Detecting Domain Polarity-Changes of Words in a Sentiment Lexicon

Shuai Wang[†*], **Guangyi Lv**[§], **Sahisnu Mazumder**[‡], **Bing Liu**[‡]
[‡]Department of Computer Science, University of Illinois at Chicago, USA
[§]University of Science and Technology of China, Hefei, China
[†]Amazon AI
shuaiwanghk@gmail.com, gylv@mail.ustc.edu.cn
sahisnumazumder@gmail.com, liub@uic.edu

Abstract

Sentiment lexicons are instrumental for sentiment analysis. One can use a set of sentiment words provided in a sentiment lexicon and a lexicon-based classifier to perform sentiment analysis. One major issue with this approach is that many sentiment words (from the lexicon) are domain dependent. That is, they may be positive in some domains but negative in some others. We refer to this problem as *domain polarity-changes of words* from a sentiment lexicon. Detecting such words and correcting their sentiment for an application domain is very important. In this paper, we propose a graph-based technique to tackle this problem. Experimental results show its effectiveness on multiple datasets from different domains.

1 Introduction

Sentiment words, also called opinion/polar words, are words that convey positive or negative sentiments (Pang and Lee, 2008). Such sentiment-bearing words are usually pre-compiled as word lists in a sentiment lexicon, which is instrumental as well as an important linguistic resource to sentiment analysis (Liu, 2012). So far, numerous studies about how to construct lexicons have been reported, which will be discussed in Section 2.

Despite the fact that there is extensive research on lexicon construction, limited work has been done to solve the problem of identifying and handling sentiment words in a given/constructed lexicon that have domain-dependent polarities. In real-life applications, there are almost always some sentiment words that express sentiments different from their default polarities provided in a general-purpose sentiment lexicon. For example, in the lexicon compiled by Hu and Liu (2004), the word "crush" is associated with a negative sentiment, but it actually shows a positive opinion in domain *blender* because "crush" indicates that a blender works well, e.g., in the sentence "it does crush the ice!". We call this problem *domain polarity-change* of a word in a sentiment lexicon.

The polarity change of words plays a crucial role in sentiment classification. As we will see in the experiment section, without identifying and correcting such domain dependent sentiment words, the performance of sentiment classification could be much poorer. Although some researchers have studied the domain-specific sentiment problem with lexicons, their focuses are quite different and their approaches are not suitable for our task. We will discuss them further in the following sections.

Regarding sentiment classification, it is important to note that our work mainly aims to help *lexicon-based approaches* (Taboada et al., 2011). It does not directly help machine-learning (ML) or supervised learning approaches (Zhang et al., 2018) because the domain-dependent polarities of words are already reflected in the manually labeled training data. Notice that for those ML approaches, the manual annotation for each application domain is required, which is a time-consuming and labor-intensive task, and is thus hard to scale up. In many real-world scenarios, lexicon-based approaches are useful and could be a better alternative (Liu, 2012).

However, to effectively apply a sentiment lexicon to an application domain, the domain-polarity change problem discussed above needs to be addressed. To this end, we propose a graph-based approach named Domain-specific Sentiment Graph (DSG) in Section 4. It works with three steps: (domain) sentiment word collection, (domain) sentiment correlation extraction, and graph construction and inference. Experimental results show its effectiveness in detecting domain polarity-changed words on multiple datasets. We will also see with the detection of those words, a huge performance gain can be achieved in sentiment classification.

[*] Work done while at University of Illinois at Chicago

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3657–3668
August 1–6, 2021. ©2021 Association for Computational Linguistics

2 Related Work

This work concerns domain polarity changes of words in lexicons. So we first discuss the works related to sentiment lexicons, and then domain sentiment, and finally domain sentiment with lexicons.

Extensive studies have been done for sentiment lexicons and the majority of them focus on lexicon construction. These approaches can be generally categorized as dictionary-based and corpus-based.

Dictionary-based approaches first used some sentiment seed words to bootstrap based on the synonym and antonym structure of a dictionary (Hu and Liu, 2004; Valitutti et al., 2004). Later on, more sophisticated methods were proposed (Kim and Hovy, 2004; Esuli and Sebastiani, 2005; Takamura et al., 2007; Blair-Goldensohn et al., 2008; Rao and Ravichandran, 2009; Mohammad et al., 2009; Hassan and Radev, 2010; Dragut et al., 2010; Xu et al., 2010; Peng and Park, 2011; Gatti and Guerini, 2012; San Vicente et al., 2014). Corpus-based approaches build lexicons by discovering sentiment words in a large corpus. The first idea is to exploit some coordinating conjunctions (Hatzivassiloglou and McKeown, 1997; Hassan and Radev, 2010). Kanayama and Nasukawa (2006) extended this approach by introducing inter-sentential sentiment consistency. Other related work includes (Kamps et al., 2004; Kaji and Kitsuregawa, 2006; Wang et al., 2017). The second idea is to use syntactic relations between opinion and aspect words (Zhuang et al., 2006; Wang and Wang, 2008; Qiu et al., 2011; Volkova et al., 2013). The third idea is to use word co-occurrences for lexicon induction (Turney and Littman, 2003; Igo and Riloff, 2009; Velikovich et al., 2010; Yang et al., 2014; Rothe et al., 2016).

However, our work is very different as we focus on detecting domain dependent sentiment words in a given general-purpose sentiment lexicon.

Also related is the existing research about domain and context dependent sentiment. First, despite the fact that several researchers have studied context dependent sentiment words, which are based on sentences and topic/aspect context (Wilson et al., 2005; Ding et al., 2008; Choi and Cardie, 2008; Wu and Wen, 2010; Jijkoun et al., 2010; Lu et al., 2011; Zhao et al., 2012; Kessler and Schütze, 2012; Teng et al., 2016; Wang et al., 2016, 2018a,b; Li et al., 2018a), our work is based on domains. Second, while the studies on transfer learning or domain adaptation for sentiment analysis deal with domain information (Bhatt et al., 2015; Yu and Jiang, 2016; Li et al., 2018b), our work does not lie in this direction. We do not have any source domain and our goal is not to transfer domain knowledge to another domain. Third, most importantly, the above works are either irrelevant to lexicons or not for detecting the sentiment discrepancy between a lexicon and the application domain.

Our work is most related to the following studies that involve both sentiment lexicons and domain sentiment problems. Choi and Cardie (2009) adapted the word-level polarities of a general-purpose sentiment lexicon to a particular domain by utilizing the expression-level polarities in that domain. However, their work targeted at reasoning the sentiment polarities of multi-word expressions. It does not detect or revise the sentiment polarities of individual words in the lexicon for a particular domain, and hence, cannot solve our problem. Du et al. (2010) studied the problem of adapting the sentiment lexicon from one domain to another domain. It further assumes that the source domain has a set of sentiment-labeled reviews. Their technique is therefore more about transfer learning and their learning settings differ from ours intrinsically. (Ortiz, 2017) designed a sentiment analysis application that allows plugin lexicons (if users can provide them) to help predict domain sentiment. It neither detects nor corrects domain polarity-changed words. Perhaps, the most related work is (Hamilton et al., 2016), which uses seed lexicon words, word embeddings, and random walk to generate a domain-specific lexicon. However, their model is for lexicon construction in essence, by (its capability of) functioning on a domain-oriented corpus. It does not aim to detect/change the sentiment polarity from a given lexicon. It is thus not directly applicable to our task. To make it workable for our task, we design a two-step approach, which will be detailed in the experiment section (Section 5).

To the best of our knowledge, this is the first study to detect domain polarity-changes of words in a sentiment lexicon. We will further discuss why this task is important and useful in Section 5.7.

3 Problem Definition

Given a general-purpose sentiment lexicon L (which contains sentiment words and their default polarities) and an application domain review corpus D, the goal (or the task of detecting domain polarity changes of words) is to identify a subset of words in L that carry different sentiment polarities in that domain (different from their default polarities), which we call *domain polarity-changed (lexical*[1]*) words* and denote them as C ($C \subseteq L$).

4 Proposed Solution

To tackle the above problem, we propose a graph-based learning approach named Domain-specific Sentiment Graph (DSG). It works with three major steps: (1) (domain) sentiment word collection, (2) (domain) sentiment correlation extraction, and (3) graph construction and inference.

Specifically, it first collects a set of mentioned sentiment words S ($L \subseteq S$) in the domain corpus D. It then mines multiple types of relationships among sentiment words in S, which are denoted as a relationship set R. The relationships are identified based on different types of linguistic connectivity. Next, it builds a probabilistic graph with each node representing a sentiment word in S and each edge representing a relation (from R) between two words. An inference method is then applied to re-estimate the domain-specific polarities (or beliefs) of sentiment words. With the re-estimated beliefs obtained in the application domain, those sentiment words with changed polarities can be detected, by measuring the sentiment shift of a lexical word between its induced (in-domain) sentiment belief and its original (lexicon-based) polarity.

In this learning manner, the proposed approach requires no prior knowledge or annotated data for a particular domain. It is thus applicable to multiple/different domains. Intuitively, this approach works based on two assumptions:

Assumption 1: Sentiment Consistency (Abelson, 1983; Liu, 2012): a sentiment expression tends to be sentimentally coherent with its context. Notice that sentiment consistency can be reflected in multiple types of conjunction like "and", "or", etc., which will be explained in Section 4.2. In fact, this assumption is common in sentiment analysis and has been used in many studies (Kanayama and Nasukawa, 2006; Hassan and Radev, 2010)

Assumption 2: The number of domain polarity-changed lexical words[1] is much smaller than the number of those (words) whose polarities do not change. This assumption ensures that we can rely on the general-purpose lexicon itself for detection[1]. In other words, the real polarity of a sentiment word in a certain domain can be distilled by its connections with other (mostly polarity-unchanged) words whose polarities are known from the lexicon.

4.1 Sentiment Word Collection

As the first step, DSG collects sentiment words in an application domain corpus, including the sentiment words not present in a lexicon. Specifically, we consider three types of (likely) sentiment words: (1) The word appears in a given lexicon. (2) The word is an adjective in the corpus. (3) The word is an adverb in the corpus and has an adjective form.

We simply accept all lexical words and adjective words as (likely) sentiment words, which does not cause serious problems in our experiments and they were also commonly used in the literature (Liu, 2012). However, we impose constraints on selecting adverbs. While adverbs like "quickly" and "nicely" do express sentiment, some others like "very" and "often" may not function the same. We thus use the adverbs having adjective forms only.

Notice that in the above setting, the sentiment words not present in the lexicon are also collected due to two reasons: first, they are useful for building connection among other lexical words for inference purposes. Suppose that "quick" is a sentiment word (found because it is an adjective) and it is not in the given lexicon. Given its sentiment correlations with other words like "efficient and quick" and "quiet and quick", it can make a path between "efficient" and "quiet" in the graph. Second, in each domain there exist a number of sentiment words uncovered by the given lexicon. Their inferred polarities can also benefit the graph reasoning process, though those words are not the focus of detection in this study (we aim at detecting the polarity change of lexical words, i.e., words from a given lexicon). For instance, if the non-lexical word "quick" is identified as expressing positive sentiment (in the application domain) , "efficient" and "quiet" are more likely to be positive as well, given their in-domain sentiment correlations. We follow (Das and Chen, 2007; Pang and Lee, 2008) to handle the negation problem, where a negation word phrase like "not bad" will be treated as a single word like "not_bad" and its sentiment polarity will be reversed accordingly. Finally, all extracted words are modeled as nodes in the graph.

4.2 Sentiment Correlation Extraction

This step is to extract multiple types of conjunction relationship among sentiment words, which

[1] In this paper, we call the words in a given lexicon *lexical words* for short. The term *detection* will generally stand for the detection of domain polarity-changes of (lexical) words.

we refer to as *sentiment correlation*[2]. The key idea here is to use the sentiment consistency (Abelson, 1983; Liu, 2012) (see Assumption 1) for relationship construction among the collected sentiment words from the above step. Specifically, in an application domain, five types of sentiment correlation are considered, each of which is presented in a triple format, denoted as ($word_1$, correlation type, $word_2$). They will be further used in the graph inference process (discussed in the next sub-section). Their definitions are shown in Table 1.

In each sentence, when a specific type of relationship between two (collected) sentiment words is found, a triple is created. For instance, in the sentence “it is efficient and quiet”, a triple (efficient, AND, quiet) will be generated. The extraction of OR sentiment correlation is similar to AND. Likewise, a specific BUT triple (powerful, BUT, noisy) will be extracted from the sentence “it is a powerful but noisy machine”. The extraction of ALT (abbreviation for although) is similar to BUT. NB means two neighboring sentiment words occur in a sentence, like “reasonably good”.

Notice that while five types of relationships are jointly considered, they are associated with different agreement levels (parameterized in the graphical model discussed below). Here the agreement level measures how likely the sentiment polarities of two connecting words are the same. Intuitively, we believe AND gives the highest-level agreement. For instance, “bad and harmful” is very common but “good and harmful” is much unlikely. It is also an intuitive belief that BUT indicates the strongest disagreement between two sentiment words. Note that we only consider pairwise relationships between sentiment words in this study, which already achieve reasonably good results, as we will see.

4.3 Graph Construction and Inference

This subsection presents how our proposed domain-specific sentiment graph is used for detecting polarity-changed words after the above two steps.

4.3.1 Constructing Markov Random Field

Markov Random Fields (MRFs) are a class of probabilistic graphical models that can deal with the inference problems with uncertainty in observed data. An MRF works on an undirected graph G, which is constructed by a set of vertexes/nodes V and edges/links E and denoted as $G = (V, E)$. In the graph G, each node $v_i \in V$ denotes a random variable and each edge $(v_i, v_j) \in E$ denotes a statistical dependency between node v_i and node v_j. Formally, $\psi_i(v_i)$ and $\psi_{ij}(v_i, v_j)$ are defined as two types of potential functions for encoding the observation (or *prior*) of a node and the dependency between two nodes, also called *node potential* and *edge potential* respectively. An MRF thus can model a joint distribution for a set of random variables and its aim is to infer the marginal distribution for all $v_i \in V$. With an inference method used, the estimation of the marginal distribution of a node can be obtained, which is also called *belief*.

The reason why we formulate our domain-specific sentiment graph as an MRF is three-fold: (1) The sentiment correlation between two words is a mutual relationship, as one word w_a can provide useful sentiment information to the other word w_b and vice versa, which can be properly formulated in an undirected graph. (2) From a probabilistic perspective, the polarity changes of sentiment words can be naturally understood as the belief estimation problem. That is, on one hand, we have an initial belief about the polarity of a lexical word (known from the lexicon, like the word “cold” is generally negative), which is essentially the prior. On the other hand, our goal is to infer the real polarity of a word in a specific application domain, which is reflected in its final estimated belief (like “cold” is positive in the domain *fridge*). Concretely, the polarity of a sentiment word is modeled as a 2-dimensional vector, standing for the probability distribution of positive and negative polarities, e.g., $[0.9, 0.1]$ indicates that a word is very likely to express a positive sentiment in an application domain. We can further use p as the parameter to simplify the representation as $[p, 1-p]$. (3) Recall that multiple types of sentiment correlation are used and treated differently in our proposed approach. These *typed sentiment correlations* can be encoded in the MRF model (will be further illustrated below).

4.3.2 Inference over Typed Correlation

As discussed above, the inference task in MRF is to compute the marginal distribution (or posterior probability) of each node given the node prior and edge potentials. Efficient algorithms for exact inference like Belief Propagation (Pearl, 1982) are available for certain graph topologies, but for general graphs involving cycles the exact inference is computationally intractable. Approximate inference is

[2] The term *sentiment correlation* used in this paper denotes the correlation between two sentiment words in a domain, which may not have to be the same as used in other studies.

Name	Correlation	Example	Representation	Agreement Level
AND	connecting with "and"	"it is efficient and quiet"	(efficient, AND, quiet)	Strongly Agree
OR	connecting with "or"	"everything as expected or better"	(expected, OR, better)	Agree
NB	neighboring words	"a reasonably quiet fridge"	(reasonably, NB, quiet)	Weakly Agree
ALT	although, though	"too noisy, though it is efficient"	(noisy, ALT, efficient)	Disagree
BUT	but, however	"it is a powerful but noisy machine"	(powerful, BUT, noisy)	Strongly Disagree

Table 1: Five types of sentiment correlation.

thus needed. Loopy Belief Propagation (Murphy et al., 1999) is such an approximate solution using iterative message passing. A message from node i to node j is based on all message from other nodes to node i except node j itself. It works as:

$$m_{i\to j}(x_j) = z \sum_{x_i\in S} \psi_{i,j}(x_i,x_j)\psi_i(x_i) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \tag{1}$$

where S denotes the possible states of a node, i.e., being a sentiment word with positive or negative polarity. x_j indicates that node j is in a certain state. $N(i)$ denotes the neighborhood of i, i.e., the other nodes linking with node i. $m_{i\to j}$ is known as the message passing from node i to node j. z is the normalization constant that makes message $m_{i\to j}(x_j)$ proportional to the likelihood of the node j being in state x_j, given the evidence from i in its all possible states. After iterative message passing, the final belief $b_i(x_i)$ is estimated as:

$$b_i(x_i) = z'\psi_i(x_i) \prod_{k\in N(i)} m_{k\to i}(x_i) \tag{2}$$

where z' is a normalization term that makes $\sum_{x_i} b_i(x_i) = 1$. In this case, $b_i(x_i)$ can be viewed as the posterior probability of a sentiment word being with positive or negative polarity.

However, notice that in the above setting, each edge is not distinguishable in terms of its type of sentiment correlation. In other words, each type of possible connections between words is treated intrinsically the same, which does not meet our modeling requirements. In order to encode the typed sentiment correlation as defined in previous sections, we propose to replace the Eq. 1 with:

$$\begin{aligned} m_{i\to j}(x_j) = & z \sum_{x_i\in S} \sum_{r_{i,j}\in R} \psi_{i,j,r_{i,j}}(x_i,x_j) \\ & \psi_i(x_i) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i), \end{aligned} \tag{3}$$

where $r_{i,j} \in R$ indicates the specific type of sentiment correlation between node i and node j, which can be any type like AND or NB as defined in Section 4.2. $\psi_{i,j,r_{i,j}}(x_i,x_j)$ thus becomes an edge potential function related to its sentiment correlation type. Each type of a correlation is parameterized as a (state) transition matrix shown in Table 2. The five types of sentiment correlation therefore result in five such tables/matrices but with different ϵ being set. For example, ϵ with AND can be set to 0.3 as it indicates the highest agreement level, while the one with NB can be set to 0.1 as it is regarded as weakly agreement. For BUT, ϵ can be set to -0.3 as it shows strong disagreement.

State	Positive	Negative
Positive	$0.5+\epsilon$	$0.5-\epsilon$
Negative	$0.5-\epsilon$	$0.5+\epsilon$

Table 2: Transition/Propagation matrix.

For each word, with its estimated beliefs $[b^+, b^-]$ obtained in the application domain, its *polarity change score (pcs)* is defined as:

$$pcs = I(l=+)b^- + I(l=-)b^+ \tag{4}$$

where l denotes the original sentiment polarity of a lexical word (known from the given lexicon), and $I(.)$ is an indicator function. According to the resulting scores of all words, a word list ranked by *pcs* is used to identify the most likely polarity-changed sentiment words, e.g., one can select the top n words or set a threshold for word extraction. In this way, the sentiment words with changed polarities in an application domain can be detected.

4.4 An Illustrating Example for DSG

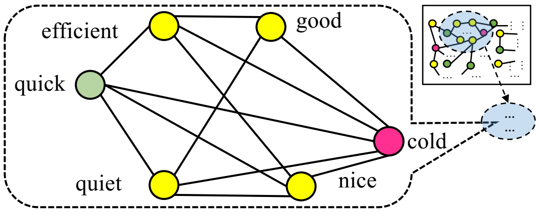

Figure 1: DSG Example. The nodes in pink (or darkest in grayscale) like "cold" represent the polarity-changed words in a domain, while nodes in yellow (or lightest) like "nice" indicate polarity-unchanged words, and nodes in green like "quick" denote non-lexical words.

Figure 1 provides an example to illustrate how DSG works in domain *fridge*. It shows a subgraph with several nodes, such as "quick", "nice", and "cold", representing the words extracted from step 1. The links between them represent their sentiment correlation as discussed in step 2. With the inference performed in step 3, "cold" is detected as a polarity-changed word (e.g., with a high pcs value 0.9 inferred), where its sentiment information is estimated by the (sentiment) message passing from other nodes in the graph. Specifically, in the *fridge* domain, through sentences like "looks nice and quiet", "it is quiet and cold", and "nice to have a quick cold beer or soda" (from the fridge), positive sentiment is directly or indirectly propagated from the words/nodes "nice", "quiet", "quick" to "cold" with their (typed) connections. Therefore, while "cold" was assigned a negative prior $[0.0, 1.0]$ (by default negative in the given lexicon), through the in-domain sentiment inference its final estimated belief becomes $[0.9, 0.1]$, resulting a high pcs score ($psc = 0.9$ by Eq. 4). "cold" is thus detected.

5 Experiments

We conducted experiments on multiple datasets with several candidate solutions. Here we first compare their performance on the word detection task. Sentiment classification will then be another task to evaluate their effect on polarity correction.

5.1 Experimental Setup

Dataset. Four real-world datasets from different domains/products were used, namely, *fridge*, *blenders*, *washer*, and *movie*. The first three datasets contain review sentences of these products. The fourth dataset (*movie*) consists of tweets from Twitter discussing movies. All these datasets are collected by us. The first dataset contains around 36,000 (36k) sentences, the second 16,000 (16k) sentences, and the rest datasets 10,000 (10k). Their sizes can be viewed as large, medium, and small. Such product diversity and variable size settings help evaluate the generality of each solution. Note that only the text is used by all candidate models.

In addition, two other datasets from domains *drill* and *vacuum cleaner* are used as development sets for hyper-parameter selections. *drill* contains 76k and *vacuum cleaner* contains 9k sentences.

Sentiment Lexicon. We used a general-purpose sentiment lexicon from (Hu and Liu, 2004), which contains 2,006 positive and 4,783 negative lexical words. A candidate model will find polarity-changed words from them for each domain/dataset.

Parameter Settings. The hyper-parameters of state priors and the (typed) transition matrices in DSG are shown in Table 3 and 4. They were empirically set based on the word detection performance on the two development datasets. We found this parameter setting works generally well on both datasets, while they are from different domains and with different data sizes. The following reported results for evaluations are based on this setting and as we will see, it already produces quite good results.

Prior	Positive	Non-lexical Words	Negative
p *in* $\psi_i(v_i)$	0.70	0.50	0.30

Table 3: Parameters of state prior.

Types	AND	OR	NB	ALT	BUT
ϵ **in** $\psi_{i,j}(v_i, v_j)$	0.20	0.10	0.05	-0.10	-0.20

Table 4: Parameters of typed transition matrix.

5.2 Lexicon-based Sentiment Classifier

Our evaluations include lexicon-based sentiment classification. We briefly illustrate how a lexicon-based sentiment classifier (called *classifier* for short) works here. Clearly, it works with a lexicon, from which each word is associated with a sentiment score (e.g., -1/+1). The classifier then calculates the sentiment score s of each sentence t by summing the score of each word. We follow the lexicon-based classifier design in (Taboada et al., 2011), incorporating sentiment negation and intensity. The sentence sentiment score s is calculated:

$$s = \sum_{w \in t} \mathit{negation}(w) * \mathit{intensity}(w) * \mathit{polarity}(w) \quad (5)$$

where w denotes a word in sentence t. A sentence is classified as positive if the resulting score s is greater than or equal to zero, otherwise negative.

Different lexicons working with this classifier will generate different results. That is, even if their lexical words are the same, the associated sentiment score of a lexical word could vary and Eq. 5 will thus make different predictions. This is how we can utilize the classifier to verify the effect of the word detection, because the classifier will perform differently using the original lexicon and the modified lexicon, whose results can be compared in a *before-and-after* manner. Here the *modified lexicon* means the sentiment polarities of (detected) lexical words are corrected from the original lexicon. For example, "crush" is associated with negative sentiment (-1) in the original lexicon, but it could be associated with positive sentiment (+1) in the modified lexicon (if detected). So the sentence-level sentiment scores will vary accordingly, e.g., “the machine does crush ice!” will be predicted as a positive sentence with the modified lexicon.

5.3 Candidate Models

Original Lexicon (OL): This is a baseline for sentiment classification evaluations only (Section 5.5), which uses the classifier with the original lexicon.
Domain-specific Sentiment Graph (DSG): This is our proposed model introduced in the previous sections. The following two models and DSG will be used for both word detection and classification.
Lexicon-Classifier Inconsistency (LCI): This is a heuristic solution to detecting the polarity-changed sentiment words. It relies on the inconsistency between the sentiment of a lexical word (obtained from the original lexicon) and the sentiment of the sentences containing the word (induced by the classifier). Concretely, it first calculates the sentiment polarities of all sentences using a classifier with the original lexicon. With the polarities of sentences known, it computes an inconsistency ratio for each lexical word. The inconsistency ratio is the ratio of (a) to (b), where (a) is the number of a word appearing in the positive/negative sentences but the word itself is negative/positive, and (b) is the number of all sentences covering that word. Finally, it ranks all lexical words based on their ratio values to produce a list of likely polarity-changed words.
SentProp (SP): SentProp (Hamilton et al., 2016) is a lexicon construction algorithm concerning domain sentiment, which is the most related work to ours. As discussed in Section 2, it is not directly applicable to the detection task. But since it can generate a list of domain-specific sentiment words and those words are associated with positive/negative scores (estimated by SentProp, which can be treated as the in-domain beliefs like DSG), we can design a two-step approach to achieve our goal. First, we download[3] and run the SentProp system to learn the domain-specific lexicon for each domain. Second, we calculate the polarity change scores for all lexicon words like DSG based on the learned domain-specific sentiment scores and the original polarities from the lexicon using Eq. 4. Similar to DSG, it produces a list of words ranked by the polarity change scores (pcs). For its parameter selection, we tried both the system default, following the code instruction and the original paper, and parameter fine-tuning based on the performance on two development sets (same as DSG), so as to achieve its best performance to report.

5.4 Correct Detection of Words

As each candidate model generates a list of words ranked by polarity-change scores, those top-ranked ones are the most likely polarity-changed words and can be used as the detected words. For evaluation, the top-n words from each model are inspected and the number of correct (word) detection is counted, which is denoted as **#C@n** in Table 5.

Specifically, two domain experts who are familiar with the domain sentiments identify and annotate the correct polarity-changed words from the top-20 shortlisted candidate words generated by each model. For each candidate word, we sampled numbers of sentences containing that word for the domain experts to judge. A candidate word needs to be agreed upon by both of them to be correct. Here the Cohen’s Kappa agreement score is 0.817.

Model	#C@n	fridge	blender	washer	movie
DSG	#C@5	5	5	4	5
	#C@10	9	10	6	9
	#C@20	12	15	12	15
LCI	#C@5	3	3	3	1
	#C@10	5	3	5	4
	#C@20	5	7	7	9
SP	#C@5	1	0	1	1
	#C@10	2	0	2	4
	#C@20	3	3	3	6

Table 5: Detection of polarity-changed words.

Evaluation results are reported in Table 5, where we can see that DSG achieves outstanding results consistently. LCI also does a decent job, while SP does not perform well on this task.

Next, we will evaluate the impact of such detection from their top 20 words, and the following sub-sections are based on their correctly detected words to give further analyses.

5.5 Sentiment Classification

After the detection of polarity-changed words, we conduct classification tasks on the sentences containing at least one word from the detected words of all models. Because the classification results on the sentences that without containing any detected word would not be affected (same prediction results using either the original or modified lexicon).

For evaluation, we sampled and labeled 925 (around 1k) sentences, from all sentences that could be affected. We used a stratified sampling strategy

[3] https://nlp.stanford.edu/projects/socialsent/

and set the minimum number of sentences contained by each word, to make sure each detected word is considered. The numbers of labeled sentences for the four domains are 232, 214, 174, and 305. The Cohen's Kappa agreement score is 0.788.

In regard to the lexicon-based classification, for DSG and LCI, the modified lexicon for each domain is based on the correction of the original lexicon (OL) on that domain. For SP, its self-generated lexicon is used with its inferred sentiment scores.

Model	fridge	blender	washer	movie	AVG
DSG	**74.56%**	**80.84%**	**77.01%**	84.91%	**79.33%**
SP	68.10%	78.97%	66.67%	**87.87%**	75.40 %
LCI	61.63%	68.22%	62.64%	62.95%	63.86%
OL	61.20%	65.42%	62.06%	56.72%	61.35%

Table 6: Sentiment classification accuracy.

Table 6 reports the classification accuracy, from which we have the following observations:

1. Compared to the baseline using the original lexicon (OL), DSG greatly improves the accuracy by 17.98% on average (AVG). We can see the usefulness of detecting polarity change of lexical words for sentiment classification.
2. SP also produces very good results. The reason is, as a lexicon-generation approach (essentially), SP itself creates a bigger lexicon for each domain (around 2 times bigger than OL), including additional sentiment words outside the original lexicon. In other words, discovering more sentiment words (with more sentiment clues provided) could also help better classification. Note that this does not contradict the importance of detecting polarity-change words, as they are two different aspects, which will be discussed in Section 5.7.
3. LCI outperforms OL but its performance gain is small. The reason is, though LCI detects polarity changed words decently, its detected words affect a much smaller number of sentences compared to the ones from DSG and SP, i.e., the words LCI detects are rarer and less frequent, with fewer sentences being affected.

5.6 Example Results

Here we show some example results. Table 7 lists the top polarity-changed words detected by DSG on domains *blender* and *movie*; we will explain how they benefit the domain sentiment analysis with example sentences. Below, an underlined word in an example sentence indicates its polarity has changed in a certain domain. In domain *blender*, the words like "crushing" and "breaks" express positive sentiments in "good for crushing ice" and "this breaks down frozen fruit", while they are provided as negative words in the original lexicon. With the detection and correction of those words, their domain-specific sentiments are identified correctly as positive and thus make the classification correct (through the modified lexicon provided by DSG). In domain *movie*, we found that many negative lexical words are in fact commonly used by users to show their love for a movie. For instance, "damn, I wanna watch this movie so bad" and "this movie was insanely brilliant". Similarly, for other domains, we also found "it keeps foods cold and frozen" in *fridge*, "you can also delay your cycle" in *washer*, and "it sucked very well" in *vacuum cleaner*. DSG also detects those words.

blender	movie
frozen shake cheap crushing	despicable fucking complex
breaks crush lost destroy	insanely damn *sad* crazy
loose crushed breaking	intense funny freaking *creepy*
broken lose grind *clog bulky*	bloody crap shit bad *retarded*
kills wobbled bitter shocked	*mad* insane terribly *eccentric*

Table 7: Example results on *blender* and *movie*. Incorrectly detected words are italicized and marked in red.

5.7 Further Analysis and Discussion

We aim to answer two questions here. Q1: What is the key difference between using SP and DSG? Q2: More generally, what is the relationship between the existing lexicon generation research and the polarity-change detection problem (which is studied in this work)?

First, let us dive a bit deeper into SP. As a lexicon generation approach, its goal is to collect sentiment words from a given corpus and infer their sentiment scores. There are two important notes: (a) while SP could discover more sentiment words, those extracted words could be wrong. For example, SP extracts the word "product" as a sentiment word and assigns it a positive (+ve) sentiment. This could lead to mis-classifications of the negative (-ve) sentences containing "product". (b) while SP directly discovers and estimates sentiment words, it does not know which sentiment words carry important domain-shifted sentiment. For example, SP discovers "excellent", "crush", "terrible" for the domain *blender* and estimates the sentiment scores as 0.9, 0.7, and 0.1 (for simplicity, let us assume all scores are rescaled to [0.0, 1.0], where 1.0 denotes most

+ve and 0.0 most -ve). Those scores indicate their polarities, but do not reflect their importance/effect of polarity change towards that domain.

For DSG, (a) could be avoided because "product" is usually excluded in a general-purpose lexicon. Regarding (b), say the scores of "excellent", "crush" and "bad" are 1.0, 0.0, and 0.0 in the original lexicon; with the domain sentiment re-estimation from DSG, they become 0.9, 0.7, and 0.1. Their polarity-changed scores are thus inferred as 0.1 $(|1.0-0.9|)$, 0.7, and 0.1, where "crush" can be found as an important domain-sentiment changed word (0.7).

Certainly, one can compare the SP generated lexicon to the original/general lexicon to detect the changes. We already did this for the detection task (Table 5). Here we design a variant SP-dsg-like, following this idea to perform the sentiment classification task. The main difference between SP and SP-dsg-like is that SP directly uses its generated lexicon and sentiment scores, while SP-dsg-like uses its generated lexicon to modify the original lexicon (OL) like DSG does. However, SP-dsg-like performs poorly (Table 8), mainly because the modified lexicon (based on OL) does not fully reflect the whole sentiment words generated by SP.

Model	fridge	blender	washer	movie	AVG
OL	61.20%	65.42%	62.06%	56.72%	61.35%
SP	68.10%	78.97%	66.67%	87.87%	75.40%
SP-dsg-like	67.67%	71.02%	64.36%	63.93%	66.75%
SP-dsg-like+SP	69.40%	77.57%	68.97%	83.28%	74.81%
OL + SP	62.07%	78.04%	70.11%	82.30%	73.13%
CLI + SP	62.07%	77.57%	70.67%	83.61%	73.48%
DSG + SP	**72.41%**	**78.97%**	**79.89%**	**88.85%**	**80.03%**

Table 8: Sentiment classification accuracy.

We then combine two lexicons together to give another variant SP-dsg-like+SP, which means the modified lexicon (based on OL) is expanded by the SP self-generated lexicon, where the SP generated lexicon can contain additional sentiment words (outside OL). Similarly, we can make OL+SP and CLI+SP, but they are all inferior to SP (Table 8). The reason is that the key polarity-changed words in the original lexicon have not been corrected, keeping their wrong sentiments for classification.

However, notice that DSG can effectively detect and correct those words; when we use DSG+SP, the overall results are improved (AVG) and even better than using DSG or SP only (Table 8 and Table 6).

It has been shown that either getting more sentiment words (from SP) or fixing a smaller number of important polarity-changed words (from DSG) can help sentiment classification (Table 6). With DSG+SP working even better, we can view the lexicon generation and the domain polarity-changed (lexical) word detection as two directions for classification improvement. Either one has its own advantage. Lexicon generation methods can induce more words and may help find rarer/infrequent words. The domain polarity-changed lexical word detection can be handy and less risky, as it directly corrects the polarities of important lexical words and would not induce noises (wrong sentiment words).

Finally, the answers to Q1 and Q2 are: using SP/lexicon-generation and DSG/polarity-change-detection can both improve sentiment classification, but in different manners (i.e., two directions as discussed above). Besides, using DSG can effectively detect important polarity-changed words, while SP does not perform very well on this task. These two directions could be complementary, as indicated by DSG+SP. Note that in this work we have shown that incorporating the DSG corrected lexical words (i.e., DSG modified lexicon) into SP can help boost its performance, which can be viewed as injecting clean/reliable domain sentiment knowledge. Another possible enhancement as future work is that, for the additional (non-lexical) sentiment words found by SP (or other domain lexicon generation method), we can use DSG to detect and correct their changed/shifted polarity in domains. However, for that, we will also need to deal with the induced noise (newly identified but wrong sentiment words like "product"), perhaps with some denoising or pruning techniques. We also hope this work and its findings can inspire more future work.

6 Conclusion

This paper studied the problem of detecting domain polarity-changed words in a sentiment lexicon. As we have seen, the wrong polarities seriously degenerate the sentiment classification performance. To address it, this paper proposed a novel solution named Domain-specific Sentiment Graph (DSG). Experimental results demonstrated its effectiveness in finding the polarity-changed words and its resulting performance gain in sentiment classification.

Acknowledgments

This work was supported in part by two grants from National Science Foundation: IIS-1910424 and IIS-1838770, a DARPA Contract HR001120C0023, and a research gift from Northrop Grumman.

References

Robert P Abelson. 1983. Whatever became of consistency theory? *Personality and Social Psychology Bulletin*, 9(1):37–54.

Himanshu Sharad Bhatt, Deepali Semwal, and Shourya Roy. 2015. An iterative similarity based adaptation technique for cross-domain text classification. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 52–61.

Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A. Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *Proceedings of WWW-2008 workshop on NLP in the Information Explosion Era*.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2008)*, pages 793–801.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-2009)*, pages 590–598.

Sanjiv R Das and Mike Y Chen. 2007. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining (WSDM-2008)*, pages 231–240.

Eduard C. Dragut, Clement Yu, Prasad Sistla, and Weiyi Meng. 2010. Construction of a sentimental word dictionary. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2010)*, pages 1761–1764.

Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. 2010. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proceedings of ACM International Confernece on Web search and data mining (WSDM-2010)*, pages 111–120.

Andrea Esuli and Fabrizio Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2005)*, pages 617–624.

Lorenzo Gatti and Marco Guerini. 2012. Assessing sentiment strength in words prior polarities. In *Proceedings of the Conference of the 24th International Conference on Computational Linguistics (COLING-2012)*.

William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 595. NIH Public Access.

Ahmed Hassan and Dragomir Radev. 2010. Identifying text polarity using random walks. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*, pages 395–403.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-1997)*, pages 174–181.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 168–177.

Sean P Igo and Ellen Riloff. 2009. Corpus-based semantic lexicon induction with web-based corroboration. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, pages 18–26. Association for Computational Linguistics.

Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. 2010. Generating focused topic-specific sentiment lexicons. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2010)*.

Nobuhiro Kaji and Masaru Kitsuregawa. 2006. Automatic construction of polarity-tagged corpus from html documents. In *Proceedings of COLING/ACL 2006 Main Conference Poster Sessions (COLING-ACL-2006)*, pages 452–459.

Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientation of adjectives. In *Proc. of LREC-2004*, volume 4, pages 1115–1118.

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2006)*, pages 355–363.

Wiltrud Kessler and Hinrich Schütze. 2012. Classification of inconsistent sentiment words using syntactic constructions. In *Proceedings of the Conference of the 24th International Conference on Computational Linguistics (COLING-2012)*.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of Interntional Conference on Computational Linguistics (COLING-2004)*, page 1367.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018a. Transformation networks for target-oriented sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 946–956.

Zheng Li, Ying Wei, Yu Zhang, and Qiang Yang. 2018b. Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, Lousiana, USA, February 2-7, 2018.*

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th international conference on World wide web (WWW-2011)*, pages 347–356.

Saif Mohammad, Cody Dunne, and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-2009)*, pages 599–608.

Kevin P Murphy, Yair Weiss, and Michael I Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc.

Antonio Moreno Ortiz. 2017. Linfmotif: Sentiment analysis for the digital humanities. In *15th Conference of the European Chapter of the Association for Computational Linguistics: Proceedings of the Software Demonstrations: April 3-7, 2017 Valencia, Spain*, pages 73–76. The Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Using very simple statistics for review search: An exploration. In *Proceedings of International Conference on Computational Linguistics (COLING-2008).*

Judea Pearl. 1982. *Reverend Bayes on inference engines: A distributed hierarchical approach.* Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles.

Wei Peng and Dae Hoon Park. 2011. Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM-2011)*, volume 51, page 61801.

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.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL-2009)*, pages 675–682.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. 2016. Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777.

Inaki San Vicente, Rodrigo Agerri, and German Rigau. 2014. Simple, robust and (almost) unsupervised generation of polarity lexicons for multiple languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 88–97.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. 2007. Extracting semantic orientations of phrases from dictionary. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL-2007).*

Zhiyang Teng, Duy Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1629–1638.

Peter D. Turney and Micharel L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems.*

Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. 2004. Developing affective lexical resources. *PsychNology Journal*, 2(1):61–83.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics (HAACL-2010)*, pages 777–785.

Svitlana Volkova, Theresa Wilson, and David Yarowsky. 2013. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-2013).*

Bo Wang and Houfeng Wang. 2008. Bootstrapping both product features and opinion words from chinese customer reviews with cross-inducing. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP-2008)*.

Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016. Mining aspect-specific opinion using a holistic lifelong topic model. In *Proceedings of the 25th international conference on world wide web*, pages 167–176.

Shuai Wang, Guangyi Lv, Sahisnu Mazumder, Geli Fei, and Bing Liu. 2018a. Lifelong learning memory networks for aspect sentiment classification. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 861–870. IEEE.

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018b. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 957–967.

Yasheng Wang, Yang Zhang, and Bing Liu. 2017. Sentiment lexicon expansion based on neural pu learning, double dictionary lookup, and polarity association. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 553–563.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 347–354.

Yunfang Wu and Miaomiao Wen. 2010. Disambiguating dynamic sentiment ambiguous adjectives. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1191–1199.

Ge Xu, Xinfan Meng, and Houfeng Wang. 2010. Build chinese emotion lexicons using a graph-based algorithm and multiple resources. In *Proceedings of the 23rd international conference on computational linguistics*, pages 1209–1217. Association for Computational Linguistics.

Min Yang, Baolin Peng, Zheng Chen, Dingju Zhu, and Kam-Pui Chow. 2014. A topic model for building fine-grained domain-specific emotion lexicon. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*. Association for Computational Linguistics (ACL).

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 236–246.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Yanyan Zhao, Bing Qin, and Ting Liu. 2012. Collocation polarity disambiguation using web-based pseudo contexts. In *Proceedings of the 2012 Conference on Empirical Methods on Natural Language Processing (EMNLP-2012)*.

Li Zhuang, Feng Jing, and Xiaoyan Zhu. 2006. Movie review mining and summarization. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM-2006)*, pages 43–50.