Lifelong Learning CRF for Supervised Aspect Extraction

Lei Shu, Hu Xu, Bing Liu
Department of Computer Science
University of Illinois at Chicago
{lshu3, hxu48, liub}@uic.edu

Abstract

This paper makes a focused contribution to supervised aspect extraction. It shows that if the system has performed aspect extraction from many past domains and retained their results as knowledge, Conditional Random Fields (CRF) can leverage this knowledge in a lifelong learning manner to extract in a new domain markedly better than the traditional CRF without using this prior knowledge. The key innovation is that even after CRF training, the model can still improve its extraction with experiences in its applications.

1 Introduction

Aspect extraction is a key task of opinion mining (Liu, 2012). It extracts opinion targets from opinion text. For example, from the sentence "*The battery is good*", it aims to extract "battery", which is a product feature, also called an *aspect*.

Aspect extraction is commonly done using a supervised or an unsupervised approach. The unsupervised approach includes methods such as frequent pattern mining (Hu and Liu, 2004; Popescu and Etzioni, 2005; Zhu et al., 2009), syntactic rules-based extraction (Zhuang et al., 2006; Wang and Wang, 2008; Wu et al., 2009; Zhang et al., 2010; Qiu et al., 2011; Poria et al., 2014), topic modeling (Mei et al., 2007; Titov and McDonald, 2008; Li et al., 2010; Brody and Elhadad, 2010; Wang et al., 2010; Moghaddam and Ester, 2011; Mukherjee and Liu, 2012; Lin and He, 2009; Zhao et al., 2010; Jo and Oh, 2011; Fang and Huang, 2012; Wang et al., 2016), word alignment (Liu et al., 2013), label propagation (Zhou et al., 2013; Shu et al., 2016), and others (Zhao et al., 2015).

This paper focuses on the supervised approach (Jakob and Gurevych, 2010; Choi and Cardie, 2010; Mitchell et al., 2013) using Conditional Random Fields (CRF) (Lafferty et al., 2001). It shows that the results of CRF can be significantly improved by leveraging some prior knowledge automatically mined from the extraction results of previous domains, including domains without labeled data. The improvement is possible because although every product (domain) is different, there is a fair amount of aspects sharing across domains (Chen and Liu, 2014). For example, every review domain has the aspect *price* and reviews of many products have the aspect *battery life* or *screen*. Those shared aspects may not appear in the training data but appear in unlabeled data and the test data. We can exploit such sharing to help CRF perform much better.

Due to leveraging the knowledge gained from the past to help the new domain extraction, we are using the idea of *lifelong machine learning* (LML) (Chen and Liu, 2016; Thrun, 1998; Silver et al., 2013), which is a continuous learning paradigm that retains the knowledge learned in the past and uses it to help future learning and problem solving with possible adaptations.

The setting of the proposed approach L-CRF (*Lifelong CRF*) is as follows: A CRF model M has been trained with a labeled training review dataset. At a particular point in time, M has extracted aspects from data in n previous domains $D_1, \ldots, D_n$ (which are unlabeled) and the extracted sets of aspects are $A_1, \ldots, A_n$. Now, the system is faced with a new domain data D_{n+1}. M can leverage some *reliable prior knowledge* in $A_1, \ldots, A_n$ to make a better extraction from D_{n+1} than without leveraging this prior knowledge.

The key innovation of L-CRF is that even after supervised training, the model can still improve its extraction in testing or its applications with expe-

riences. Note that L-CRF is different from semi-supervised learning (Zhu, 2005) as the n previous (unlabeled) domain data used in extraction are not used or not available during model training.

There are prior LML works for aspect extraction (Chen et al., 2014; Liu et al., 2016), but they were all unsupervised methods. Supervised LML methods exist (Chen et al., 2015; Ruvolo and Eaton, 2013), but they are for classification rather than for sequence learning or labeling like CRF. A semi-supervised LML method is used in NELL(Mitchell et al., 2015), but it is heuristic pattern-based. It doesn't use sequence learning and is not for aspect extraction. LML is related to transfer learning and multi-task learning (Pan and Yang, 2010), but they are also quite different (see (Chen and Liu, 2016) for details).

To the best of our knowledge, this is the first paper that uses LML to help a supervised extraction method to markedly improve its results.

2 Conditional Random Fields

CRF learns from an observation sequence $\mathbf{x}$ to estimate a label sequence $\mathbf{y}$: $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a set of weights. Let l be the l-th position in the sequence. The core parts of CRF are a set of feature functions $\mathcal{F} = \{f_h(y_l, y_{l-1}, \mathbf{x}_l)\}_{h=1}^{H}$ and their corresponding weights $\boldsymbol{\theta} = \{\theta_h\}_{h=1}^{H}$.

Feature Functions: We use two types of feature functions (FF). One is *Label-Label (LL)* FF:

$$f_{ij}^{LL}(y_l, y_{l-1}) = \mathbb{1}\{y_l = i\}\mathbb{1}\{y_{l-1} = j\}, \forall i, j \in \mathcal{Y}, \quad (1)$$

where $\mathcal{Y}$ is the set of labels, and $\mathbb{1}\{\cdot\}$ an indicator function. The other is *Label-Word (LW)* FF:

$$f_{iv}^{LW}(y_l, \mathbf{x}_l) = \mathbb{1}\{y_l = i\}\mathbb{1}\{\mathbf{x}_l = v\}, \forall i \in \mathcal{Y}, \forall v \in \mathcal{V}, \quad (2)$$

where $\mathcal{V}$ is the vocabulary. This FF returns 1 when the l-th word is v and the l-th label is v's specific label i; otherwise 0. $\mathbf{x}_l$ is the current word, and is represented as a multi-dimensional vector. Each dimension in the vector is a feature of $\mathbf{x}_l$.

Following the previous work in (Jakob and Gurevych, 2010), we use the feature set {W, -1W, +1W, P, -1P, +1P, G}, where W is the word and P is its POS-tag, -1W is the previous word, -1P is its POS-tag, +1W is the next word, +1P is its POS-tag, and G is the generalized dependency feature.

Under the Label-Word FF type, we have two sub-types of FF: *Label-dimension* FF and *Label-G* FF. Label-dimension FF is for the first 6 features, and Label-G is for the G feature.

The *Label-dimension (Ld)* FF is defined as

$$f_{iv^d}^{Ld}(y_l, \mathbf{x}_l) = \mathbb{1}\{y_l = i\}\mathbb{1}\{\mathbf{x}_l^d = v^d\}, \forall i \in \mathcal{Y}, \forall v^d \in \mathcal{V}^d, \quad (3)$$

where $\mathcal{V}^d$ is the set of observed values in feature $d \in \{W, -1W, +1W, P, -1P, +1P\}$ and we call $\mathcal{V}^d$ feature d's feature values. Eq. (3) is a FF that returns 1 when $\mathbf{x}_l$'s feature d equals to the feature value v^d and the variable y_l (lth label) equals to the label value i; otherwise 0.

We describe G and its feature function next, which also holds the key to the proposed L-CRF.

3 General Dependency Feature (G)

Feature G uses generalized dependency relations. What is interesting about this feature is that it enables L-CRF to use past knowledge in its sequence prediction at the test time in order to perform much better. This will become clear shortly. This feature takes a *dependency pattern* as its value, which is generalized from dependency relations.

The general dependency feature (G) of the variable $\mathbf{x}_l$ takes a set of feature values $\mathcal{V}^G$. Each feature value v^G is a dependency pattern. The *Label-G (LG)* FF is defined as:

$$f_{iv^G}^{LG}(y_l, \mathbf{x}_l) = \mathbb{1}\{y_l = i\}\mathbb{1}\{\mathbf{x}_l^G = v^G\}, \forall i \in \mathcal{Y}, \forall v^G \in \mathcal{V}^G. \quad (4)$$

Such a FF returns 1 when the dependency feature of the variable $\mathbf{x}_l$ equals to a dependency pattern v^G and the variable y_l equals to the label value i.

3.1 Dependency Relation

Dependency relations have been shown useful in many sentiment analysis applications (Johansson and Moschitti, 2010; Jakob and Gurevych, 2010). A dependency relation [1] is a quintuple-tuple: *(type, gov, govpos, dep, deppos)*, where *type* is the type of the dependency relation, *gov* is the *governor word*, *govpos* is the POS tag of the governor word, *dep* is the *dependent word*, and *deppos* is the POS tag of the dependent word. The l-th word can either be the governor word or the dependent word in a dependency relation.

3.2 Dependency Pattern

We generalize dependency relations into *dependency patterns* using the following steps:

[1] We obtain dependency relations using Stanford CoreNLP: http://stanfordnlp.github.io/CoreNLP/.

Index	Word	Dependency Relations
1	The	{*(det, battery, 2, NN , The, 1, DT)* }
2	battery	{*(nsubj, great, 7, JJ , battery, 2, NN), (det, battery, 2, NN , The, 1, DT), (nmod, battery, 2, NN, camera, 5, NN)* }
3	of	{*(case, camera, 5, NN, of, 3, IN)* }
4	this	{*(det, camera, 5, NN, this, 4, DT)* }
5	camera	{*(case, camera, 5, NN, of, 3, IN), (det, camera, 5, NN, this, 4, DT), (nmod, battery, 2, NN, camera, 5, NN)* }
6	is	{*(cop, great, 7, JJ , is, 6, VBZ)* }
7	great	{*(root, ROOT, 0, VBZ, great, 7, JJ), (nsubj, great, 7, JJ , battery, 2, NN), (cop, great, 7, JJ , is, 6, VBZ)* }

Table 1: Dependency relations parsed from "The battery of this camera is great"

1. For each dependency relation, replace the current word (governor word or dependent word) and its POS tag with a wildcard since we already have the word (W) and the POS tag (P) features.

2. Replace the context word (the word other than the l-th word) in each dependency relation with a knowledge label to form a more general dependency pattern. Let the set of aspects annotated in the training data be K^t. If the context word in the dependency relation appears in K^t, we replace it with a knowledge label 'A' (aspect); otherwise 'O' (other).

For example, we work on the sentence "The battery of this camera is great." The dependency relations are given in Table 1. Assume the current word is "battery," and "camera" is annotated as an aspect. The original dependency relation between "camera" and "battery" produced by a parser is (nmod, battery, NN, camera, NN). Note that we do not use the word positions in the relations in Table 1. Since the current word's information (the word itself and its POS-tag) in the dependency relation is redundant, we replace it with a wild-card. The relation becomes (nmod, *, camera, NN). Secondly, since "camera" is in K^t, we replace "camera" with a general label 'A'. The final dependency pattern becomes (nmod,*, A, NN).

We now explain why dependency patterns can enable a CRF model to leverage the past knowledge. The key is the knowledge label 'A' above, which indicates a likely aspect. Recall that our problem setting is that when we need to extract from the new domain D_{n+1} using a trained CRF model M, we have already extracted from many previous domains $D_1, \ldots, D_n$ and retained their extracted sets of aspects $A_1, \ldots, A_n$. Then, we can mine reliable aspects from $A_1, \ldots, A_n$ and add them in K^t, which enables many knowledge labels in the dependency patterns of the new data A_{n+1} due to sharing of aspects across domains. This enriches the dependency pattern features, which consequently allows more aspects to be extracted from the new domain D_{n+1}.

Algorithm 1 Lifelong Extraction of L-CRF

$K_p \leftarrow \emptyset$
loop
  $F \leftarrow$ FeatureGeneration(D_{n+1}, K)
  $A_{n+1} \leftarrow$ Apply-CRF-Model(M, F)
  $S \leftarrow S \cup \{A_{n+1}\}$
  $K_{n+1} \leftarrow$ Frequent-Aspects-Mining(S, λ)
  if $K_p = K_{n+1}$ **then**
    break
  else
    $K \leftarrow K^t \cup K_{n+1}$
    $K_p \leftarrow K_{n+1}$
    $S \leftarrow S - \{A_{n+1}\}$
  end if
end loop

4 The Proposed L-CRF Algorithm

We now present the L-CRF algorithm. As the dependency patterns for the general dependency feature do not use any actual words and they can also use the prior knowledge, they are quite powerful for cross-domain extraction (the test domain is not used in training).

Let K be a set of *reliable aspects* mined from the aspects extracted in past domain datasets using the CRF model M. Note that we assume that M has already been trained using some labeled training data D^t. Initially, K is K^t (the set of all annotated aspects in the training data D^t). The more domains M has worked on, the more aspects it extracts, and the larger the set K gets. When faced with a new domain D_{n+1}, K allows the general dependency feature to generate more dependency patterns related to aspects due to more knowledge labels 'A' as we explained in the previous section. Consequently, CRF has more informed features to produce better extraction results.

L-CRF works in two phases: *training phase* and *lifelong extraction phase*. The training phase trains a CRF model M using the training data D^t, which is the same as normal CRF training, and

Domain	# Sent.	# Asp.	# non-asp. words
Computer	536	1173	7675
Camera	609	1640	9849
Router	509	1239	7264
Phone	497	980	7478
Speaker	510	1299	7546
DVD Player	506	928	7552
Mp3 Player	505	1180	7607

Table 2: Annotation details of the datasets

will not be discussed further. In the lifelong extraction phase, M is used to extract aspects from coming domains (M does not change and the domain data are unlabeled). All the results from the domains are retained in past aspect store S. At a particular time, it is assumed M has been applied to n past domains, and is now faced with the $n+1$ domain. L-CRF uses M and reliable aspects (denoted K_{n+1}) mined from S and K^t ($K = K^t \cup K_{n+1}$) to extract from D_{n+1}. Note that aspects K_t from the training data are considered always reliable as they are manually labeled, thus a subset of K. We cannot use all extracted aspects from past domains as reliable aspects due to many extraction errors. But those aspects that appear in multiple past domains are more likely to be correct. Thus K contains those frequent aspects in S. The lifelong extraction phase is in Algorithm 1.

Lifelong Extraction Phase: Algorithm 1 performs extraction on D_{n+1} iteratively.

1. It generates features (F) on the data D_{n+1} (line 3), and applies the CRF model M on F to produce a set of aspects A_{n+1} (line 4).

2. A_{n+1} is added to S, the past aspect store. From S, we mine a set of frequent aspects K_{n+1}. The frequency threshold is λ.

3. If K_{n+1} is the same as K_p from the previous iteration, the algorithm exits as no new aspects can be found. We use an iterative process because each extraction gives new results, which may increase the size of K, the reliable past aspects or past knowledge. The increased K may produce more dependency patterns, which can enable more extractions.

4. Else: some additional reliable aspects are found. M may extract additional aspects in the next iteration. Lines 10 and 11 update the two sets for the next iteration.

5 Experiments

We now evaluate the proposed L-CRF method and compare with baselines.

5.1 Evaluation Datasets

We use two types of data for our experiments. The first type consists of seven (7) annotated benchmark review datasets from 7 domains (types of products). Since they are annotated, they are used in training and testing. The first 4 datasets are from (Hu and Liu, 2004), which actually has 5 datasets from 4 domains. Since we are mainly interested in results at the domain level, we did not use one of the domain-repeated datasets. The last 3 datasets of three domains (products) are from (Liu et al., 2016). These datasets are used to make up our CRF training data D^t and test data D_{n+1}. The annotation details are given in Table 2.

The second type has 50 unlabeled review datasets from 50 domains or types of products (Chen and Liu, 2014). Each dataset has 1000 reviews. They are used as the past domain data, i.e., $D_1, \ldots, D_n$ ($n = 50$). Since they are not labeled, they cannot be used for training or testing.

5.2 Baseline Methods

We compare L-CRF with CRF. We will not compare with unsupervised methods, which have been shown improvable by lifelong learning (Chen et al., 2014; Liu et al., 2016). The frequency threshold λ in Algorithm 1 used in our experiment to judge which extracted aspects are considered reliable is empirically set to 2.

CRF: We use the linear chain CRF from [2]. Note that CRF uses all features including dependency features as the proposed L-CRF but does not employ the 50 domains unlabeled data used for lifelong learning

CRF+R: It treats the reliable aspect set K as a dictionary. It adds those reliable aspects in K that are not extracted by CRF but are in the test data to the final results. We want to see whether incorporating K into the CRF extraction through dependency patterns in L-CRF is actually needed.

We do not compare with domain adaptation or transfer learning because domain adaption basically uses the source domain labeled data to help learning in the target domain with few or no labeled data. Our 50 domains used in lifelong learning have no labels. So they cannot help in transfer

[2] https://github.com/huangzhengsjtu/pcrf/

Training	Testing	CRF			CRF+R			L-CRF		
		$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
Cross-Domain										
−Computer	Computer	86.6	51.4	64.5	23.2	90.4	37.0	82.2	62.7	71.1
−Camera	Camera	84.3	48.3	61.4	21.8	86.8	34.9	81.9	60.6	69.6
−Router	Router	86.3	48.3	61.9	24.8	92.6	39.2	82.8	60.8	70.1
−Phone	Phone	72.5	50.6	59.6	20.8	81.2	33.1	70.1	59.5	64.4
−Speaker	Speaker	87.3	60.6	71.6	22.4	91.2	35.9	84.5	71.5	77.4
−DVDplayer	DVDplayer	72.7	63.2	67.6	16.4	90.7	27.7	69.7	71.5	70.6
−Mp3player	Mp3player	87.5	49.4	63.2	20.6	91.9	33.7	84.1	60.7	70.5
	Average	82.5	53.1	64.3	21.4	89.3	34.5	79.3	63.9	70.5
In-Domain										
−Computer	−Computer	84.0	71.4	77.2	23.2	93.9	37.3	81.6	75.8	78.6
−Camera	−Camera	83.7	70.3	76.4	20.8	93.7	34.1	80.7	75.4	77.9
−Router	−Router	85.3	71.8	78.0	22.8	93.9	36.8	82.6	76.2	79.3
−Phone	−Phone	85.0	71.1	77.5	25.1	93.7	39.6	82.9	74.7	78.6
−Speaker	−Speaker	83.8	70.3	76.5	20.1	94.3	33.2	80.1	75.8	77.9
−DVDplayer	−DVDplayer	85.0	72.2	78.1	20.9	94.2	34.3	81.6	76.7	79.1
−Mp3player	−Mp3player	83.2	72.6	77.5	20.4	94.5	33.5	79.8	77.7	78.7
	Average	84.3	71.4	77.3	21.9	94.0	35.5	81.3	76.0	78.6

Table 3: Aspect extraction results in precision, recall and F_1 score: Cross-Domain and In-Domain (−X means all except domain X)

learning. Although in transfer learning, the target domain usually has a large quantity of unlabeled data, but the 50 domains are not used as the target domains in our experiments.

5.3 Experiment Setting

To compare the systems using the same training and test data, for each dataset we use 200 sentences for training and 200 sentences for testing to avoid bias towards any dataset or domain because we will combine multiple domain datasets for CRF training. We conducted both cross-domain and in-domain tests. Our problem setting is cross-domain. In-domain is used for completeness. In both cases, we assume that extraction has been done for the 50 domains.

Cross-domain experiments: We combine 6 labeled domain datasets for training (1200 sentences) and test on the 7th domain (not used in training). This gives us 7 *cross-domain* results. This set of tests is particularly interesting as it is desirable to have the trained model used in cross-domain situations to save manual labeling effort.

In-domain experiments: We train and test on the same 6 domains (1200 sentences for training and 1200 sentences for testing). This also gives us 7 *in-domain* results.

Evaluating Measures: We use the popular precision $\mathcal{P}$, recall $\mathcal{R}$, and $\mathcal{F}_1$-score.

5.4 Results and Analysis

All the experiment results are given in Table 3.

Cross-domain: Each −X in column 1 means that domain X is not used in training. X in column 2 means that domain X is used in testing. We can see that L-CRF is markedly better than CRF and CRF+R in $\mathcal{F}_1$. CRF+R is very poor due to poor precisions, which shows treating the reliable aspects set K as a dictionary isn't a good idea.

In-domain: −X in training and test columns means that the other 6 domains are used in both training and testing (thus in-domain). We again see that L-CRF is consistently better than CRF and CRF+R in $\mathcal{F}_1$. The amount of gain is smaller. This is expected because most aspects appeared in training probably also appear in the test data as they are reviews from the same 6 products.

6 Conclusion

This paper proposed a lifelong learning method to enable CRF to leverage the knowledge gained from extraction results of previous domains (unlabeled) to improve its extraction. Experimental results showed the effectiveness of L-CRF. The current approach does not change the CRF model itself. In our future work, we plan to modify CRF so that it can consider previous extraction results as well as the knowledge in previous CRF models.

Acknowledgments

This work was supported in part by grants from National Science Foundation (NSF) under grant no. IIS-1407927 and IIS-1650900.

References

Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *NAACL '10*. pages 804–812.

Zhiyuan Chen and Bing Liu. 2014. Topic modeling using topics from many domains, lifelong learning and big data. In *ICML '14*. pages 703–711.

Zhiyuan Chen and Bing Liu. 2016. *Lifelong Machine Learning*. Morgan & Claypool Publishers.

Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong learning for sentiment classification. *Volume 2: Short Papers* page 750.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *ACL '14*. pages 347–358.

Yejin Choi and Claire Cardie. 2010. Hierarchical sequential learning for extracting opinions and their attributes. In *ACL '10*. pages 269–274.

Lei Fang and Minlie Huang. 2012. Fine granular aspect analysis using latent structural models. In *ACL '12*. pages 333–337.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD '04*. pages 168–177.

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

Yohan Jo and Alice H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *WSDM '11*. pages 815–824.

Richard Johansson and Alessandro Moschitti. 2010. Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. pages 67–76.

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

Fangtao Li, Minlie Huang, and Xiaoyan Zhu. 2010. Sentiment analysis with global topics and local dependency. In *AAAI '10*. pages 1371–1376.

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

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

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

Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving opinion aspect extraction using semantic similarity and aspect associations. In *Thirtieth AAAI Conference on Artificial Intelligence*.

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

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

T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, M Gardner, B Kisiel, J Krishnamurthy, N Lao, K Mazaitis, T Mohamed, N Nakashole, E Platanios, A Ritter, M Samadi, B Settles, R Wang, D Wijaya, A Gupta, X Chen, A Saparov, M Greaves, and J Welling. 2015. Never-ending learning. In *AAAI'2015*.

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

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *ACL '12*. volume 1, pages 339–348.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22(10):1345–1359.

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

Soujanya Poria, Erik Cambria, Lun-Wei Ku, Chen Gui, and Alexander Gelbukh. 2014. A rule-based approach to aspect extraction from product reviews. In *SocialNLP '14*. pages 28–37.

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

Paul Ruvolo and Eric Eaton. 2013. Ella: An efficient lifelong learning algorithm. *ICML (1)* 28:507–515.

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

Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong machine learning systems: Beyond learning algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*. Citeseer, pages 49–55.

Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*, Springer, pages 181–209.

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

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

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *KDD '10*. pages 783–792.

Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016. Mining aspect-specic opinion using a holistic lifelong topic model. In *WWW '16*.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *EMNLP '09*. pages 1533–1541.

Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents. In *COLING '10: Posters*. pages 1462–1470.

Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *EMNLP '10*. pages 56–65.

Yanyan Zhao, Bing Qin, and Ting Liu. 2015. Creating a fine-grained corpus for chinese sentiment analysis. *IEEE Intelligent Systems* 30(1):36–43.

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

Jingbo Zhu, Huizhen Wang, Benjamin K. Tsou, and Muhua Zhu. 2009. Multi-aspect opinion polling from textual reviews. In *CIKM '09*. pages 1799–1802.

Xiaojin Zhu. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

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