Learning Sentiment Composition from Sentiment Lexicons

Orith Toledo-Ronen Roy Bar-Haim* Alon Halfon Charles Jochim
Amir Menczel Ranit Aharonov Noam Slonim
IBM Research
{oritht,roybar,alonhal,amir.menczel,ranita,noams}@il.ibm.com
charlesj@ie.ibm.com

Abstract

Sentiment composition is a fundamental sentiment analysis problem. Previous work relied on manual rules and manually-created lexical resources such as negator lists, or learned a composition function from sentiment-annotated phrases or sentences. We propose a new approach for learning sentiment composition from a large, unlabeled corpus, which only requires a word-level sentiment lexicon for supervision. We automatically generate large sentiment lexicons of bigrams and unigrams, from which we induce a set of lexicons for a variety of sentiment composition processes. The effectiveness of our approach is confirmed through manual annotation, as well as sentiment classification experiments with both phrase-level and sentence-level benchmarks.

1 Introduction

Many sentiment analysis systems rely primarily on the sentiment of individual words. However, precise sentiment analysis often requires lexical-semantic knowledge that goes beyond word-level sentiment, even when dealing with short phrases such as bigrams. Phrases may be idiomatic, in which case their polarity cannot be inferred from the sentiment of their constituent words, e.g., *black market*, *on track*, and *let down*. The sentiment of most phrases, however, is *compositional*, namely, it can be determined from the interaction between their constituents.

Sentiment composition is a fundamental problem in sentiment analysis. It involves a variety of semantic phenomena, the most studied of which are *valence shifters* (Polanyi and Zaenen, 2004) that reverse sentiment polarity, e.g., ***not*** *happy*, ***hardly*** *helpful*, ***decrease*** *unemployment*, or change its intensity (***deeply/somewhat*** *disappointed*). Sentiment composition also needs to resolve the polarity of phrases containing opposing polarities, such as *cure cancer*, *sophisticated crime*, and *fake success*. Another, less studied type of sentiment composition is expressions such as *high income*, *long queue*, and *fast learner*, that include gradable adjectives. The sentiment of these expressions cannot be derived from the sentiment of their individual words (none of which have clear sentiment in the above examples). While they can be treated as fixed expressions, similar to idioms, it is beneficial to exploit their semantic compositionality. For example, for a given word *w*, the polarity of *high w*, *higher w*, and *increasing w* is likely to be the same, and opposite to the polarity of *low/lower w*. This allows more robust sentiment learning, as well as sentiment inference for such expressions.

Previous approaches for sentiment composition relied on hand-written lists of negation words and composition rules. Such knowledge is hard to acquire and is typically incomplete. Other approaches aim to learn sentiment composition from sentiment-labeled texts. However, sentiment-labeled texts might not be available for certain domains or languages.

In this work we propose a new approach for learning sentiment composition. We define a set of *sentiment composition classes* that cover a variety of sentiment composition processes, and propose a novel method for automatic acquisition of lexicons for each of these classes from a large corpus. For

* First two authors contributed equally.

Proceedings of the 27th International Conference on Computational Linguistics, pages 2230–2241
Santa Fe, New Mexico, USA, August 20-26, 2018.

example, our method learns that words like *reduce* and *preventing*, when applied to a negative expression, change its polarity from negative to positive (e.g., *preventing violence*). It also learns that the positive sentiment of words like *strong* and *incredibly* is overridden when composed with a negative expression (e.g., *strong disapproval*). In addition, we learn composition lexicons for specific gradable adjectives, e.g., words *w* such that *high w* is positive (*morale, standard, competitiveness*), or negative (*cost ,anxiety, noise*).

Unlike previous approaches, our method does not require sentiment-labeled texts, or manually-created sentiment composition lexicons. The only annotated input required is a sentiment lexicon for individual words. We train an n-gram sentiment classifier on the input sentiment lexicon, using a novel sentiment-oriented phrase embedding that is very efficient to compute and scales well for very large corpora. The classifier is then applied to a large lexicon of bigrams, as well as to the unigrams they are made of. Finally, we induce from the automatically generated sentiment lexicon of bigrams and unigrams a set of fine-grained lexicons for sentiment composition.

The accuracy of the resulting sentiment composition lexicons is confirmed via manual assessment. We also show their contribution to both phrase-level and sentence level sentiment analysis. Our results illustrate the value of our automatically-generated sentiment lexicons for investigating sentiment composition phenomena. We made both these sentiment lexicons and the resulting composition lexicons publicly available, to facilitate further research on sentiment composition.[1]

2 Related Work

Many of the previous works on sentiment analysis include some treatment of valence shifters, based on manual lists of negators, intensifiers, etc. (Wilson et al., 2005; Kennedy and Inkpen, 2006; Taboada et al., 2011). Moilanen and Pulman (2007) apply manually-composed syntactic rules for sentiment composition over the syntactic parse. Neviarouskaya et al. (2010) manually created a lexicon with fine-grained categories for sentiment composition such as *propagation* and *domination* which may resolve conflicting sentiments. Schulder et al. (2017) applied an SVM classifier with linguistic features to bootstrap manual construction of a verbal polarity shifters lexicon. Some researchers, like Choi and Cardie (2008), combined manual composition rules with machine learning.

Other works aimed to learn a sentiment composition function from sentiment-labeled phrases or sentences by employing conditional random fields (Nakagawa et al., 2010), compositional matrix-space models (Yessenalina and Cardie, 2011), statistical parsing (Dong et al., 2015) and recursive neural networks (Socher et al., 2013; Tai et al., 2015). Several researchers aimed to learn sentiment shifters from sentiment-labeled texts, e.g. Ikeda et al. (2008), Noferesti and Shamsfard (2016).

In contrast to previous methods, our method does not require sentiment-labeled texts or feature engineering. It also does not rely on manually-composed lexicons of negators, propagators or dominators, but rather learns such lexicons automatically.

Kiritchenko and Mohammad (2016) wrote that *"lexicons that include sentiment associations for multi-word phrases as well as their constituent words can be very useful in studying sentiment composition"*. They manually developed such a lexicon for a few hundreds of bigrams and trigrams with opposing sentiments (each phrase has both negative and positive words), and experimented with both supervised and unsupervised methods for classifying and ranking the sentiment of these phrases. Their work has shown that rules based on part-of-speech patterns and unigram sentiment strengths are not sufficient for determining the sentiment of phrases with opposing sentiments. This motivated the current work, which is focused on learning lexical knowledge for sentiment composition.

Following Kiritchenko and Mohammad, we also build a sentiment lexicon for phrases and their constituent words, but our lexicon is built automatically, and is three orders of magnitude larger (over 250,000 bigrams). The size of the lexicon makes it suitable for learning lexical knowledge for sentiment composition.

[1] Available at `http://www.research.ibm.com/haifa/dept/vst/debating_data.shtml`

3 Method

Our method for learning sentiment composition lexicons comprises the following steps:

1. Train an n-gram sentiment classifier on a given sentiment lexicon for unigrams.
2. Use the sentiment classifier to automatically generate large sentiment lexicons of bigrams and unigrams.
3. Extract sentiment composition lexicons based on statistics from the bigram and unigram sentiment lexicons.

The rest of the section describes each of the above steps.

3.1 The Sentiment Classifier

Following previous work (Amir et al., 2015; Rothe et al., 2016; Bar-Haim et al., 2017b), we train a sentiment classifier on a sentiment lexicon, where the word's polarity is taken to be the label, and the word embedding is the feature vector. We use the publicly-available sentiment lexicon of Hu and Liu (2004) (hereafter, HL). After removing 224 multi-word expressions, the lexicon contains 6,565 words.

The above previous work focused on learning sentiment of unigrams, and optionally of multi-word expressions (conflated into single tokens), and used word2vec embeddings (Mikolov et al., 2013). In contrast, we are interested in learning sentiment composition from bigrams, and therefore we are mostly interested in compositional bigrams, and aim to construct a sentiment lexicon that contains hundreds of thousands of them. Due to bigram sparsity, learning their embeddings requires a very large corpus.

We use in this work a proprietary English corpus containing news articles and other types of publications from over 10,000 sources. The size of this corpus is in the order of 10^{11} tokens. While word2vec is known as a scalable method for deriving word embeddings, applying it to a corpus of this size is still computationally expensive.

We therefore opted for a more lightweight method for computing sentiment-oriented embeddings. Our method is inspired by the classical work of Turney and Littman (2003), who learned word sentiments based on their pointwise mutual information (PMI) with seed sentiment words. Our method represents each n-gram (unigram or bigram) as a 6,565 dimensional vector of its *Positive PMI (PPMI)* (Levy et al., 2015) with all the words in the HL lexicon. PPMI is defined for phrases u,v as

$$PPMI(u,v) = max(0, log\frac{f(u,v)\cdot N}{f(u)\cdot f(v)}), \quad (1)$$

where $f(u)$ and $f(v)$ are the number of sentences that contain u and v in the corpus, respectively, $f(u,v)$ is the number of sentences in which u and v co-occur, and N is the number of sentences in the corpus. u and v are said to co-occur if they are found in the same sentence, within a maximum distance of 10 tokens from each other, and have no overlap. We define $PPMI(u,u) = 0$ for any u.

We convert these sparse word vectors into dense embeddings as follows. First, we train a linear SVM[2] on the PPMI vectors of the words in HL, and apply it to a large lexicon of unigrams (to be described in the next section). We then select the 2,500 most positive words and 2,500 most negative words as predicted by the classifier (words in the HL lexicon were ignored). Based on the PPMI vectors of these 5,000 selected words, we learn a projection from the 6,565 dimensional space into a reduced space of 100 dimensions. Let M be a 5,000 $\times$ 6,565 matrix whose rows are the representations of the selected words. We compute the truncated Singular Value Decomposition (Deerwester et al., 1990) for M, $M_d = U_d \cdot \Sigma_d \cdot V_d^T$, with $d = 100$, following the notation of Levy et al. (2015). A PPMI row vector x for any new n-gram u (not in HL) can then be projected into the reduced space by taking the product $x \cdot V_d$. The resulting dense representation is much more compact, and is expected to work better for lower-frequency bigrams with sparse PPMI vectors.

[2]We used LIBSVM, (Fan et al., 2008).

#	Class	Condition UG1	Condition UG2	Predicted Bigram Polarity	
	Composition Classes				Sample match
1	REV$\oplus$	w	$\ominus$	$\oplus$	**combat** loneliness
2	REV$\ominus$	w	$\oplus$	$\ominus$	**lacked** courage
3	PROP$\oplus$	$w \wedge \ominus$	$\oplus$	$\oplus$	**overwhelming** success
4	PROP$\ominus$	$w \wedge \oplus$	$\ominus$	$\ominus$	**fresh** trouble
5	DOM$\oplus$	w		$\oplus$	**improving** nutrition
6	DOM$\ominus$	w		$\ominus$	**reckless** decision
	Adjective Classes				Sample match for (high,low)
7	ADJ(a_1,a_2)$\oplus\ominus$	$E(a_1)$	w	$\oplus$	high **morale**
8	ADJ(a_1,a_2)$\oplus\ominus$	$E(a_2)$	w	$\ominus$	low **morale**
9	ADJ(a_1,a_2)$\ominus\oplus$	$E(a_1)$	w	$\ominus$	higher **inflation**
10	ADJ(a_1,a_2)$\ominus\oplus$	$E(a_2)$	w	$\oplus$	lower **inflation**

Table 1: Definition of composition and adjective classes. Words in bold in the sample matches belong to the class.

The resulting 100-dimensional embeddings of the words in HL were used to retrain the SVM classifier. In order to assess the accuracy of the classifier, we performed a 100-fold cross validation experiment over HL words. For each fold, we removed from the training set words that have the same stem or lemma as any word in the test set. The classifier achieved high accuracy of 94.5%, confirming the effectiveness of our method.

3.2 Generating Sentiment Lexicons

Next, we created lexicons of unigrams and bigrams, computed their embeddings and applied the classifier to predict their sentiments. For the unigrams lexicon we selected words whose length is between 2 and 20 characters, are alphabetic and in WordNet (Miller, 1995), and their frequency in the corpus is at least 2000. This resulted in about 66,000 unigrams. For the bigram lexicon we selected bigrams that contain only unigrams from the unigram lexicon, and have minimum frequency of 250. The resulting lexicon of 2.8 million bigrams was still very noisy. In order to obtain a bigram lexicon of coherent phrases, we filtered out bigrams that the PMI between their unigrams is below 3.0. Finally, we only kept bigrams whose part-of-speech tag sequence form grammatical bigrams, e.g., *Adjective-Noun*, *Noun-Noun*, *Verb-Noun*, *Adverb-Adjective* etc. The resulting lexicon contains 262,555 bigrams. The classifier's real-valued predictions were normalized to the range of $[-1, 1]$. Unigrams that had sentiment in HL kept their original sentiment ($+1$ or -1).

3.3 Extracting Composition Lexicons

Finally, we define two types of classes that cover a variety of sentiment composition processes: *composition classes* model sentiment reversals and conflicting (opposing) sentiments; *adjective classes* determine the sentiment of phrases that contain specific gradable adjectives or their expansions. Our method learns a lexicon for each class, based on the unigram and bigram sentiment lexicons acquired in the previous step.

The classes are formally defined in Table 1. Each class definition has two parts: the *condition* for matching the class in a bigram, and the *predicted bigram polarity* (positive $\oplus$ or negative $\ominus$), in case the class is matched. The condition specifies in which unigram a word w from the class should be matched (the first unigram UG1 or the second unigram UG2), and optionally additional constraints on the other unigram. The table also shows a sample match for each class. For example, REV$\oplus$ (row 1 in the table) is a *Reverser*. Words in this class are matched in UG1, and if in addition UG2 is negative, the resulting bigram would be positive, e.g., ***combat*** *loneliness*. Similarly, REV$\ominus$ (row 2) flips the polarity from

positive to negative.

The other two types of composition classes are *Propagators (*PROP*)* and *Dominators (*DOM*)*. Propagators (rows 3-4) are sentiment words that, when followed by a word with opposite sentiment, "propagate" the sentiment of the other word to the bigram level, e.g., ***pure*** *vandalism*. Finally, dominators (rows 5-6) dictate the sentiment of the bigram, overriding any conflicting sentiment of the other unigram, if any.

Composition classes are extracted as follows. Let $C(c, w)$ be the set of bigrams in the bigram lexicon that satisfy the condition of class c for word w, and let $S(c)$ be the set of bigrams whose polarity is the one predicted by c. The precision of w with respect to c can be defined as:

$$P(c, w) = \frac{|S(c) \cap C(c, w)|}{|C(c, w)|} \tag{2}$$

The above formula gives uniform weights for all the bigrams in $C(c, w)$. However, we found that it is beneficial to also take into account the uncertainty stemming from the automatic sentiment prediction of the unigrams and bigrams. We do so by weighting each bigram according to strength of the predicted bigram sentiment, and for classes where the sentiment of UG2 is part of the match (REV and PROP), also according to the sentiment strength of UG2. Let $s(x)$ be the sentiment score of an n-gram x in the lexicon, and let wu be a bigram where UG1=w and UG2=u. The weighted formula is:

$$P(c, w) = \frac{\sum_{wu \in S(c) \cap C(c,w)} |s(u) \times s(wu)|}{\sum_{wu \in C(c,w)} |s(u) \times s(wu)|} \tag{3}$$

For DOM classes, we take $s(u) = 1$, since the sentiment of u is not part of the match. Weak sentiment of bigrams and unigrams (between -0.1 and 0.1 for unigrams, and between -0.05 and 0.08 for bigrams) is considered neutral sentiment.[3]

We include in c words w such that $P(c, w) > \alpha$ and $|C(c, w)| \geq k$. We use $|C(c, w)|$ as our confidence measure, by which we rank the words in each class. We took $(\alpha = 0.8, k = 10)$ for reversers and propagators, and $(\alpha = 0.95, k = 20)$ for dominators.[4]

Adjective classes are defined for specific pairs of opposing gradable adjectives (a_1, a_2) (rows 7-10). In this work we considered the pairs *(high, low)* and *(fast, slow)*. Since a_1 and a_2 are antonyms, we assume that if the bigram $a_1 w$ is positive, then the bigram $a_2 w$ is likely to be negative, and vice versa. We therefore learn two classes for each pair: ADJ(a_1,a_2)$\oplus\ominus$ contains words that are positive with a_1 and negative with a_2. ADJ(a_1,a_2)$\ominus\oplus$ contains words that are negative with a_1 and positive with a_2.

Adjective classes are learned as follows. We manually define a set of expansions $E(a)$ for each adjective a ($E(a)$ also includes a itself), e.g. *higher*, and *increasing* for *high*.[5] Let $B(a, w)$ be the set of all the bigrams in the lexicon such that UG1 is in $E(a)$ and UG2 is w. The score of a given word w with respect to the adjective pair (a_1,a_2) is defined as:

$$S_{(a_1,a_2)}(w) = \frac{\sum_{x \in B(a_1,w)} s(x)}{|B(a_1, w)|} - \frac{\sum_{x \in B(a_2,w)} s(x)}{|B(a_2, w)|} \tag{4}$$

We select for ADJ(a_1,a_2)$\oplus\ominus$ and ADJ(a_1,a_2)$\ominus\oplus$ words w with $S_{(a_1,a_2)}(w)$ above 0.1 or below -0.1, respectively, and use the absolute value of this score to rank the predictions.

4 Results

In this section we briefly describe the lexicons that were learned for each class by our method. Table 2 shows the number of words in each class. 2,783 words were learned in total. The biggest classes are the dominators, in spite of the higher thresholds that are applied to their results, compared to the reversers and the propagators. This happens because their matching condition is the weakest.

[3]The thresholds were determined heuristically by observing the resulting lexicons.

[4]α and k were selected heuristically by observing the resulting lexicons.

[5]The expansions we used are listed as part of the released dataset.

Class	Word Count
REV⊕	54
REV⊖	106
PROP⊕	48
PROP⊖	152
DOM⊕	666
DOM⊖	662
ADJ(*high,low*)⊕⊖	316
ADJ(*high,low*)⊖⊕	416
ADJ(*fast,slow*)⊕⊖	294
ADJ(*fast,slow*)⊖⊕	70
Total	2,783

Table 2: Number of words per class.

REV⊕		PROP⊕		DOM⊕	
less	wasteful	**critical**	acclaim	**unique**	culture
reduce	stress	**cloud**	solution	**beautiful**	gardens
reducing	inflammation	**proprietary**	insights	**wonderful**	atmosphere
preventing	violence	**complex**	ideas	**innovative**	approach
fewer	dropouts	**challenging**	endeavor	**digital**	capabilities
combat	inequality	**overwhelming**	victory	**excellent**	service
reduces	fraud	**invasive**	therapy	**strategic**	thinker
healthy	ageing	**intense**	commitment	**creative**	spirit
eliminate	racism	**strict**	compliance	**diverse**	environment
reduced	bureaucracy	**disruptive**	innovation	**good**	deeds
REV⊖		PROP⊖		DOM⊖	
poor	reputation	**pretty**	awful	**too**	often
too	powerful	**significant**	concern	**illegal**	actions
not	happy	**incredibly**	boring	**serious**	consequences
sexual	liberation	**strong**	disapproval	**severe**	problems
inadequate	protection	**powerful**	adversary	**alleged**	plot
lacked	commitment	**sharp**	pain	**allegedly**	abusive
bad	luck	**great**	danger	**poor**	sanitation
otherwise	decent	**fresh**	injury	**heavy**	debt
negative	contribution	**hot**	mess	**dangerous**	situation
insufficient	protection	**clearly**	drunk	**violent**	outbursts

Table 3: Top-ranked words in each composition class.

Table 3 shows the top-ranked words in each composition class. Each class word (in bold), is followed by a sample UG2 match for that word and the class in the bigram lexicon. For example, ***critical*** *acclaim* is a bigram in $C(\text{PROP}\oplus, critical)$. For the propagator classes, we only show words in HL. These words have the strongest sentiment and therefore are the most challenging for opposing sentiment resolution. In addition, in the experiment to be described in Section 6.2, we use HL as our sentiment lexicon, so only propagators in HL will be matched. Interestingly, the classes learned for REV⊕ and for REV⊖ are disjoint, indicating that the reversers operating on positive words are rather different from those operating on negative words.

Table 4 shows the top-ranked words for adjective classes. For example, *high morale* is positive, and *low morale* is negative, while for *unemployment* it is the other way around.

ADJ(*high,low*)⊕⊖	ADJ(*high,low*)⊖⊕	ADJ(*fast,slow*)⊕⊖	ADJ(*fast,slow*)⊖⊕
morale	unemployment	economic	turns
productivity	poverty	broadband	temper
ceilings	crime	implementation	spiral
literacy	costs	loading	escalation
standards	anxiety	response	buck
quality	noise	progress	spreading
ethical	violence	wit	spread
profitability	debt	learner	breathing
visibility	inequality	reaction	escalates
competitiveness	handedness	growth	population

Table 4: Top-ranked words in adjective classes.

5 Manual Asssessment

We conducted manual assessment of the precision of each class, by measuring the fraction of bigrams that have the polarity predicted by the class, out of the bigrams matched by the class. To this end, we selected from the bigram lexicon a random sample of bigrams matched by each class. Recall that some of the classes require positive or negative sentiment in UG1 and/or UG2. Our goal is to estimate the precision of the predicted bigram polarity, given that the matching is correct. In order to avoid incorrect sentiment matches caused by errors in our automatically-learned unigram sentiment lexicon, we restricted here the sentiment matches to words from the HL lexicon. We sampled 100 bigrams for each composition class, except for PROP⊕, which only had 59 matching bigrams[6]. In addition, we sampled 100 bigrams for each adjective pair (conflating the positive and negative classes for each pair).

Each of the resulting 759 bigrams was annotated by three in-house human annotators, who were asked to assess the bigram sentiment (positive/negative/neutral). They also had the option to indicate that the bigram is an incomplete phrase. Fleiss' kappa (Fleiss, 1971) for inter-annotator agreement was 0.61, indicating *substantial agreement* (Landis and Koch, 1977). The gold label for each bigram was obtained by taking the majority annotation. Ties were adjudicated by one of the authors (34 in total). We then removed the 59 bigrams assessed as incomplete, and computed precision for each class over the remaining 700 bigrams.

The results are summarized in Table 5. The precision achieved by the various classes seems very promising, in particular considering that the only sentiment-annotated input for learning these classes was a unigram sentiment lexicon. Error analysis reveals that the vast majority of misclassifications (84.6%) were labeled as neutral by the majority of the annotators, and in 25.1% of them, one annotator agreed with the class prediction, and the other two were neutral.

The learned classes provide a good starting point for semi-automatic construction of composition lexicons. We believe that their quality can be easily improved with manual filtering of our results, which can be done rather quickly.

6 Experiments

In this section we demonstrate the contribution of the learned composition and adjective classes to both phrase-level and sentence-level sentiment classification tasks. In line with our method for learning the classes, we focus in this section on sentiment analysis methods that do not rely on sentiment-labeled phrases or sentences for training.

6.1 Phrase Polarity Classification

In this experiment we test the extracted lexicons for composition and adjective classes and the bigram sentiment lexicon on the Opposing Polarity Phrases (OPP) lexicon released by Kiritchenko and Mo-

[6]Note that for this class we require here that UG1 is negative in HL and UG2 is positive in HL.

Class	Precision		
	⊕	⊖	Total
REV	0.79	0.78	0.78
PROP	0.75	0.81	0.79
DOM	0.71	0.79	0.75
ADJ(*high,low*)			0.68
ADJ(*fast,slow*)			0.75
All			0.76

Table 5: Precision assessment.

Exp	Sentiment Score	Accuracy (All)	Coverage	Accuracy (Covered)
c0	Most polar unigram	0.668		
c1	Composition	0.713	0.355	0.761
c2	Bigram score	0.726	0.309	0.863
c3	Bigram score or Composition	0.746	0.541	0.807

Table 6: OPP bigram binary classification accuracy and coverage results.

hammad (2016), which consists of phrases with words of opposing polarity. The dataset contains the sentiment score of each n-gram, the part-of-speech tags of the constituent unigrams, and the sentiment of all the unigrams included in the phrases, with a total of 307 bigram and 262 trigram phrases with a non-zero sentiment score. The phrases were collected from Twitter and the sentiment scores were obtained by manual annotations. The purpose of the OPP lexicon is to train and test sentiment composition models for opposing polarity phrases and therefore it is suitable for testing our classes and bigram sentiment prediction methods. Kiritchenko and Mohammad tested both supervised and unsupervised methods for sentiment classification and ranking. Here, we focus on unsupervised classification of bigrams.

Table 6 shows our experimental results. The baseline, c0, is the best-performing unsupervised method reported by Kiritchenko and Mohammad. This method determines the bigram polarity according to the most polar unigram score. Then, in experiment c1 (*Composition*), we predict the sentiment of the bigram by trying to match each of the classes in the bigram according to the following order: ADJ, REV, PROP, DOM. The order is based on the match specificity of the other word in the bigram (the non-class word): ADJ requires a match in a small set of adjective expansions, REV and PROP only require a certain polarity (positive/negative), and DOM does not restrict the other word. The first class that is applicable for the bigram is selected. For the matched class, we predict the bigram sentiment according to the rules in Table 1. In our second experiment c2, we use the predicted bigram score, and in c3 we use the two methods in cascade: if the bigram score is not available, we try the composition method of c1. In all three experiments, we back off to the most polar unigram score (c0) in cases that the test bigram is not covered by our predictions, so the accuracy results under the *Accuracy (All)* column are reported over the complete dataset. The *coverage* of each method is defined as the fraction of predicted bigrams out of all the bigrams in the dataset. Finally, *Accuracy (Covered)* is the accuracy only for the bigrams covered by each of the methods c1-c3.

The best performance is achieved by configuration c3, which utilizes both the bigram sentiment scores and the composition and adjective classes. By combining the two methods, we are able to address both idiomatic phrases (e.g. *top gun*) and compositional phrases (e.g. *great loss*) in the OPP dataset. c3 achieves an absolute improvement of 7.8% in accuracy with respect to the baseline c0, over the whole dataset. The differences between c1-c3 and the baseline c0 are all statistically significant, with $p < 0.05$ for c1 and $p < 0.01$ for c2 and c3, according to McNemar's test (Everitt, 1992). The results show that our sentiment and composition lexicons are effective even in a domain such as Twitter, which is rather different from the corpus we used for learning.

	Positive			Negative				
	Precision	Reccall	F_1	Precision	Recall	F_1	Accuracy	Coverage
Baseline	0.703	0.462	0.558	0.810	0.525	0.637	0.761	0.653
+ADJ	0.703	**0.480**	**0.571**	**0.815**	**0.530**	**0.642**	0.764	**0.665**
+REV	**0.724**	**0.487**	**0.582**	**0.822**	**0.541**	**0.653**	**0.778**	**0.665**
+PROP	**0.724**	**0.488**	**0.583**	**0.823**	**0.545**	**0.656**	**0.778**	**0.667**
+DOM	**0.720**	**0.547**	**0.622**	0.820	**0.546**	**0.655**	0.772	**0.708**

Table 7: Results from sentence polarity assessment. Bold indicates significant difference ($p < 0.05$) from baseline using approximate randomization test (Noreen, 1989).

	Sentence				Bigram		
	Match	$\uparrow$	$\downarrow$	$\leftrightarrow$	Match	$=$	$\neq$
+ADJ	59	27	9	23	63	52	11
+REV	40	31	4	5	40	35	5
+PROP	10	6	0	4	10	8	2
+DOM	903	164	132	607	1160	808	352

Table 8: Impact of class matches.

6.2 Sentence Polarity

We further assess the contribution of the composition and adjective classes to a lexicon-based sentiment classification of sentences.

The baseline system matches terms from the HL lexicon in the sentence. If the number of positive matches is greater than negative matches, we predict positive; if there are more negative matches, we predict negative; otherwise we predict neutral.

We then add the classes incrementally. If a class is matched in a bigram, it may modify the sentiment of that bigram. For example, suppose that the sentence contains the bigram *preventing cancer*. The baseline system will match *cancer* from the HL lexicon in the bigram, giving it negative sentiment of -1. Then the word *preventing* from the class REV$\oplus$ will be matched (line 1 in Table 1), since it is followed by a negative unigram. As a result, the sentiment of the bigram will become positive, and will get the score 1. The classes were added in the same order as in the previous experiment: adjectives, reversers, propagators, and dominators. We do not allow mutiple classes to match the same bigram.

For testing these lexicons we use the Claim Stance Dataset introduced by Bar-Haim et al. (2017a). This dataset contains Pro and Con claims found in Wikipedia for 55 controversial topics, and also includes labels for sentiment. We chose this dataset due to its diversity - it covers a variety of topics, and since it was also approached with a lexicon-based sentiment analysis by Bar-Haim et al. This dataset has 2,252 claim sentences that contain sentiment annotations (1012 positive and 1240 negative). We ignore sentiment found in the sentiment target that is given by the Claim Stance Dataset.

Table 7 shows results for the sentiment prediction with the baseline and after the addition of each class. We use precision, recall, and F_1 to measure performance for positive and negative classes and *accuracy* and *coverage* to measure overall performance, following previous work on this dataset. Bar-Haim et al. define *accuracy* as $\frac{\#correct}{\#predicted}$ and *coverage* as $\frac{\#predicted}{\#sentences}$. If the system returns a neutral label it is not counted as a prediction. There is consistent improvement over the baseline and each of the classes contributes to some improvement. F_1 for the positive class continues to improve with each additional lexicon. The case is similar for negative F_1 and *accuracy*, which improve up to DOM.

In Table 8 we look closer at how the classes are being matched and the impact this has on sentiment prediction. The left side of the table reports the number of sentences with class matches and breaks down how often they help ($\uparrow$), hurt ($\downarrow$), or stay the same ($\leftrightarrow$) compared to the previous lexicon (e.g., ADJ vs the baseline). We see that for the first three classes we match fewer bigrams but consistently help sentiment prediction. Adding the DOM lexicon helps more than it hurts but the numbers are more

balanced. We also see more cases where the prediction remains the same. This is because much of the dominant sentiment learned is consistent with the original lexicon. For example, matching the dominator *good* in the bigram *good decision*, while a correct match, will have no effect on the bigram polarity, since *good* was already matched from the HL lexicon as a unigram.

Without manual labels for all bigrams, another option for roughly checking the quality of the bigram matches is to check if the sentiment of the bigram is consistent with that of the sentence. This is shown in the right side of Table 8 with a report of the bigram matches. Multiple bigrams can match in a sentence so there may be more bigram matches than sentence matches. If the bigram is given the same sentiment as the sentence we count it as consistent ($=$), otherwise inconsistent ($\neq$). This does not strictly mean that the bigram polarity is correct or not, but it does give us a rough idea if the sentiment is correct and if it helps improve the sentence sentiment prediction. The results show that all the classes are consistent with the sentence sentiment in most of the cases. The dominators have far more matches but they are also much noisier than the other classes.

7 Conclusion

We presented a new approach for learning sentiment composition from a large, unlabeled corpus, which only requires a word-level sentiment lexicon for supervision. Our method learns lexicons that can address important sentiment composition phenomena such as sentiment reversals and mixed sentiment expressions. Manual assessment of the predictions made by these classes showed promising results. The classes were also shown to improve both phrase-level and sentence-level sentiment classification.

The automatically learned sentiment lexicons of bigrams and unigrams proved to be very useful for research on sentiment composition. We made them publicly available, and hope it will promote further research in this area. It would also be interesting to apply our method to languages other than English. Finally, we would like to investigate how much our results can be further improved in a semi-supervised setting, where the automatically-acquired classes are filtered by human annotators.

References

Silvio Amir, Ramón Astudillo, Wang Ling, Bruno Martins, Mario J. Silva, and Isabel Trancoso. 2015. Inesc-id: A regression model for large scale twitter sentiment lexicon induction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 613–618, Denver, Colorado, June. Association for Computational Linguistics.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017a. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 251–261, Valencia, Spain, April. Association for Computational Linguistics.

Roy Bar-Haim, Lilach Edelstein, Charles Jochim, and Noam Slonim. 2017b. Improving claim stance classification with lexical knowledge expansion and context utilization. In *Proceedings of the 4th Workshop on Argument Mining*, pages 32–38, Copenhagen, Denmark, September. Association for Computational Linguistics.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 793–801, Honolulu, Hawaii, October. Association for Computational Linguistics.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Li Dong, Furu Wei, Shujie Liu, Ming Zhou, and Ke Xu. 2015. A statistical parsing framework for sentiment classification. *Computational Linguistics*, 41(2):293–336, June.

Brian S. Everitt. 1992. *The analysis of contingency tables*. Chapman & Hall/CRC.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Daisuke Ikeda, Hiroya Takamura, Lev-Arie Ratinov, and Manabu Okumura. 2008. Learning to shift the polarity of words for sentiment classification. In *In Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP)*.

Alistair Kennedy and Diana Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22:2006.

Svetlana Kiritchenko and Saif M. Mohammad. 2016. Sentiment composition of words with opposing polarities. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1102–1108, San Diego, California, June. Association for Computational Linguistics.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of RANLP 2007*, Borovets, Bulgaria.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using crfs with hidden variables. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794, Los Angeles, California, June. Association for Computational Linguistics.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2010. Recognition of affect, judgment, and appreciation in text. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 806–814, Beijing, China, August. Coling 2010 Organizing Committee.

Samira Noferesti and Mehrnoush Shamsfard. 2016. Using data mining techniques for sentiment shifter identification. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses: An introduction*. Wiley.

Livia Polanyi and Annie Zaenen. 2004. Contextual valence shifters. In *Working Notes — Exploring Attitude and Affect in Text: Theories and Applications (AAAI Spring Symposium Series)*.

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, San Diego, California, June. Association for Computational Linguistics.

Marc Schulder, Michael Wiegand, Josef Ruppenhofer, and Benjamin Roth. 2017. Towards bootstrapping a polarity shifter lexicon using linguistic features. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 624–633, Taipei, Taiwan, November. Asian Federation of Natural Language Processing.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October. Association for Computational Linguistics.

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

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, October.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 172–182, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.