ReNew: A Semi-Supervised Framework for Generating Domain-Specific Lexicons and Sentiment Analysis

Zhe Zhang
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
zzhang13@ncsu.edu

Munindar P. Singh
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
singh@ncsu.edu

Abstract

The sentiment captured in opinionated text provides interesting and valuable information for social media services. However, due to the complexity and diversity of linguistic representations, it is challenging to build a framework that accurately extracts such sentiment. We propose a semi-supervised framework for generating a domain-specific sentiment lexicon and inferring sentiments at the *segment* level. Our framework can greatly reduce the human effort for building a domain-specific sentiment lexicon with high quality. Specifically, in our evaluation, working with just 20 manually labeled reviews, it generates a domain-specific sentiment lexicon that yields weighted average F-Measure gains of 3%. Our sentiment classification model achieves approximately 1% greater accuracy than a state-of-the-art approach based on elementary discourse units.

1 Introduction

Automatically extracting sentiments from user-generated opinionated text is important in building social media services. However, the complexity and diversity of the linguistic representations of sentiments make this problem challenging.

High-quality sentiment lexicons can improve the performance of sentiment analysis models over general-purpose lexicons (Choi and Cardie, 2009). More advanced methods such as (Kanayama and Nasukawa, 2006) adopt domain knowledge by extracting sentiment words from the domain-specific corpus. However, depending on the context, the same word can have different polarities even in the same domain (Liu, 2012).

In respect to sentiment classification, Pang et al. (2002) infer the sentiments using basic features, such as bag-of-words. To capture more complex linguistic phenomena, leading approaches (Nakagawa et al., 2010; Jo and Oh, 2011; Kim et al., 2013) apply more advanced models but assume one document or sentence holds one sentiment. However, this is often not the case. Sentiments can change within one document, one sentence, or even one clause. Also, existing approaches infer sentiments without considering the changes of sentiments within or between clauses. However, these changes can be successfully exploited for inferring fine-grained sentiments.

To address the above shortcomings of lexicon and granularity, we propose a semi-supervised framework named ReNew. (1) Instead of using sentences, ReNew uses *segments* as the basic units for sentiment classification. Segments can be shorter than sentences and therefore help capture fine-grained sentiments. (2) ReNew leverages the relationships between consecutive segments to infer their sentiments and automatically generates a domain-specific sentiment lexicon in a semi-supervised fashion. (3) To capture the contextual sentiment of words, ReNew uses dependency relation pairs as the basic elements in the generated sentiment lexicon.

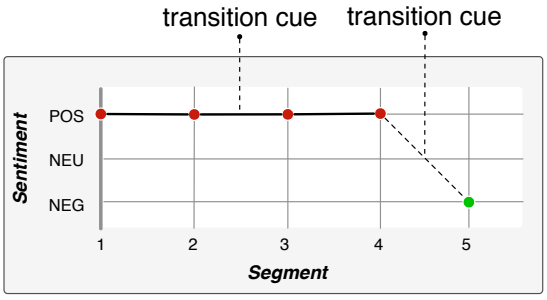

Figure 1: Segments in a Tripadvisor review.

Consider a part of a review from Tripadvisor.[1] We split it into six segments with sentiment labels.

[1] http://www.tripadvisor.com/ShowUserReviews-g32655-d81765-r100000013

"... (**1: POS**) *The hotel was clean and comfortable.* (**2: POS**) *Service was friendly* (**3: POS**) *even providing us a late-morning check-in.* (**4: POS**) *The room was quiet and comfortable,* (**5: NEG**) *but it was beginning to show a few small signs of wear and tear. ...*"

Figure 1 visualizes the sentiment changes within the text. The sentiment remains the same across Segments 1 to 4. The sentiment transition between Segments 4 and 5 is indicated by the transition cue "but"—which signals conflict and contradiction. Assuming we know Segment 4 is positive, given the fact that Segment 5 starts with "but," we can infer with high confidence that the sentiment in Segment 5 changes to neutral or negative even without looking at its content. After classifying the sentiment of Segment 5 as NEG, we associate the dependency relation pairs {"sign", "wear"} and {"sign", "tear"} with that sentiment.

ReNew can greatly reduce the human effort for building a domain-specific sentiment lexicon with high quality. Specifically, in our evaluation on two real datasets, working with just 20 manually labeled reviews, ReNew generates a domain-specific sentiment lexicon that yields weighted average F-Measure gains of 3%. Additionally, our sentiment classification model achieves approximately 1% greater accuracy than a state-of-the-art approach based on elementary discourse units (Lazaridou et al., 2013).

The rest of this paper is structured as follows. Section 2 introduces some essential background. Section 3 illustrates ReNew. Section 4 presents our experiments and results. Section 5 reviews some related work. Section 6 concludes this paper and outlines some directions for future work.

2 Background

Let us introduce some of the key terminology used in ReNew. A **segment** is a sequence of words that represents at most one sentiment. A segment can consist of multiple consecutive clauses, up to a whole sentence. Or, it can be shorter than a clause. A **dependency relation** defines a binary relation that describes whether a pairwise syntactic relation among two words holds in a sentence. In ReNew, we exploit the Stanford typed dependency representations (de Marneffe et al., 2006) that use triples to formalize dependency relations. A domain-specific sentiment lexicon contains three lists of dependency relations, associated respectvely with positive, neutral, or negative sentiment.

Given a set of reviews, the tasks of sentiment analysis in ReNew are (1) splitting each review into segments, (2) associating each segment with a sentiment label (positive, neutral, negative), and (3) automatically generating a domain-specific sentiment lexicon. We employ Conditional Random Fields (Lafferty et al., 2001) to predict the sentiment label for each segment. Given a sequence of segments $\bar{x} = (x_1, \cdots, x_n)$ and a sequence of sentiment labels $\bar{y} = (y_1, \cdots, y_n)$, the CRFs model $p(\bar{y}|\bar{x})$ as follows.

$$p(\bar{y}|\bar{x}) = \frac{1}{Z(\bar{x})} \exp \sum_{j}^{J} (\omega_j \cdot F_j(\bar{x}, \bar{y}))$$

$$F_j(\bar{x}, \bar{y}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \bar{x}, i)$$

where ω is a set of weights learned in the training process to maximize $p(\bar{y}|\bar{x})$. $Z(\bar{x})$ is a normalization constant that is the sum of all possible label sequences. And, F_j is a feature function that sums f_j over $i \in (1, n)$, where n is the length of $\bar{y}$, and f_j can have arbitrary dependencies on the observation sequence $\bar{x}$ and neighboring labels.

3 Framework

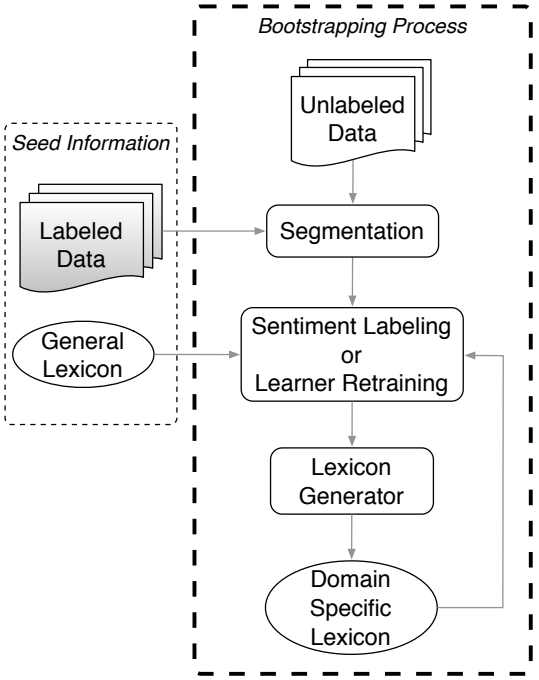

Figure 2: The ReNew framework schematically.

Figure 2 illustrates ReNew. Its inputs include

a general sentiment lexicon and a small labeled training dataset. We use a general sentiment lexicon and the training dataset as prior knowledge to build the initial learners.

On each iteration in the bootstrapping process, additional unlabeled data is first segmented. Second, the learners predict labels for segments based on current knowledge. Third, the lexicon generator determines which newly learned dependency relation triples to promote to the lexicon. At the end of each iteration, the learners are retrained via the updated lexicon so as to classify better on the next iteration. After labeling all of the data, we obtain the final version of our learners along with a domain-specific lexicon.

3.1 Rule-Based Segmentation Algorithm

Algorithm 1 Rule-based segmentation.

Require: Review dataset T
1: **for all** review r in T **do**
2: Remove HTML tags
3: Expand typical abbreviations
4: Mark special name-entities
5: **for all** sentence m in r **do**
6: **while** m contains a transition cue **and** m is not empty **do**
7: Extract subclause p that contains the transition cue
8: Add p as segment s into segment list
9: Remove p from m
10: **end while**
11: Add the remaining part in m as segment s into segment list
12: **end for**
13: **end for**

The algorithm starts with a review dataset T. Each review r from dataset T is first normalized by a set of hard-coded rules (lines 2–4) to remove unnecessary punctuations and HTML tags, expand typical abbreviations, and mark special name entities (e.g., replace a URL by #LINK# and replace a monetary amount "$78.99" by #MONEY#).

After the normalization step, it splits each review r into sentences, and each sentence into subclauses (lines 6–10) provided transition cues occur. In effect, the algorithm converts each review into a set of segments.

Note that ReNew captures and uses the sentiment changes. Therefore, our segmentation algorithm considers only two specific types of transition cues including contradiction and emphasis.

3.2 Sentiment Labeling

ReNew starts with a small labeled training set. Knowledge from this initial training set is not sufficient to build an accurate sentiment classification model or to generate a domain-specific sentiment lexicon. Unlabeled data contains rich knowledge, and it can be easily obtained. To exploit this resource, on each iteration, the sentiment labeling component, as shown in Figure 3, labels the data by using multiple learners and a label integrator. We have developed a forward (FR) and a backward relationship (BR) learner to learn relationships among segments.

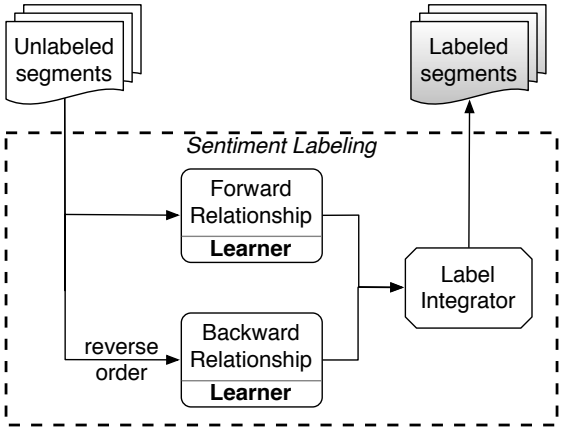

Figure 3: Sentiment labeling.

3.2.1 FR and BR Learners

The FR learner learns the relationship between the current segment and the next. Given the sentiment label and content of a segment, it tries to find the best possible sentiment label of the next segment. The FR Learner tackles the following situation where two segments are connected by a transition word, but existing knowledge is insufficient to infer the sentiment of the second segment. For instance, consider the following review sentence.[2]

(1) *The location is great,* (2) ***but*** *the staff was pretty ho-hum about everything from checking in, to AM hot coffee, to PM bar.*

The sentence contains two segments. We can easily infer the sentiment polarity of Segment 1 based on the word "great" that is commonly included in many general sentiment lexicons. For Segment 2, without any context information, it is difficult to infer its sentiment. Although the

[2]http://www.tripadvisor.com/ShowUserReviews-g60763-d93589-r10006597

word "ho-hum" indicates a negative polarity, it is not a frequent word. However, the conjunction "but" clearly signals a contrast. So, given the fact that the former segment is positive, a pre-trained FR learner can classify the latter as negative. The Backward Relationship (BR) learner does the same but with the segments in each review in reverse order.

3.2.2 Label Integrator

Given the candidate sentiment labels suggested by the two learners, the label integrator first selects the label with confidence greater than or equal to a preset threshold. Segments are left unlabeled if their candidate labels belong to mutually exclusive categories with the same confidence.

3.3 Lexicon Generator

In each iteration, after labeling a segment, the lexicon generator identifies new triples automatically. As shown in Figure 4, this module contains two parts: a Triple Extractor and a Lexicon Integrator. For each sentiment, the Triple Extractor (TE) extracts candidate dependency relation triples using a novel rule-based approach. The Lexicon Integrator (LI) evaluates the proposed candidates and promotes the most supported candidates to the corresponding sentiment category in the domain-specific lexicon.

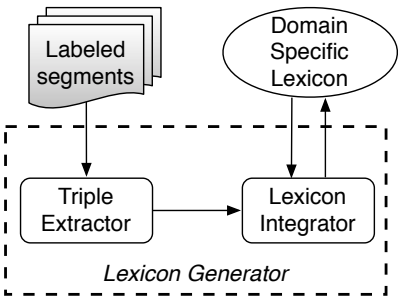

Figure 4: Lexicon generator module.

3.3.1 Triple Extractor (TE)

The TE follows the steps below, for segments that contain only one clause, as demonstrated in Figure 5 for *"The staff was slow and definitely not very friendly."* The extracted triples are $root\_nsubj$(slow, staff), $nsubj$(slow, staff), and $nsubj$(not_friendly, staff).

1. Generate a segment's dependency parse tree.
2. Identify the root node of each clause in the segment.
3. Remove all triples except those marked E in Table 1.
4. Apply the rules in Table 2 to add or modify triples.
5. Suggest the types of triples marked L in Table 1 to the lexicon integrator.

Table 1: Dependency relation types used in extracting (E) and domain-specific lexicon (L).

Types	Explanation	E	L
$amod$	adjectival modifier	√	√
$acomp$	adjectival complement	√	√
$nsubj$	nominal subject	√	√
neg	negation modifier	√	
$conj\_and$	words coordinated by "and" or similar	√	
$prep\_with$	words coordinated by "with"	√	
$root$	root node	√	
$root\_amod$	amod root node		√
$root\_acomp$	acomp root node		√
$root\_nsubj$	nsubj root node		√
$neg\_pattern$	"neg" pattern		√

Table 1 describes all seven types of triples used in the domain-specific lexicon. Among them, $amod$, $acomp$, and $nsubj$ are as in (de Marneffe et al., 2006). And, $root\_amod$ captures the root node of a sentence when it also appears in the adjectival modifier triple, similarly for $root\_acomp$ and $root\_nsubj$. We observe that the word of the root node is often related to the sentiment of a sentence and this is especially true when this word also appears in the adjectival modifier, adjectival complement, or negation modifier triple.

Zhang et al. (2010) propose the $no\_pattern$ that describes a word pair whose first word is "No" followed by a noun or noun phrase. They show that this pattern is a useful indicator for sentiment analysis. In our dataset, in addition to "No," we observe the frequent usage of "Nothing" followed by an adjective. For example, users may express a negative feeling about a hotel using sentence such as "Nothing special." Therefore, we create the $neg\_pattern$ to capture a larger range of possible word pairs. In ReNew, $neg\_pattern$ is "No" or "Nothing" followed by a noun or noun phrase or an adjective.

3.3.2 Lexicon Integrator (LI)

The Lexicon Integrator promotes candidate triples with a frequency greater than or equal to a preset

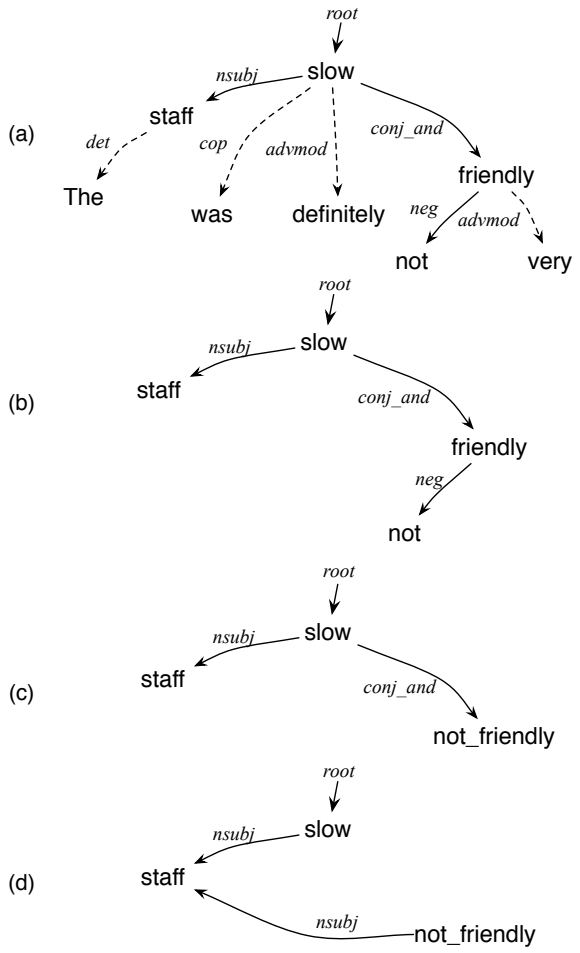

Figure 5: Extracting sentiment triples from a segment that contains one clause. (a) The initial dependency parse tree. (b) Remove nonsentiment triples. (c) Handle negation triples. (d) Build relationships.

threshold. The frequency list is updated in each iteration. The LI first examines the prior knowledge represented as an ordered list of the governors of all triples, each is attached with an ordered list of its dependents. Then, based on the triples promoted in this iteration, the order of the governors and their dependents is updated. Triples are not promoted if their governors or dependents appear in a predetermined list of stopwords.

The LI promotes triples by respecting mutual exclusion and the existing lexicon. In particular, it does not promote triples if they exist in multiple sentiment categories or if they already belong to a different sentiment category.

Finally, for each sentiment, we obtain seven sorted lists corresponding to *amod*, *acomp*, *nsubj*, *root_amod*, *root_acomp*, *root_nsubj*, and *neg_pattern*. These lists form the domain-specific sentiment lexicon.

Table 2: Rules for extracting sentiment triples.

Rule	Function	Condition	Result
R_1	Handle Negation	word w_i; $neg(w_{gov}, w_{dep})$; $w_i = w_{gov}$;	$w_i = w_{dep} + \text{“\_”} + w_i$
R_2	Build Relationships $(conj\_and, amod)$	word w_i and w_j; $conj\_and(w_i, w_j)$; $amod(w_{gov}, w_i)$;	$amod(w_{gov}, w_i)$ $amod(w_{gov}, w_j)$
R_3	Build Relationships $(conj\_and, acomp)$	word w_i and w_j; $conj\_and(w_i, w_j)$; $acomp(w_{gov}, w_i)$;	$acomp(w_{gov}, w_i)$ $acomp(w_{gov}, w_j)$
R_4	Build Relationships $(conj\_and, nsubj)$	word w_i and w_j; $conj\_and(w_i, w_j)$; $nsubj(w_i, w_{dep})$;	$nsubj(w_i, w_{dep})$ $nsubj(w_j, w_{dep})$

3.4 Learner Retraining

At the end of each iteration, ReNew retrains each learner as shown in Figure 6. Newly labeled segments are selected by a filter. Then, given an updated lexicon, learners are retrained to perform better on the next iteration. Detailed description of the filter and learner are presented below.

3.4.1 Filter

The filter seeks to prevent labeling errors from accumulating during bootstrapping. In ReNew, newly acquired training samples are segments with labels that are predicted by old learners. Each predicted label is associated with a confidence value. The filter is applied to select those labeled segments with confidence greater than or equal to a preset threshold.

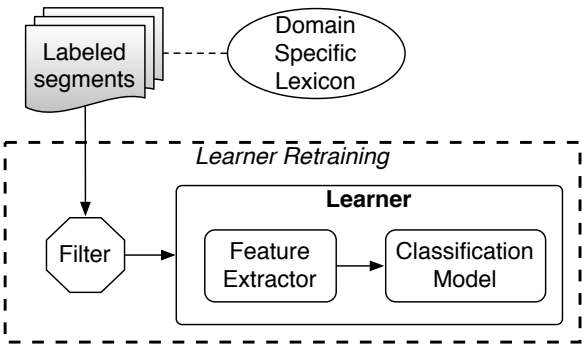

Figure 6: Retrain a relationship learner.

3.4.2 Learner

As Section 3.2 describes, ReNew uses learners to capture different types of relationships among segments to classify sentiment by leveraging these relationships. Each learner contains two components: a feature extractor and a classification model. To train a learner, the feature extractor first converts labeled segments into feature vectors

Table 3: A list of transition types used in ReNew.

Transition Types	Examples
Agreement, Addition, and Similarity	also, similarly, as well as, ...
Opposition, Limitation, and Contradiction	but, although, in contrast, ...
Cause, Condition, and Purpose	if, since, as/so long as, ...
Examples, Support, and Emphasis	including, especially, such as, ...
Effect, Consequence, and Result	therefore, thus, as a result ...
Conclusion, Summary, and Restatement	overall, all in all, to sum up, ...
Time, Chronology, and Sequence	until, eventually, as soon as, ...

for training a CRF-based sentiment classification model. The feature extractor generates five kinds of features as below.

Grammar: part-of-speech tag of every word, the type of phrases and clauses (if known).

Opinion word: To exploit a general sentiment lexicon, we use two binary features indicating the presence or absence of a word in the positive or negative list in a general sentiment lexicon.

Dependency relation: The lexicon generated by ReNew uses the Stanford typed dependency representation as its structure.

Transition cue: For tracking the changes of the sentiment, we exploit seven types of transition cues, as shown in Table 3.

Punctuation, special name-entity, and segment position: Some punctuation symbols, such as "!", are reliable carriers of sentiments. We mark special named-entities, such as time, money, and so on. In addition, we use segment positions (beginning, middle, and end) in reviews as features.

4 Experiments

To assess ReNew's effectiveness, we prepare two hotel review datasets crawled from Tripadvisor. One dataset contains a total of 4,017 unlabeled reviews regarding 802 hotels from seven US cities. The reviews are posted by 340 users, each of whom contributes at least ten reviews. The other dataset contains 200 reviews randomly selected from Tripadvisor. We collected ground-truth labels for this dataset by inviting six annotators in two groups of three. Each group labeled the same 100 reviews. We obtained the labels for each segment consist as positive, neutral, or negative. Fleiss' kappa scores for the two groups were 0.70 and 0.68, respectively, indicating substantial agreement between our annotators.

The results we present in the remainder of this section rely upon the following parameter values. The confidence thresholds used in the Label Integrator and filter are both set to 0.9 for positive labels and 0.7 for negative and neutral labels. The minimum frequency used in the Lexicon Integrator for selecting triples is set to 4.

4.1 Feature Function Evaluation

Our first experiment evaluates the effects of different combinations of features. To do this, we first divide all features into four basic feature sets: *T* (transition cues), *P* (punctuations, special name-entities, and segment positions), *G* (grammar), and *OD* (opinion words and dependency relations). We train 15 sentiment classification models using all basic features and their combinations. Figure 7 shows the results of a 10-fold cross validation on the 200-review dataset (light grey bars show the accuracy of the model trained without using transition cue features).

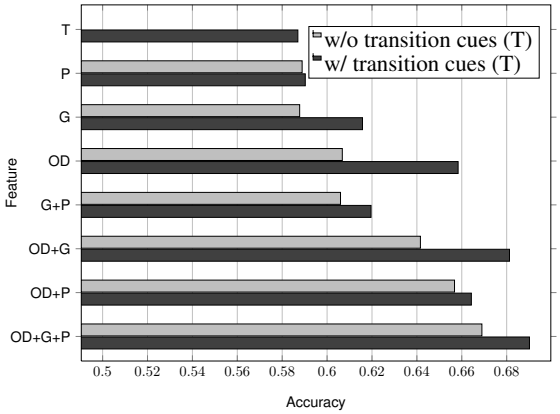

Figure 7: Accuracy using different features.

The feature *OD* yields the best accuracy, followed by *G*, *P*, and *T*. Although *T* yields the worst accuracy, incorporating it improves the resulting accuracy of the other features, as shown by the dark grey bars. In particular, the accuracy of *OD* is markedly improved by adding *T*. The model trained using all the feature sets yields the best accuracy.

4.2 Relationship Learners Evaluation

Our second experiment evaluates the impact of the relationship learners and the label integrator. To this end, we train and compare sentiment classification models using three configurations. The first configuration (FW-L) uses only the FR learner; the second (BW-L) only the BR learner. ALL-L uses both the FR and BR learners, together with a label integrator. We evaluate them with 10-fold cross

validation on the 200-review dataset.

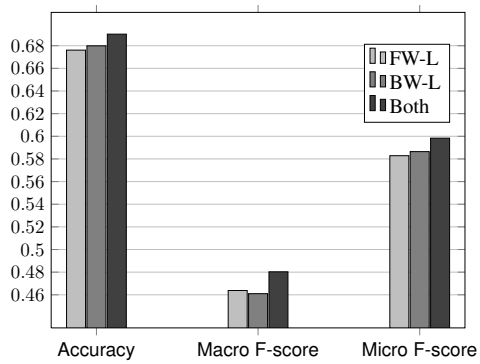

Figure 8: Comparison among the learners.

Figure 8 reports the accuracy, macro F-score, and micro F-score. It shows that the BR learner produces better accuracy and a micro F-score than the FR learner but a slightly worse macro F-score. Jointly considering both learners with the label integrator achieves better results than either alone. The results demonstrate the effectiveness of our sentiment labeling component.

4.3 Domain-Specific Lexicon Assessment

Our third experiment evaluates the quality of the domain-specific lexicon automatically generated by ReNew. To do this, we first transform each of the 200 labeled reviews into feature vectors. Then we retrain Logistic Regression models using WEKA (Hall et al., 2009). Note that we use only the features extracted from the lexicons themselves. This is important because to compare only the lexicons' impact on sentiment classification, we need to avoid the effect of other factors, such as syntax, transition cues, and so on. We compare models trained using (1) our domain-specific lexicon, (2) Affective Norms for English Words (ANEW) (Bradley and Lang, 1999), and (3) Linguistic Inquiry and Word Count (LIWC) (Tausczik and Pennebaker, 2010). ANEW and LIWC are well-known general sentiment lexicons.

Table 4 shows the results obtained by 10-fold cross validation. Each weighted average is computed according to the number of segments in each class. The table shows the significant advantages of the lexicon generated by ReNew. ANEW achieves the highest recall for the positive class, but the lowest recalls in the negative and neutral classes. Regarding the neutral class, both ANEW and LIWC achieve poor results. The weighted average measures indicate our lexicon has the highest overall quality.

Our domain-specific lexicon contains distinguishable aspects associated with sentiment words. For example, the aspect "staff" is associated with positive words (e.g., "nice," "friendli," "help," "great," and so on) and negative words (e.g., "okai," "anxiou," "moodi," "effici," and so on). We notice that some positive words also occur on the negative side. This may be for two reasons. First, some sentences that contain positive words may convey a negative sentiment, such as "The staff should be more efficient." Second, the bootstrapping process in ReNew may introduce some wrong words by mistakenly labeling the sentiment of the segments. These challenges suggest useful directions for the future work.

4.4 Lexicon Generation and Sentiment Classification

Our fourth experiment evaluates the robustness of ReNew's lexicon generation process as well as the performance of the sentiment classification models using these lexicons. We first generate ten domain-specific lexicons by repeatedly following these steps: For the first iteration, (1) build a training dataset by randomly selecting 20 labeled reviews (about 220 segments) and (2) train the learners using the training dataset and LIWC. For each iteration thereafter, (1) label 400 reviews from the unlabeled dataset (4,071 reviews) and (2) update the lexicon and retrain the learners. After labeling all of the data, output a domain-specific lexicon.

To evaluate the benefit of using domain-specific sentiment lexicons, we train ten sentiment classification models using the ten lexicons and then compare them, pairwise, against models trained with the general sentiment lexicon LIWC. Each model consists of an FR learner, a BR learner, and a label integrator. Each pairwise comparison is evaluated on a testing dataset with 10-fold cross validation. Each testing dataset consists of 180 randomly selected reviews (about 1,800 segments). For each of the pairwise comparisons, we conduct a paired t-test to determine if the domain-specific sentiment lexicon can yield better results.

Figure 9 shows the pairwise comparisons of accuracy between the two lexicons. Each group of bars represents the accuracy of two sentiment classification models trained using LIWC (CRFs-General) and the generated domain-specific lexicon (CRFs-Domain), respectively. The solid line corresponds to a baseline model that takes the ma-

Table 4: Comparison results of different lexicons.

	ANEW			LIWC			ReNew		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Positive	0.59	**0.994**	0.741	0.606	0.975	0.747	**0.623**	0.947	**0.752**
Negative	0.294	0.011	0.021	**0.584**	0.145	0.232	0.497	**0.202**	**0.288**
Neutral	0	0	0	0	0	0	**0.395**	**0.04**	**0.073**
Weighted average	0.41	0.587	0.44	0.481	0.605	0.489	**0.551**	**0.608**	**0.518**

jority classification strategy. Based on the distribution of the datasets, the majority class of all datasets is positive. We can see that models using either the general lexicon or the domain-specific lexicon achieve higher accuracy than the baseline model. Domain-specific lexicons produce significantly higher accuracy than general lexicons. In the figures below, we indicate significance to 10%, 5%, and 1% as '$\cdot$', '$*$', and '$**$', respectively.

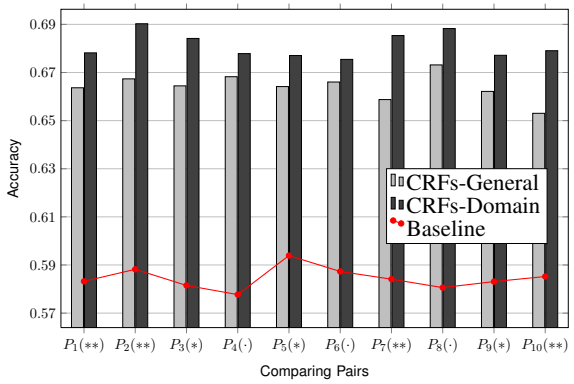

Figure 9: Accuracy with different lexicons.

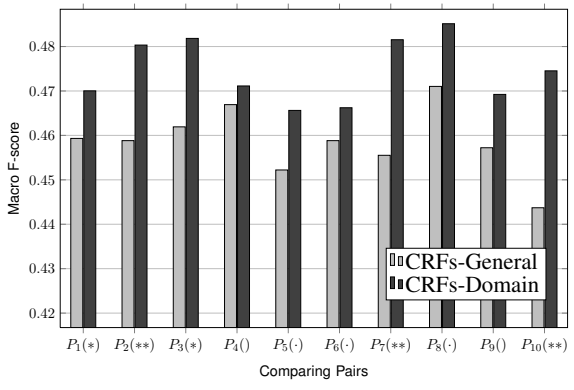

Figure 10: Macro F-score with different lexicons.

Figure 10 and 11 show the pairwise comparisons of macro and micro F-score together with the results of the paired t-tests. We can see that the domain-specific lexicons (dark-grey bars) consistently yield better results than their corresponding general lexicons (light-grey bars).

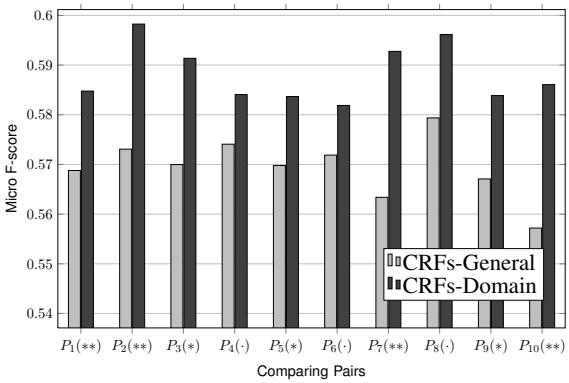

Figure 11: Micro F-score with different lexicons.

ReNew starts with LIWC and a labeled dataset and generates ten lexicons and sentiment classification models by iteratively learning 4,017 unlabeled reviews without any human guidance. The above results show that the generated lexicons contain more domain-related information than the general sentiment lexicons. Also, note that the labeled datasets we used contain only 20 labeled reviews. This is an easy requirement to meet.

4.5 Comparison with Previous Work

Our fifth experiment compares ReNew with Lazaridou et al.'s (2013) approach for sentiment classification using discourse relations. Like ReNew, Lazaridou et al.'s approach works on the sub sentential level. However, it differs from ReNew in three aspects. First, the basic units of their model are elementary discourse units (EDUs) from Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). Second, their model considers the forward relationship between EDUs, whereas ReNew captures both forward and backward relationship between segments. Third, they use a generative model to capture the transition distributions over EDUs whereas ReNew uses a discriminative model to capture the transition sequences of segments.

EDUs are defined as minimal units of text and consider many more relations than the two types

Table 5: Comparison of our framework with previous work on sentiment classification.

Method	Accuracy
EDU-Model (Lazaridou et al.)	0.594
ReNew (our method)	**0.605**

of transition cues underlying our segments. We posit that EDUs are too fine-grained for sentiment analysis. Consider the following sentence from Lazaridou et al.'s dataset with its EDUs identified.

(1) *My husband called the front desk* (2) *to complain.*

Unfortunately, EDU (1) lacks sentiment and EDU (2) lacks the topic. Although Lazaridou et al.'s model can capture the forward relationship between any two consecutive EDUs, it cannot handle such cases because their model assumes that each EDU is associated with a topic and a sentiment. In contrast, ReNew finds just one segment in the above sentence.

Just to compare with Lazaridou et al., we apply our sentiment labeling component at the level of EDUs. Their labeled dataset contains 65 reviews, corresponding to 1,541 EDUs. Since this dataset is also extracted from Tripadvisor, we use the domain-specific lexicon automatically learned by ReNew based on our 4,071 unlabeled reviews. Follow the same training and testing regimen (10-fold cross validation), we compare ReNew with their model. As shown in Table 5, ReNew outperforms their approach on their dataset: Although ReNew is not optimized for EDUs, it achieves better accuracy.

5 Related Work

Two bodies of work are relevant. First, to generate sentiment lexicons, existing approaches commonly generate a sentiment lexicon by extending dictionaries or sentiment lexicons. Hu and Liu (2004), manually collect a small set of sentiment words and expand it iteratively by searching synonyms and antonyms in WordNet (Miller, 1995). Rao and Ravichandran (2009) formalize the problem of sentiment detection as a semi-supervised label propagation problem in a graph. Each node represents a word, and a weighted edge between any two nodes indicates the strength of the relationship between them. Esuli and Sebastiani (2006) use a set of classifiers in a semi-supervised fashion to iteratively expand a manually defined lexicon. Their lexicon, named SentiWordNet, comprises the synset of each word obtained from WordNet. Each synset is associated with three sentiment scores: positive, negative, and objective.

Second, for sentiment classification, Nakagawa et al. (2010) introduce a probabilistic model that uses the interactions between words within one sentence for inferring sentiments. Socher et al. (2011) introduce a semi-supervised approach that uses recursive autoencoders to learn the hierarchical structure and sentiment distribution of a sentence. Jo and Oh (2011) propose a probabilistic generative model named ASUM that can extract aspects coupled with sentiments. Kim et al. (2013) extend ASUM by enabling its probabilistic model to discover a hierarchical structure of the aspect-based sentiments. The above works apply sentence-level sentiment classification and their models are not able to capture the relationships between or among clauses.

6 Conclusions and Future Work

The leading lexical approaches to sentiment analysis from text are based on fixed lexicons that are painstakingly built by hand. There is little a priori justification that such lexicons would port across application domains. In contrast, ReNew seeks to automate the building of domain-specific lexicons beginning from a general sentiment lexicon and the iterative application of CRFs. Our results are promising. ReNew greatly reduces the human effort for generating high-quality sentiment lexicons together with a classification model. In future work, we plan to apply ReNew to additional sentiment analysis problems such as review quality analysis and sentiment summarization.

Acknowledgments

Thanks to Chung-Wei Hang, Chris Healey, James Lester, Steffen Heber, and the anonymous reviewers for helpful comments. This work is supported by the Army Research Laboratory in its Network Sciences Collaborative Technology Alliance (NS-CTA) under Cooperative Agreement Number W911NF-09-2-0053 and by an IBM Ph.D. Scholarship and an IBM Ph.D. fellowship.

References

Margaret M. Bradley and Peter J. Lang. 1999. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL.

Yejin Choi and Claire Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the* 14^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 590–598, Singapore.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the* 5^{th} *Conference on Language Resources and Evaluation (LREC)*, pages 449–454, Genoa, Italy.

Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the* 5^{th} *Conference on Language Resources and Evaluation (LREC)*, pages 417–422, Genoa, Italy.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *SIGKDD Explorations Newsletter*, 11(1):10–18, November.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the* 10^{th} *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177, Seattle.

Yohan Jo and Alice Haeyun Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the* 4^{th} *ACM International Conference on Web Search and Data Mining (WSDM)*, pages 815–824, Hong Kong.

Hiroshi Kanayama and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the* 11^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–363, Sydney.

Suin Kim, Jianwen Zhang, Zheng Chen, Alice H. Oh, and Shixia Liu. 2013. A hierarchical aspect-sentiment model for online reviews. In *Proceedings of the* 27^{th} *AAAI Conference on Artificial Intelligence (AAAI)*, pages 804–812, Bellevue.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the* 18^{th} *International Conference on Machine Learning (ICML)*, pages 282–289, San Francisco.

Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. 2013. A Bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *Proceedings of the* 51^{st} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1630–1639, Sofia, Bulgaria.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, San Rafael, CA.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of the* 11^{th} *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 786–794, Los Angeles.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the* 7^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, Philadelphia.

Delip Rao and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the* 12^{th} *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 675–682, Athens.

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* 16^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–161, Edinburgh.

Yla R. Tausczik and James W. Pennebaker. 2010. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, March.

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* 23^{rd} *International Conference on Computational Linguistics (COLING)*, pages 1462–1470, Beijing.