteration increases. For *OCDNN*, although error propagation is inevitable, the precision curve retains at a high level. Therefore, the joint approach produces more precise results.

For more detailed analysis, we give a variation of the proposed method named *3NN*, which uses 3 individual autoencoders to classify opinion words/targets/relations separately. An opinion relation candidate is classified as positive only when the three factors are all classified as positive. Then opinion relations are ranked by the sum of reconstruction scores of the three factors. In the results of opinion relation detection, when the recall is fixed at 0.6, the precisions of *3NN* are 0.67 for *MP3* and 0.65 for *Hotel*, while the precisions of *OCDNN* are 0.73 for *MP3* and 0.70 for *Hotel*. Therefore, *OCDNN* achieves much better performance than *3NN*.

An example may explain the reason of why *3NN* gets worse performance. In our experiment on *Hotel*, a false opinion relation *happy day* is misclassified as positive by *3NN*. It is because the word *day* has a small reconstruction score in *3NN*. At the same time, *happy* is a correct opinion word, so the whole expression *happy day* also has a small reconstruction score and then be misclassified. In contrast, the reconstruction score of *happy day* from *OCDNN* is quite large so the phrase is dropped. The reason is that the joint approach captures the semantic of a whole phrase rather than its single components. Therefore, it is more reasonable.
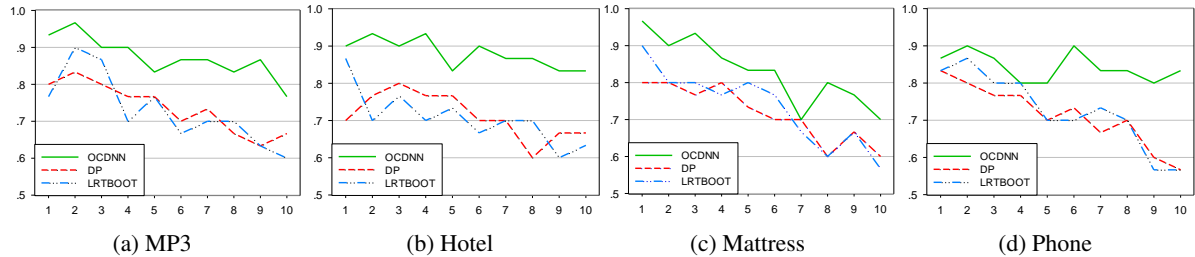

Figure 3: Precision (y-axis) of opinion target seed expansion at each bootstrapping iteration (x-axis).

5 Conclusion and Future Work

This paper proposes One-Class Deep Neural Network for joint opinion relation detection in one-class classification scenario, where opinion words/targets/relations are simultaneously verified during classification. Experimental results show the proposed method significantly outperforms state-of-the-art weakly supervised methods that only verify two factors in an opinion relation.

In future work, we plan to adapt our method and make it be capable of capturing implicit opinion relations.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Seventh conference on International Language Resources and Evaluation*, pages 2200–2204.

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2683–2688, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Claire Cardie, Janyce Wiebe, Theresa Wilson, and Diane Litman. 2004. Low-level annotations and summary representations of opinions for multi-perspective question answering.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 431–439, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*. The Stanford Natural Language Processing Group.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, March.

Zhen Hai, Kuiyu Chang, and Gao Cong. 2012. One seed to find them all: mining opinion features via association. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 255–264, New York, NY, USA. ACM.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1035–1045, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nathalie Japkowicz, Catherine Myers, and Mark Gluck. 1995. A novelty detection approach to classification. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI'95, pages 518–523, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 585–594, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wei Jin and Hung Hay Ho. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 465–472.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September.

Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting aspect-evaluation and aspect-of relations in opinion mining. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1065–1074, Prague, Czech Republic, June. Association for Computational Linguistics.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 653–661, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 387–394, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kang Liu, Liheng Xu, and Jun Zhao. 2013. Syntactic patterns versus word alignment: Extracting opinion targets from online reviews. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1754–1763, August.

Larry Manevitz and Malik Yousef. 2007. One-class document classification via neural networks. *Neurocomputing*, 70(7C9):1466–1481.

Mary M. Moya, Mark W. Koch, and Larry D. Hostetler. 1993. One-class classifier networks for target recognition applications. In *Proceedings world congress on neural networks*, pages 797–801.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 339–346.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1199–1204.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 105–112, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, John Bauer, Christopher D. Manning, and Ng Andrew Y. 2013. Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 455–465, Sofia, Bulgaria, August. Association for Computational Linguistics.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1533–1541, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen, and Jun Zhao. 2013. Mining opinion words and opinion targets in a two-stage framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1764–1773, Sofia, Bulgaria, August. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649, Sofia, Bulgaria, August. Association for Computational Linguistics.

Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1462–1470, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 43–50, New York, NY, USA. ACM.